



UNIVERSITY
OF
INFORMATION TECHNOLOGY

ICAiT 2017

**Proceedings of
the 1st International Conference on
Advanced Information Technologies**

**November 1 - 2, 2017
Yangon, Myanmar**

Proceedings of the 1st International Conference on
Advanced Information Technologies

ICAIT 2017

1st and 2nd November

Organized by

University of Information Technology, Yangon

The Proceedings of 1st International Conference on Advanced Information Technologies (ICAIT) Committee Members

General Chair:

- Prof. Saw Sanda Aye, Rector of University of Information Technology, Myanmar

Co-Chair:

- Prof. Mie Mie Thet Thwin, Rector of University of Computer Studies, Yangon, Myanmar

Program-Chair:

- Prof. Aung Htein Maw, University of Information Technology, Myanmar

Organizing Committee:

- Prof. Moe Pwint, University of Computer Studies, Mandalay, Myanmar
- Prof. Win Aye, Myanmar Institute of Information Technology, Myanmar
- Prof. Thanda Thein, University of Computer Studies (Maubin), Myanmar
- Prof. Thinn Thu Naing, University of Computer Studies (Taunggyi), Myanmar
- Prof. Khin Mar Lar Tun, University of Computer Studies (Hinthada), Myanmar
- Prof. Aung Win, Rector of University of Technology (Yatanarpon Cyber City), Myanmar
- Prof. Soe Soe Khaing, University of Technology (Yatanarpon Cyber City), Myanmar
- Prof. Myat Thida Mon, University of Information Technology, Yangon, Myanmar
- Prof. Khin Moh Moh Tun, University of Information Technology, Yangon, Myanmar

- Prof. Swe Zin Hlaing, University of Information Technology, Yangon, Myanmar
- Prof. Thiri Haymar Kyaw, University of Information Technology, Yangon, Myanmar
- Prof. Aung Htein Maw, University of Information Technology, Yangon, Myanmar
- Prof. Htar Htar Lwin, University of Information Technology, Yangon, Myanmar
- Prof. Myat Thuzar Tun, University of Information Technology, Yangon, Myanmar
- Prof. Myint Thuzar Tun, University of Information Technology, Yangon, Myanmar
- Prof. Khin Kyawt Kyawt Khaing, University of Information Technology, Yangon, Myanmar
- Prof. Swe Swe Oo, University of Information Technology, Yangon, Myanmar

Programme Committee:

Foreign Professors

- Prof. Hiroyuki Iida, Japan Advanced Institute of Science and Technology (JAIST), Japan
- Prof. Hiroyuki Miyazaki, University of Tokyo, Japan
- Prof. Jaeyoung Ahn, Electronics and Telecommunications Research, Korea
- Prof. Kiyooki Shirai, Japan Advanced Institute of Science and Technology (JAIST), Japan
- Prof. Koichiro Ochimizu, Visiting Professor, University of Information Technology, Yangon, Myanmar
- Prof. Mitsuo Ikeda, Color Research Center, Rajamangala University of Technology Thanyaburi (RMUTT), Thailand
- Prof. Shinichi Honiden, Deputy Director General National Institute of Informatics (NII), Department of Computer Science, Graduate School of Information Science and Technology, University of Tokyo, Japan

- Prof. Takashi Watanabe, Graduate School of Information Science and Technology, Osaka University, Japan
- Prof. Tetsuya Shimamura, Department of Information and Computer Sciences, Saitama University, Japan
- Prof. Tomio Takara, University of the Ryukyus, Japan
- Prof. Toshitaka Tsuda, Waseda University, Japan
- Prof. Yoshinori Sagisaka, Waseda University, Japan
- Prof. Yuichi Otsuka, Nagaoka University of Technology, Japan
- Prof. Yutaka Ohsawa, Department of Information and Computer Sciences, Saitama University, Japan
- Associate Prof. Yuto Lim, Japan Advanced Institute of Science and Technology (JAIST), Japan
- Associate Prof. Komuro Takashi, Information and Computer Sciences, Saitama University, Japan
- Assistant Prof. Manasawee Kaenampornpan, Department of Computer Science, Mahasarakham University, Thailand
- Assistant Prof. Olarik Surinta, Department of Information Technology, Mahasarakham University, Thailand
- Dr. Ding Chenchen, Researcher, National Institute of Information and Communications Technology (NICT)/ Universal Communication Research Institute, Japan
- Dr. Ye Kyaw Thu, Researcher, Okayama Prefectural University (OPU), Japan

Local Professors

- Prof. Moe Pwint, University of Computer Studies, Mandalay, Myanmar
- Prof. Win Aye, Myanmar Institute of Information Technology, Myanmar
- Prof. Thanda Thein, University of Computer Studies (Maubin), Myanmar
- Prof. Thinn Thu Naing, University of Computer Studies (Taunggyi), Myanmar
- Prof. Khin Mar Lar Tun, University of Computer Studies (Hinthada), Myanmar
- Prof. Aung Win, Rector of University of Technology (Yatanarpon Cyber City), Myanmar
- Prof. Ei Chaw Htoon, Computer University (Kyaing Tong), Myanmar
- Prof. Kalyar Myo San, University of Computer Studies, Mandalay, Myanmar
- Prof. Khine Moe Nwe, University of Computer Studies, Yangon, Myanmar
- Prof. Khin Mar Soe, University of Computer Studies, Yangon, Myanmar
- Prof. Khin Nweni Tun, University of Computer Studies (Taunggyi), Myanmar
- Prof. Khin Than Mya, University of Computer Studies, Yangon, Myanmar
- Prof. Khin Thaida Lynn, University of Computer Studies, Mandalay, Myanmar
- Prof. Mie Mie Su Thwin, University of Computer Studies, Yangon, Myanmar
- Prof. Myint Myint Sein, University of Computer Studies, Yangon, Myanmar
- Prof. Nang Saing Moon Kham, University of Computer Studies, Yangon, Myanmar
- Prof. Nyein Nyein Myo, University of Computer Studies, Mandalay, Myanmar
- Prof. Nyein Aye, Computer University (Hpa-an), Myanmar
- Prof. Soe Soe Khaing, University of Technology (Yatanarpon Cyber City), Myanmar
- Prof. Sabai Phyu, University of Computer Studies, Yangon, Myanmar
- Prof. Su Thawda Win, University of Computer Studies, Mandalay, Myanmar
- Prof. Than Nwet Aung, University of Computer Studies, Mandalay, Myanmar

- Prof. Thi Thi Soe Nyunt, University of Computer Studies, Yangon, Myanmar
- Prof. Zin May Aye, University of Computer Studies, Yangon, Myanmar
- Associate Prof. Win Pa Pa, University of Computer Studies, Yangon, Myanmar
- Prof. Myat Thida Mon, University of Information Technology, Yangon, Myanmar
- Prof. Khin Moh Moh Tun, University of Information Technology, Yangon, Myanmar
- Prof. Htar Htar Lwin, University of Information Technology, Yangon, Myanmar
- Prof. Khin Kyawt Kyawt Khaing, University of Information Technology, Yangon, Myanmar
- Prof. Myat Thuzar Tun, University of Information Technology, Yangon, Myanmar
- Prof. Myint Thuzar Tun, University of Information Technology, Yangon, Myanmar
- Prof. Swe Zin Hlaing, University of Information Technology, Yangon, Myanmar
- Prof. Aung Htein Maw, University of Information Technology, Yangon, Myanmar
- Prof. Thiri Haymar Kyaw, University of Information Technology, Yangon, Myanmar

The Proceedings of 1st International Conference on Advanced Information Technologies (ICAIT)

November, 2017

Contents

1st November, 2017 (Wednesday)

Keynote Speech

Professor Yoichi Shinoda, School of Information Science, Japan Advanced Institute of Science and Technology	i
Associate Professor Fuyuki Ishikawa, National Institute of Informatics, Japan	ii

Cloud Computing and Big Data Analytics

Dynamic Replication Management Scheme for Cloud Storage <i>May Phyo Thu, Khine Moe New, Kyar Nyo Aye</i>	1-7
Performance Analysis of a Scalable Naïve Bayes Classifier on Beyond MapReduce <i>Myat Cho Mon Oo, Thandar Thein</i>	8-13
Cloud Based Big Data Application of FP-Growth Algorithm and K-Means Clustering Algorithm Based on MapReduce Hadoop <i>Than Htike Aung, Nang Saing Moon Kham</i>	14-19
Analytics of Reliability for Real-Time Big Data Pipeline Architecture <i>Thandar Aung, Aung Htein Maw</i>	20-25
Optimum Checkpoint Interval for MapReduce Fault-Tolerance <i>Naychi Nway Nway, Julia Myint</i>	26-30
Forensic Analysis of Residual Artifacts on CDH Storage <i>Myat Nandar Oo, Thandar Thein</i>	31-37

Networking and Network Security

Stateful Firewall Application on Software Defined Networking <i>Nan Haymarn Oo, Aung Htein Maw</i>	39-45
An Analysis of Decision Tree Based Intrusion Detection System <i>Yi Yi Aung, Myat Myat Min</i>	46-51
Delay Controlled Elephant Flow Rerouting in Software Defined Network <i>Hnin Thiri Zaw, Aung Htein Maw</i>	52-57
Bandwidth Allocation Scheme using Segment Routing on Software-Defined Network <i>Ohmmar Min Mon, Myat Thida Mon</i>	58-64

Data Science

Uniformly Integrated Database Approach for Heterogenous Databases <i>Hlaing Phyu Phyu Mon, , Thin Thin San, Zinmar Naing, Thandar Swe</i>	65-69
Analysis of Historical Census Household data with Similarity Threshold Method <i>Khin Su Mon Myint, Thet Thet Zin, Kyaw May Oo</i>	70-75
Multidimensional Analysis for Census Data by Applying Star Schema Model <i>Myint Myint Thein, Myint Myint Lwin, Aye Chan Mon, May Thu Aung</i>	76-81
Data Compression Strategy for Reference-Free Sequencing FASTQ Data <i>Hsu Mon Lei Aung, Swe Zin Hlaing</i>	82-85

Natural Language Processing

Domain-specific Sentiment Dictionary Construction for Sentiment Classification <i>Aye Aye Ma, Nyein Thwet Thwet Aung, Su Su Htay</i>	87-92
Domain-Specific Sentiment Lexicon for Classification <i>Thet Thet Zin, Kay Thi Yar, Su Su Htay, Khine Khine Htwe, Nyein Thwet Thwet Aung, Win Win Thant</i>	93-98
Feature Selection for Categorization of Online News Articles in Myanmar Language <i>Myat Sapal Phyu, Win Win Thant, Thet Thet Zin</i>	99-105

Software Engineering and Web Mining

- Defining a Software Engineering Process with Cost-effective Security Requirements Implementation 107-111
Swe Zin Hlaing, Koichiro Ochimizu
- A Lightweight Size Estimation Approach for Embedded System using COSMIC Functional Size Measurement 112-118
Thandar Zaw, Swe Zin Hlaing, Myint Myint Lwin, Koichiro Ochimizu
- Mining Web Content Outliers by using Term Weighting Technique and Rank Correlation Coefficient Approach 119-123
Thinzar Tun, Khin Mo Mo Tun

Image and Signal Processing

- Lane Detection System based on Hough Transform with Retinex Algorithm 125-130
Shwe Yee Win, Htar Htar Lwin
- Sparse Representation for Paddy Plants Nutrient Deficiency Tracking System 131-137
Zar Zar Tun, Khin Htar Nwe

Mobile and Distributed Computing

- The Home Safety System Based on Competing Functions 139-144
Zaw Myint Naing Oo, Khin Kyawt Kyawt Khaing
- Range Tree Based Indexing of Mobile Tracking System 145-150
Thu Thu Zan, Sabai Phyu
- Automatic Adjustment of Read Consistency Level of Distributed Key-value Storage by a Replica Selection Approach 151-156
Thazin New, Tin Tin Yee, Myat Pwint Phyu, Ei Chaw Htoon, Junya Nakamura

2nd November, 2017 (Thursday)

Workshop Session

An Approach of Accessing Small Files on HDFS for Cloud Storage <i>Khin Su Su Wai</i>	157-160
Resource-based Data Placement Strategy for Hadoop Distributed File System <i>Nang Kham Soe</i>	161-164
Parallel PAM Clustering Algorithm for Learning Analytics <i>Nway Yu Aung</i>	151-169
An Integrative Access Control with an Attributes-based Event Handler for Data Protection in Cloud Storage <i>Phyo Wah Wah Myint</i>	170-175
Availability Modelling for SDN switch in Cloud based Infrastructure <i>May Thae Naing</i>	176-179
Computation Offloading Decision in Mobile Cloud Computing: Enhance Battery Life of Mobile Device <i>Mi Swe Zar Thu</i>	180-185
Land Use Classification using Deep Convolutional Neural Network <i>Su Wai Tun</i>	186-189
Evaluation of Face Recognition Techniques for Facial Expression Analysis <i>Hla Myat Maw</i>	189-194
RFSgIndex: Frequent Subgraph Index for Subgraph Matching in RDF Data <i>Khin Myat Kyu</i>	195-199
A Functional Resonance Analysis Method to risk analysis of functional flood defenses in Yangon <i>Kyi Pyar Hlaing</i>	200-203
Feature Extraction Method for Aspect-Based Sentiment Analysis <i>Win Lei Kay Khine</i>	204-207
A Personalized Recommendation System Using Collaborative Filtering With Feature Based Sentiment Analysis <i>Nyein Ei Ei Kyaw</i>	208-210
Efficient Classification of Concept Drift in Data Stream <i>Ei Thwe Khaing</i>	211-213

Keynote Speech

Keynote Speech



Professor Yoichi Shinoda
School of Information Science, Security and Networks Area
Japan Advanced Institute of Science and Technology

Degrees:

B.E., M.E. and Ph.D. from Tokyo Institute of Technology (1983,1985,1989)

Professional Career:

Associate at Tokyo Institute of Technology (1988),
Professor of School of Information Science at JAIST (1991)

Specialties:

Distributed and Parallel Computing
Networking Systems
Operating Systems
Information Environment

Keynote Speech

Emerging Challenges in Software Dependability under Uncertain World



Associate Professor Fuyuki Ishikawa
National Institute of Informatics, Japan

Associate Professor of Content Science, National Institute of Informatics

Visiting Associate Professor, University of Electro-Communications, Graduate School of Informatics

Institute National Polytechnique de Toulouse (France) Visiting Professor

Research Area: Trustworthy & Smart Software Engineering

Abstract

The society and human activities have been depending more and more on software-intensive systems. Novel emerging systems are stepping into more depth of human activities as well as real world entities, such as smart AI systems and cyber-physical systems. Those systems are required to deal with more uncertainty in human users and the real, physical world. This fact makes traditional approaches for software dependability insufficient. First, it is at least too difficult and costly to define “right answers” about what smart AI systems should do for prediction and recommendation for a variety of possible inputs. Then we cannot test the software by preparing expected outputs for a lot of possible inputs as in traditional testing approaches. Second, it is almost impossible to give comprehensive and valid assumptions about what may happen at runtime in the real world. Then we cannot be confident of our software only by testing or verification beforehand at the development time. This talk discusses these challenges and changes in approaches to dependability by testing and verification in the state-of-the-art research.

Cloud Computing and Big Data Analytics

Dynamic Replication Management Scheme for Cloud Storage

May Phyo Thu, Khine Moe Nwe, Kyar Nyo Aye

University of Computer Studies, Yangon

mayphyothu.mpt1@gmail.com, khinemoenwe@ucsy.edu.mm, kyarnyoaye@gmail.com

Abstract

Nowadays, replication technique is widely used in data center storage systems to prevent data loss. Data popularity is a key factor in data replication as popular files are accessed most frequently and then they become unstable and unpredictable. Moreover, replicas placement is one of key issues that affect the performance of the system such as load balancing, data locality etc. Data locality is a fundamental problem to data-parallel applications that often happens (i.e., a data block should be copied to the processing node when a processing node does not possess the data block in its local storage), and this problem leads to the decrease in performance. To address these challenges, this paper proposes a dynamic replication management scheme based on data popularity and data locality; it includes replica allocation and replica placement algorithms. Data locality, disk bandwidth, CPU processing speed and storage utilization are considered in the proposed data placement algorithm in order to achieve better data locality and load balancing effectively. Our proposed scheme will be effective for large-scale cloud storage.

Keywords- Replication, Data Popularity, Data locality, Storage utilization, Disk Bandwidth

1. Introduction

Cloud storage is a technology that allows us to save files in storage and then access those files via Cloud. The cloud storage system convergences data storage among multiple servers into a single storage pool and provides users with immediate access to a broad range of resources and applications hosted in the infrastructure of another organization via a web service interface [6].

Cloud storage systems may consist of a cluster of storage nodes or even geographically distributed data centers. At present, the existing Cloud storage products are Google (Google File System GFS), Amazon (Simple Storage Service S3), IBM (Blue Cloud), Yahoo (Hadoop Distributed File System HDFS) etc. HDFS provides reliable storage and high throughput access to application data. In HDFS, data is split in a fixed size (e.g., 32MB, 64MB, and 128MB) and the split data blocks (chunks) are distributed and stored in multiple data nodes with replication.

In HDFS, to provide data locality, Hadoop tries to automatically collocate the data with the computing node. Hadoop schedules Map tasks to set the data on same node and the same rack. Data locality is a principal factor of Hadoop's performance. The data locality problem occurs when the assigned node should load the data block from a different node storing the data block. Data locality means the degree of distance between data and the processing node for the data.

There are two ways in order to improve data locality:

1. The replica allocation problem occurs when popular data are assigned a larger number of replicas to improve data locality of concurrent accesses.
2. The replica placement problem occurs when different data blocks accessed concurrently are placed on different nodes to reduce contention on a particular node.

There are three types of data locality in Hadoop: node locality, rack locality and rack-off locality. Uniform data replication is used in current implementations of MapReduce systems (e.g., Hadoop). The concept of popularity of files is introduced to replication strategies for selecting a popular file in reality. File popularity represents whether a file has been hot in recent time intervals, which is computed by file access rate.

In this paper, therefore, data popularity based replication method is proposed to overcome the problems of static replication in HDFS and to support better efficiency in cloud storage. Firstly, the rate of change of file popularity is calculated by analyzing the access histories with first order differential equation. Secondly, the replication degree for each file is calculated according to the rate of change of file popularity. Finally, the replicas will be placed based on proposed data placement algorithm.

The rest of this paper is organized as follows. Section 2 describes related works and background theory is presented in section 3. Section 4 presents proposed system architecture and finally, section 5 describes the conclusion and future work.

2. Related Works

In cloud storage environment, data can be stored with some geographical or logical distance and this data is accessible to cloud based applications. Data is replicated

and stored in multiple data nodes to provide high availability and load balancing. There were several previous researches of data replication in HDFS. A cost effective replication management scheme for cloud storage cluster was proposed by Qingsong Wei [2]. That paper aimed to improve file availability by centrally determining the ideal number of replicas for a file, and an adequate placement strategy based on the blocking probability. However, this method wasn't good for very large file that was file size was Terabyte and the effects of increasing locality were not studied.

One approach, Latest Access Largest Weight (LALW) algorithm [8], that was proposed by R.S. Chang and H.P.Chang for data grids. LALW found out the most popular file in the end of each time interval and calculated a suitable number of copies for that popular file and decides which grid sites were suitable to locate the replicas.

A. Hunger and J. Myint compared two data popularity-based replication algorithms: PopStore and Latest Access Largest Weight (LALW) [1]. In that paper, both algorithms found more popular files according to the time intervals through the concept of Half-life. However, this paper did not consider for load balance in replica placement.

Recently, a few studies attempted to improve data locality with data replication in Hadoop. Scarlett [5] adopted a proactive replication scheme that periodically replicates files based on predicted data popularity. It focused on data that receives at least three concurrent accesses. However, it did not consider node popularity caused by co-location of moderately popular data.

In DARE[3], the authors proposed a dynamic data replication scheme based on access patterns of data blocks during runtime to improve data locality. DARE adopted a reactive approach that probabilistically retained remotely retrieved data and evicted aged replicas. DARE allowed to increase the data replication factor automatically by replicating the data to the fetched node. However, removing the replicated data was performed when only the available data storage was insufficient. Thus, it had a limit to provide the optimized replication factor with data access patterns.

In [9], the authors proposed a delay scheduling method that focused on the conflict between data locality and fairness among jobs. Although the delay scheduling method was designed to improve data locality, it let the jobs wait for a small amount of time, resulting in violating the fairness for jobs. Moreover, delay scheduling made assumptions that might not hold universally: (a) task durations were short and bimodal, and (b) a fixed waiting time parameter worked for all loads and skewness of traffic. These assumptions made it difficult for delay scheduling to adapt to changes in workload, network conditions, or node popularity.

In [4], the authors proposed an efficient data replication scheme based on access count prediction in a Hadoop framework. This data replication scheme determined the replication factor with the predicted data access count, whether it generated a new replica or it used the loaded data as cache selectively. Although this scheme was designed to improve data locality, it considered file level replication did not consider block level replication.

3. Background Theory

In large-scale distributed system, replication is a general technology that can improve the efficiency of data access and the fault-tolerance. Data locality is a principal factor of Hadoop's performance. In Hadoop scheduling policy, the data locality problem occurs; that is, the assigned node should load the data block from a different node storing the data block. The proposed dynamic replication management scheme considers the data popularity and data locality. This section describes architecture of Hadoop cluster and data locality.

3.1 Architecture of Hadoop Cluster

Hadoop is an open source software framework that supports data intensive distributed applications. The architecture of a Hadoop cluster can be divided into two layers: MapReduce and HDFS (Hadoop Distributed File System). The MapReduce layer maintains MapReduce jobs and their tasks, and the HDFS layer is responsible for storing and managing data blocks and their metadata. HDFS stores three replicas of each block like Google File System (GFS) [7].

A job tracker in the master node splits a MapReduce job into several tasks and the split tasks are scheduled to task trackers by the job tracker. For the purpose of monitoring the state of task trackers, the job tracker aggregates the heartbeat messages from the task trackers. When storing input data into the HDFS, the data are split in fixed sized data blocks with replication (the default replication factor is 3) and the split data blocks (chunks) are stored in slave nodes. A task tracker of a slave node is in charge of scheduling tasks in the node. A task tracker requests a task from a job tracker by sending a heartbeat message when it has an empty task slot.

When storing input data from a client, the data are divided into chunks and the chunks are stored to nodes. The job tracker deals with a MapReduce job request from a client. Upon reception of a job request, the job tracker divides a job into tasks, and then, the tasks are assigned to task trackers. At this stage, it schedules tasks by considering data locality. Next, each task tracker assigns a task to a node, and then, the node performs the task by loading the data block from HDFS when needed.

Users submit jobs consisting of a map function and a reduce function. Hadoop breaks each job into tasks. First, input data are divided into fixed size units processed independently and in parallel by map tasks, which are executed distributively across the nodes in the cluster. There is one map task per input block. After the map tasks are executed, their output is shuffled, sorted and then processed in parallel by one or more reduce tasks.

3.2 Data Locality

Data in Hadoop is stored in HDFS. This data is divided into blocks and stored across the data nodes in a Hadoop cluster. When a MapReduce job is executed against the dataset, the individual Mappers will process the blocks (input splits). When data is not available for Mapper in the same node, then data has to be copied over the network from the data node that has data to the data node that is executing the Mapper task. This is known as a data locality.

Data locality related with the distance between data and the processing node. So, if the closer distance between data and node, it has the better data locality. There are three types of data locality in Hadoop:

- (1) Node locality: when data for processing are stored in the local storage,
- (2) Rack locality: when data for processing are not stored in the local storage, but another node within the same rack,
- (3) Rack-off locality: when data for processing are not stored in the local storage and nodes within the same rack, but another node in a different rack.

Figure 1 shows three types of data locality in Hadoop: node locality, rack locality, and rack-off locality.

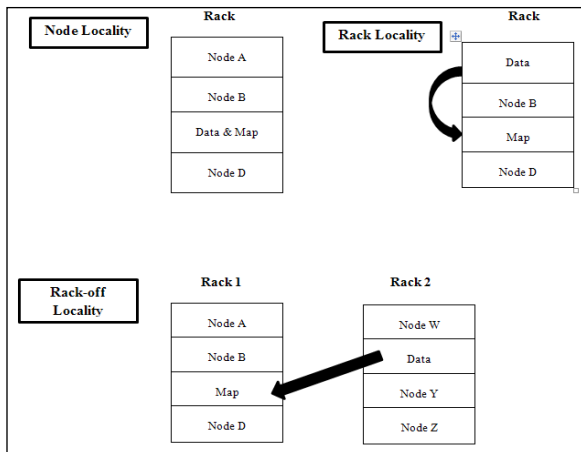


Figure 1. Types of data locality

Among these types of data locality, the most preferred scenario is node locality and the least preferred scenario is rack-off locality. The data locality problem is a situation

where a task is scheduled with rack or rack-off locality. Moreover, the overhead of rack-off locality is greater than that of rack locality. To prevent the data locality problem, we propose a dynamic data replication scheme using prediction by the access count of data files and a data placement algorithm reducing case of rack and rack-off locality.

4. Proposed System Architecture

The basic idea of replication is based on the different replication degree per data file. Maintaining the static number of replicas in the system results highly storage cost for unpopular data and inefficient for most accessed data. Moreover, maintaining too much replication degree than the current access count for a data file does not always guarantee the better data locality for all data blocks.

The goal of proposed system is to design an adaptive replication scheme that seeks to increase data locality by replicating “popular” data while keeping a minimum number of replicas for unpopular data. Because the nature of data access pattern is random, a method that predicts the rate of change of file popularity for the next time slot is required. The proposed system flow diagram is shown in figure 2.

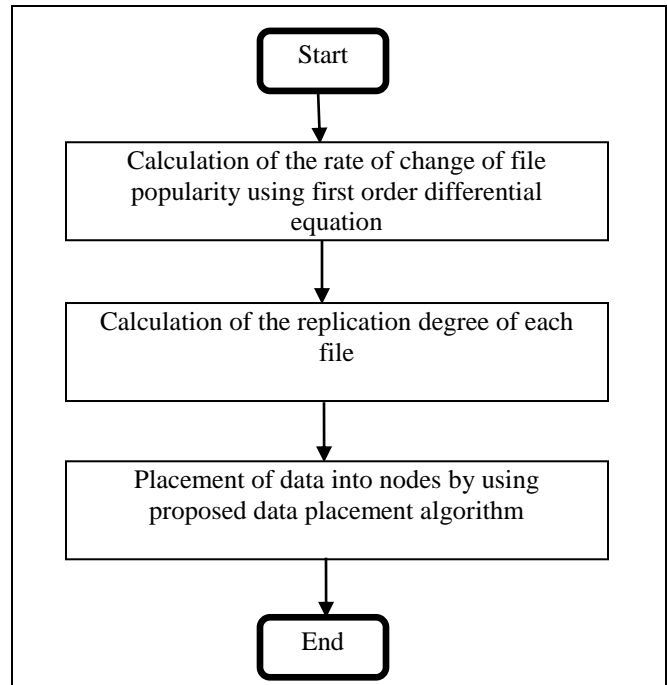


Figure 2. Proposed system flow diagram

The proposed scheme includes three-step processes: , the rate of change of file popularity will be calculated using first order differential equation in the first step and

the number of replicas of each file will be calculated in the second step and then the replicas will be placed into nodes based on proposed data placement algorithm in the third step.

4.1. Proposed Popularity Growth Rate Algorithm

In this step, the rate of change of file popularity will be calculated using first order differential equation. LALW and Pop-Store algorithms applied half-life strategy which means the weight of the records in an interval decays to half of its previous weight.

The idea of popularity is the assumption that the rate at which a popularity of an item grows at a certain time is proportional to the total popularity of the item at that time. In mathematical terms, if $P(t)$ denotes the total population at time t , then this assumption can be expressed as

$$\frac{dp}{dt} = kP(t) \quad (1)$$

where $P(t)$ denotes population at time t and k is the growth constant or the decay constant, as appropriate. If $k > 0$, we have growth, and if $k < 0$, we have decay. It is a linear differential equation which solve into

$$P(t) = P_0 e^{kt} \quad (2)$$

Then,

$$k = \frac{\ln\left(\frac{P(t)}{P_0}\right)}{t} \quad (3)$$

Where P_0 is the initial population, i.e. $p(0) = P_0$, and k is called the growth or the decay constant. In this step, the rate of change of file popularity will be calculated by using Yahoo Hadoop audit log file [10] as data source. Users can enable audit logging from the NameNode. Audit events are emitted as a set of key/value pairs for the following keys as shown in table 1.

Table 1. Key/Value Pairs of User Audit Log

Key	Value
ugi	<user>, <group>[,<group>]*
ip	<client ip address>
cmd	(open create delete rename mkdirs listStatus setReplication setOwner setPermission)
src	<path>
dst	(<path> "null")
perm	(<user>:<group>:<perm mask> "null")

The Yahoo HDFS User Audit log format is shown in figure 3.

```
2016-12-10 11:11:59,693 INFO
org.apache.hadoop.hdfs.server.namenode.FSNamesystem.
audit: ugi=hduser ip=/134.91.100.59 cmd=delete
src=/app/hadoop/tmp/test.txt dst=null perm=null
```

Figure 3. HDFS user audit log format

To get the frequency count of each file, user audit log is split into small files based on timeslot duration and number of records. Then the required fields are extracted such as Date, Time, IP and src. After that, the user access frequency is counted from src source link from figure 4. In each time slot, the access frequency is counted and stored for individual files. Then the rate of change of file popularity of each file is calculated on each time slot according to table 2 and algorithm 1.

Table 2. Notations Used in Popularity Growth Rate Algorithm

Notation	Description
$P(t_f)$	Popularity values of file f
$AF(t_f)$	Total access frequency of file f at each time slot
inLog	The input log file
k	The rate of change of file popularity

Algorithm 1: Popularity Growth Rate Algorithm

Input: inLog

Output: k

1. Read inLog
2. Calculate access frequency of each file by using $P(t_f) = AF(t_f), \forall f \in F$
3. Calculate the rate of change of file popularity k of each file by substituting $P(t) = P(t_f)$ in (3)
4. return k .

Figure 4. Popularity growth rate algorithm

In order to verify our proposed popularity growth rate algorithm, we suppose three files (x_1, x_2 and x_3) in three time slots. Each time slot duration is set as 10 seconds, therefore, ($t_1 = t_2 = t_3 = 10$ seconds). Let $P_0 = 1, P(t) = AF(t_f)$ and calculate k by using equation (3). Suppose access frequencies of file x_1, x_2 and x_3 in time slot 1 are 40, 1100 and 200. In time slot 1, the rate of change of file popularity k in file x_1, x_2 and x_3 is 0.3688, 0.7003 and 0.5298. Also, in time slot 2, access frequencies of file x_1, x_2 and x_3 are 400, 100 and 900. Therefore, the rate of change of file popularity k in file x_1, x_2 and x_3 for time slot 2 is 0.5991, 0.4605 and 0.6802. Also, in time slot 3, access frequencies of file x_1, x_2 and x_3 are 2200, 1200 and 20. Therefore, the rate of change of file popularity k

in file x1, x2 and x3 for time slot 3 is 0.7696, 0.7090 and 0.2996.

According to the rate of change of file popularity, replica degree for each file is considered as follows. If the rate of change of file popularity is greater than 0.0, then existing replica degree is increased by one. If the rate of change of file popularity is less than 0.0, then existing replica degree is decreased by one. If the rate of change of file popularity is equal to 0.0 then existing replica degree is remained unchanged. Otherwise, if the accessed file is new and there is no access record history, the replica degree for this file will be assigned 3 as like HDFS default replica number.

4.2. Proposed Data Placement Algorithm

After determining the number of replicas, we will consider how to place these replicas efficiently in order to improve data locality and load balancing. In this step, let me assume that the jobs will have to access this replica in the next time slot. The incoming job is broken into tasks and each map task is assigned into nodes within the cluster. There is one map task per input block.

In this system, the input data file is divided into 64 MB blocks and place them into blocks within the cluster. For instance, if the replica for this file is 3 and this file has 4 blocks, then the total replica block number of this file is 12. Let the maximum number of replicas be the number of nodes in the cluster and the minimum number of replicas be 1.

Suppose that at the assigned node, there is no replica block for the incoming map task. In this case, this system considers for improvement of node locality. In this case, the remote data retrieval is performed by loading the replica data block into this node. While loading this data block, if the load factor of this node is less than the predefined threshold, this replica data block is loaded into this node. Otherwise, the replacement is performed by replacing this replica data block with existing block into this node.

The proposed data replacement algorithm is based on Least Recently Used (LRU). It will be more reliable than the LRU and will have the more efficient results than the LRU algorithm because it considers access frequency for replacement. Firstly, this proposed algorithm considers the block with minimum access frequency for replacement. Secondly, if one or more blocks with minimum access frequency, it considers least recently accessed block (outgoing block) for replacement according to LRU mechanism. The proposed enhanced LRU replacement algorithm is shown in figure 5.

Algorithm 2: Enhanced LRU Algorithm

Step 1. When loading the replica data block into the assigned node, it will calculate total number of access frequencies (TAF) for all blocks in that node.

Step 2. If only one block with minimum TAF is found, that block will be selected to evict from that node.

Step 3. If one or more minimum TAF blocks are found, least recently accessed block (outgoing block) will be selected to evict from that node as LRU.

Figure 5. Enhanced LRU Algorithm

The existing Hadoop block placement strategy does not take into account DataNodes' utilization, which leads to in an imbalanced load. Since the DataNode selection for the block placement is random, the disk bandwidth of the allotted DataNode may be less than or greater than the available bandwidth. This policy assumes that all nodes in the cluster are homogeneous, and randomly place blocks without considering any nodes' resource characteristics, which decreases self-adaptability of the system. Therefore, this system considers the heterogeneous environment for nodes in the cluster. We need to consider the load factor such as storage utilization, disk bandwidth and CPU processing speed. During the process of placement, the storage utilization, disk bandwidth and CPU processing speed of DataNode are important factors to affect the load balancing in HDFS. Therefore, the capacity of DataNode stored should be proportional to its total disk capacity, in the condition of effective load balancing. We can carry out the storage utilization model as

$$U(D_i) = \frac{D_i (use)}{D_i (total)} \quad (4)$$

Where, $U(D_i)$ is the storage utilization of the i^{th} DataNode. $D_i(use)$ is the used disk capacity of the i^{th} DataNode, and its unit is GB. $D_i(total)$ is the total disk capacity of the i^{th} DataNode, it is a fixed value of each DataNode, and its unit is GB.

Then, we can carry out the disk bandwidth model as

$$BW(D_i) = \frac{T_b}{T_s} \quad (5)$$

Where, $BW(D_i)$ is the disk bandwidth of the i^{th} DataNode. T_b is the total number of bytes transferred, and T_s is the total time between the first request for service and the completion of the last transfer.

Then, the CPU processing speed is used as one of the important factors and each node has different CPU processing speed due to the heterogeneous environment. Among these three factors, storage utilization is set as first priority, disk bandwidth as second priority and CPU processing speed as last priority. So, we put the coefficients of storage utilization, disk bandwidth and CPU processing speed are set as 0.5, 0.3 and 0.2. Therefore, we can carry out the load factor model as

$$LF(D_i) = 0.5U(D_i) + 0.3BW(D_i) + 0.2SP(D_i) \quad (6)$$

The predefined threshold T_i of the i^{th} cluster is assumed as the sum of maximum storage utilization, maximum disk bandwidth and maximum CPU processing speed in cluster is divided by the number of nodes in the cluster. Therefore, we can carry out the predefined threshold of cluster C_i as

$$T_i = \frac{Max_i (U) + Max_i (BW) + Max_i (SP)}{N} \quad (7)$$

Where, T_i is the predefined threshold of the i^{th} cluster and N is the number of nodes in the i^{th} cluster. If the load factor of this node is less than the predefined threshold, this replica data block is loaded into this node. Therefore, the storage utilization, disk utilization and CPU processing speed of DataNode are used in proposed data placement algorithm as shown in table 3 and algorithm 3.

Table 3. Notations Used in Data Placement Algorithm

Notation	Description
DN	DataNodes list
BW	Bandwidth
U	Storage utilization
RP	Replica List
MT	Map task list
SP	CPU processing speed
C	Cluster list
LF	Load factor list

5. Conclusion

In cloud storage environment, data can be stored with some geographical or logical distance and this data is accessible for cloud based applications. Data is replicated and stored in multiple data nodes to provide for data availability. In this paper, a dynamic replication management scheme is proposed for cloud storage. At each time intervals, the proposed system collects the data access history in cloud storage. According to access frequencies for all files that have been requested, the change of popularity rate can be calculated and replicated them to suitable DataNodes in order to achieve load balance and node locality of system. As a future work, many experimental evaluations have to be carried out in order to get the efficiency of proposed data placement algorithm. In addition, many experimental evaluations have to be performed in order to get better threshold value and load factor value. And as well, replica deallocation will be considered for overall system improvement.

Algorithm 3: Data Placement Algorithm

Input: DataNodes List $DN = \{DN_1, DN_2, \dots, DN_n\}$,
Replica List $RP = \{RP_1, RP_2, RP_3, \dots, RP_n\}$, **Map Task List** $MT = \{MT_1, MT_2, MT_3, \dots, MT_n\}$, **Load Factor List** $LF = \{LF_1, LF_2, LF_3, \dots, LF_n\}$, **Predefined Threshold** T_i ,
Cluster List $C = \{C_1, C_2, C_3, \dots, C_n\}$
Output: DataNodes List DN
for each incoming map task MT **do**
 for each DataNode DN **do**
 Check node locality of task MT_i
 if there is node locality **then** assign task MT_i to that DataNode DN_i
 else
 Perform remote data replica retrieval for task MT_i
 Calculate storage utilization U of this assigned DataNode DN_i using (4)
 Calculate disk bandwidth BW of this assigned DataNode DN_i using (5)
 Check CPU processing speed SP of this assigned DataNode DN_i
 Calculate load factor LF_i for this assigned DataNode DN_i using (6)
 Calculate predefined threshold T_i for the cluster C_i
 if $LF_i >$ predefined threshold T_i **then**
 Perform replacement using algorithm 2
 Place replica RP_i for this task on that DataNode DN_i
 break
 else
 Place replica RP_i for this task on that DataNode DN_i
 break
 end if
 end for
end for

Figure 6. Data Placement Algorithm

6. References

- [1] A. Hunger and J. Myint, "Comparative Analysis of Adaptive File Replication Algorithms for Cloud Data Storage", *2014 International Conference on Future Internet of Things and Cloud*, 2014.
- [2] B. Gong, B. Veeravalli, D. Feng L. Zeng, and Q. Wei, "CDRM: A Cost-Effective Dynamic Replication Management Scheme for Cloud Storage Cluster", *2010 IEEE International Conference on Cluster Computing*, Sep. 2010, pp. 188–196.
- [3] C.L. Abad, Yi Lu, R.H. Campbell, "DARE: Adaptive Data Replication for Efficient Cluster Scheduling", *IEEE International Conference on Cluster Computing (CLUSTER 2011)*, pp.159-168, 2011.
- [4] D.Lee, J.Lee, and J.Chung, "Efficient Data Replication Scheme based on Hadoop Distributed File System",

International Journal of Software Engineering and Its Applications Vol. 9, No. 12 (2015), pp. 177-186,2015.

[5] G. Ananthanarayanan *et al.*, “Scarlett: Coping with skewed content popularity in mapreduce clusters,” in *Proc. Conf. Comput. Syst. (EuroSys)*, 2011, pp. 287–300.

[6] H. Gobioff, S. Ghemawat, and S.-T. Leung, “The Google File System”, *Proceedings of 19th ACM Symposium on Operating Systems Principles (SOSP 2003)*, New York, USA, October, 2003.

[7] H. Hardware, and P. Across, “The Hadoop Distributed File System: Architecture and Design”, 2007, pp. 1–14.

[8] H.-P. Chang, R.-S. Chang, and Y.-T. Wang, “A dynamic weighted data replication strategy in data grids”, *2008 IEEE/ACS International Conference on Computer Systems and Applications*, Mar. 2008, pp. 414–421.

[9] M. Zaharia, D. Borthakur, J. Sen Sarma, K. Elmeleegy, S. Shenker, and I. Stoica, “Delay scheduling: A simple technique for achieving locality and fairness in cluster scheduling”, *In Proceeding of uropean Conference Computer System (EuroSys)*, 2010.

[10] <https://webscope.sandbox.yahoo.com>.

[11] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice-Hall, 1992.

Performance Analysis of a Scalable Naïve Bayes Classifier on Beyond MapReduce

Myat Cho Mon Oo¹, Thandar Thein²

University of Computer Studies, Yangon¹, University of Computer Studies, Maubin²
myatchomonoo@ucsy.edu.mm¹, thandartheinn@gmail.com²

Abstract

Many real world areas from different sources generate the massive data with large volume of high velocity, complex and variable data. The massive data becomes a challenge for machine learning algorithm because they are difficult to process and extract knowledge using traditional analysis tools. Therefore, a massively scalable and parallel algorithm is needed to process and analyze such massive data. Recently Hadoop MapReduce framework has been adapted for processing large data in an extremely parallel mining. MapReduce may not a very good fit for most of the scalable machine learning that Mahout pioneered. For large scale machine learning on distributed system, Mahout Samsara is used with efficient distributed execution on Spark. This paper analyses the scalability of Naïve Bayes classifier which is implemented by Mahout Samsara. The performance of scalable Naïve Bayes classifier (SNB) on Beyond MapReduce and traditional Naïve Bayes classifier are also compared over different data sets. The experimental results show that SNB on Beyond MapReduce is more suitable to classify massive datasets in distributed computing environment and it provides a better accuracy and minimal processing time than traditional Naïve Bayes classifier.

Keywords- Apache Spark, Beyond MapReduce, HDFS, Mahout Samsara, Massive data, Naïve Bayes Classifier

1. Introduction

The scale of data is increasing overwhelmingly during the past decades. The IBM Data Flood Infographic shows that 2.5 quintillion bytes of data are created every day [1]. Traditional data mining algorithms are not well suited to process the full value of massive data. Therefore, a massively scalable and parallel algorithm is needed to process and analyze such datasets. For processing huge amount of data, Hadoop is becoming the core technology to solve the huge data problems for large organizations with cloud storage. A commonly used architecture for Hadoop consists of Hadoop Distributed File System (HDFS) for massive data storage and MapReduce for distributed parallel computing.

Machine learning is also ideal for exploiting the opportunities hidden insight in massive data. Apache Mahout is an open source machine learning library built on top of Hadoop to provide distributed analytics capabilities. Although it was originally developed with MapReduce based algorithm, MapReduce was inefficient for most of the scalable machine learning that Mahout pioneered because of its limitation. Also, alternative frameworks such as Spark have finally become much more viable.

Spark absorbs the advantages of Hadoop MapReduce, unlike MapReduce, the intermediate and output results of the Spark jobs can be stored in memory, which is called Memory Computing. Memory Computing improves the efficiency of data computing. So, Spark is better suited for iterative applications, such as Data Mining and Machine Learning [2].

Starting from the release 0.10.0, a new generation of mahout was born for building backend independent programming environment, also called the code name, "Samsara". Mahout Samsara is backend-agnostic and uses a Scala-based programming environment to support writing parallel mathematical languages.

This paper aims to analyze the performance and scalability of SNB on Beyond MapReduce for the massive data classification. The remainder of this paper is organized as follow. Section 2 describes the related work of this paper and section 3 introduces some background information. And then architecture of SNB on beyond MapReduce is presented in section 4. After that the performance evaluation and result discussions are shown in section 5. Finally, the conclusion of this paper is summarized in section 6.

2. Related Work

It has become difficult to process massive amount of data with rapid growth in internet although traditional data mining techniques have achieved good performance for small amount of data scale on single machine. A large number of approaches have been proposed to solve this issue. Most of them based on MapReduce model and distributed file system. The authors [3] proposed Map Reduce-based Naïve Bayes classifier (MP-NB) to process CRIA (Customer Requirement Information Acquisition) classification on large scale mobile data. MP-NB was implemented on

Hadoop Platform with MapReduce programming model to solve data-intensive computing problems.

To analyze the enormous data set, the authors[4] proposed an improved Naïve Bayes classifier for large-scale text classification. They used MapReduce programming model to improve the accuracy of proposed classifier and to provide good scalability and extensibility for text classification.

The main challenges of becoming massive data are high dimensionality, diversity and high analysis value return. For processing massive data, the authors [5] proposed a scalable random forest algorithm based on MapReduce (SMRF). They showed that their new algorithm, SMRF is more suitable to classify massive data sets in distributed computing environment than traditional Random Forest algorithm. Although SMRF algorithm had higher performance while comparing with traditional Random Forest algorithm, it had the equally accuracy degradation because of using MapReduce. MapReduce is inefficient for application that share data across multiple steps, including iterative algorithms or iterative queries.

This paper intends to analyze the performance and scalability of SNB for the massive data classification. By using SNB, it performs efficiently processing on massive datasets in distributed environment and provides good performance results than traditional Naïve Bayes classifier.

3. Preliminaries

This section provides the preliminaries of this paper. First, Apache hadoop and the Spark framework are introduced in Section 3.1 and 3.2. Then Apache Mahout and Mahout Samsara are presented in Section 3.2 and 3.4.

3.1. Apache Hadoop Framework

Apache Hadoop[6] is an open-source software framework for storing and processing large data in a distributed manner. It supports data intensive distributed applications on large clusters built of commodity hardware. This framework is designed to be scalable, which allows the user to add more nodes as the application requires. Hadoop consists of the Hadoop Distributed File System for big data storage and MapReduce engine for distributed processing. Hadoop cluster consists of a single master node and many worker nodes. HDFS is based on master/slave architecture. The development of Hadoop-based data mining techniques has been widely spread, because of its fault-tolerant mechanism and its ease of use.

Despite its popularity, MapReduce becomes inefficient to develop some scalable machine learning

algorithms because of its limitations. Since MapReduce is only suitable for batch processing job, implementing interactive jobs and model become expensive due to the huge space consumption by each job. This costs much time for disk I/O operations and also massive resources for communication and storage. To overcome this problem, many distributed processing framework are emerged.

3.2 Spark Framework

Apache Spark [7] is a fast and general engine for large-scale data processing. It has an advanced DAG (Directed Acyclic Graph) execution engine that supports acyclic data flow and in-memory computing. The core abstraction of Spark is Resilient Distributed Dataset (RDD) which has a better ability of computing and fault tolerance. So, Spark can allow us to store a data cache in memory, to perform computation and iteration for the same data directly from memory. It saves huge amount of disk I/O operation time. Therefore, it is more suitable for developing scalable machine learning algorithms.

3.3. Apache Mahout

Apache Mahout [8] is an environment for creating scalable, performant, machine learning applications. It is a machine learning library that runs over the hadoop system for solving clustering, and classification problems. It born a new generation of mahout, linear algebra environment, known as the code name “Mahout Samsara”(release 0.10.0 or later). MapReduce was not a very good fit for most of the machine learning that mahout pioneered. In place of Hadoop MapReduce, Mahout has been focusing on implementing flexible and backend-agnostic machine learning environment.

3.4 Mahout Samsara

Mahout Samsara is a new generation of mahout. It is also known as “Beyond MapReduce” because it is the part of mahout that deals with more advanced backends, post-mapreduce generation: Spark, Flink, and H₂O. These backends extend the set of distributed paradigms beyond just MapReduce. Therefore, machine learning algorithms built with the mahout Samsara DSL are better served for iterative nature of applications.

4. Scalable Naïve Bayes Classifier (SNB) on Beyond MapReduce

Naïve Bayes algorithm is a popular algorithm in text classification field. It is a simple probabilistic classifier based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. Samsara currently has two flavors of Naïve Bayes implemented in its distribution. The first is standard Multinomial Naïve Bayes ("MNB" or "Bayes") and the latter is a variation on Transformed Weight-normalized Complement Naïve Bayes ("TWCNB" or "CBayes") [9]. In this paper, Samsara's MNB is used for massive data classification in distributed environment. Figure 1 shows a SNB on Beyond MapReduce Architecture.

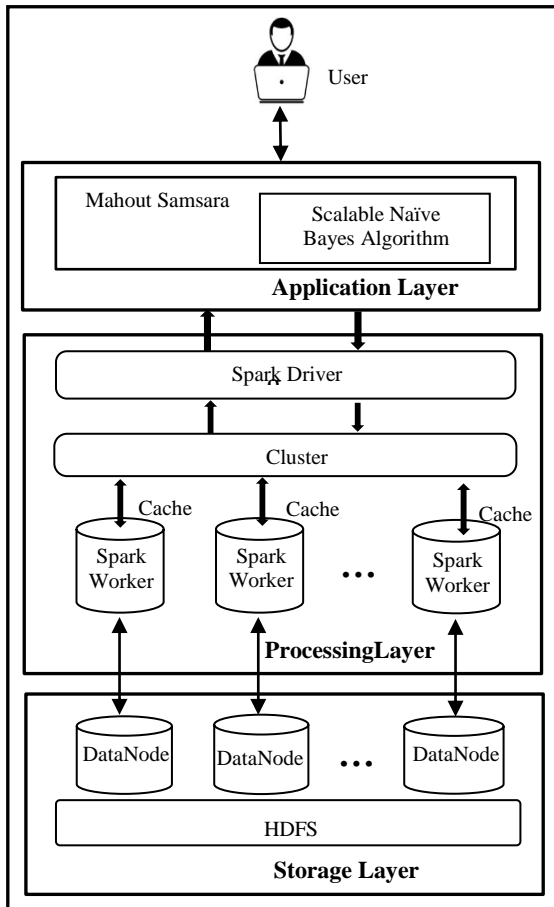


Figure 1. SNB on Beyond MapReduce Architecture

4.1 SNB for Text Classification

The structure of the SNB is shown in Figure 2. It is divided into three stages: initializing, training and label assignment. In the transformation stage, the dataset is acquired and vectorized the document. In order to

make good use of the computing resources, Hadoop cluster is implemented. Hadoop is a framework that admits the data in sequence file format. Mahout Samsara over hadoop also admits the data in the sequence file directory. The input data is converted to sequence file format to parse the <text> element of each document. After taking the sequence file conversion phase, the documents are vectorized using *mahout seq2sparse*. The command *seq2sparse* converts the sequence file directory to vector format. The sequence file will be accepted as input and produce the output as vector using a weighting factor like TF-IDF (Term frequency-Inverse Document Frequency) scheme.

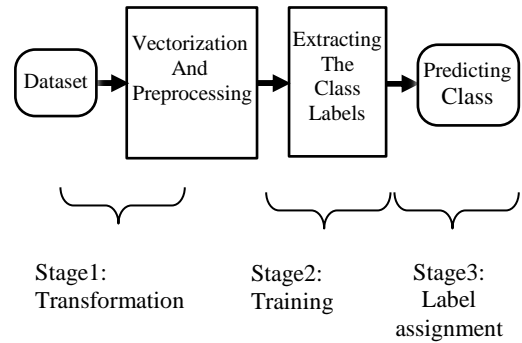


Figure 2. The Structure of SNB

In the training stage, SNB uses the Spark random Split API to split the training and testing sets. Since Spark backend environment is chosen, the algorithm in Mahout Samsara can take advantages of Spark native function. SNB stores the class label of each vectorized document as the row keys. And then, it extracts the all possible document identifier for each document.

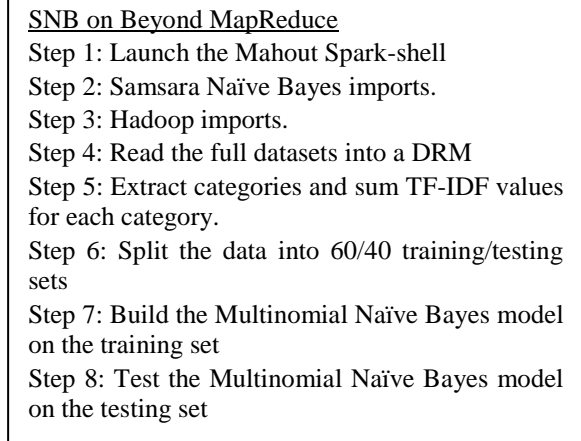


Figure 3. SNB on Beyond MapReduce

In the label assignment stage, SNB assigns a label to a vectorized document using a classification function.

It predicts a classification of the document by assigning a class with the largest posterior probability. The SNB on Beyond MapReduce with term weighting scheme is described in figure 3.

5. Performance Evaluation

This section analyzes the performance of SNB on Beyond MapReduce. Firstly, the performance of SNB is compared with traditional Naïve Bayes classifier (NB) in terms of accuracy. And then the scalability of SNB is tested on distributed nodes. The effectiveness in classification for the performance analysis of SNB will be evaluated using confusion matrix which records correctly and incorrectly recognized examples for each class. The four matrices of performance that measure the classification quality for the positive and negative classes independently are:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$\text{TP}_{\text{rate}} = \frac{TP}{TP+FN} \quad (2)$$

- **True Positive rate** is the percentage of positive cases correctly classified as belonging to the positive class.

$$\text{TN}_{\text{rate}} = \frac{TN}{FP+TN} \quad (3)$$

- **True Negative rate** is the percentage of negative cases correctly classified as belonging to the negative class.

$$\text{FP}_{\text{rate}} = \frac{FP}{FP+TN} \quad (4)$$

- **False Positive rate** is the percentage of negative cases misclassified as belonging to the positive class.

$$\text{FN}_{\text{rate}} = \frac{FN}{TP+FN} \quad (5)$$

- **False Negative rate** is the percentage of positive cases misclassified as belonging to the negative class.

5.1. Experimental Environment

This section performs comprehensive experiment to compare accuracy level of SNB on Beyond MapReduce with the traditional multinomial Naïve Bayes algorithm (NB). All the experiments have been carried out over a Hadoop cluster of three computing nodes having high configuration machine of Intel CORE i7 processor, 8 GB of RAM, 1 TB HDD on Linux Ubuntu-16.04 system. The computer cluster is set up with Hadoop,

one name node and three data nodes. The specific details of the software used and its configuration are open-sour Apache Hadoop distribution (hadoop 2.6.0), apache spark (1.5.2), maven (3.3.3), scala version (2.10.4), java version (jdk-7.79) and the latest release of Mahout Samsara (0.12.3). The descriptions of datasets used in this paper are presented in table 1.

Table 1. Experimental Datasets

Dataset	Description
D1	Breast-cancers [10] -Wisconsin Diagnostic Breast Cancer (WDBC) dataset -Number of instances : 569 -Number of attributes : 32
D2	Iris [10] -Number of instances : 150 -Number of attributes : 4
D3	Adult [10] -Number of instances : 48842 -Number of attributes : 14
D4	Movie Reviews [11] - Rotten Tomatoes movies review dataset -contain 1000 positive and 1000 negative reviews
D5	Reuter-21578[12] - newswire dataset -contain 21578 newswire documents
D6	OHSUMED[13] - medical abstract dataset -contain 348,566 references and 6 categories

5.2. Performance Evaluation and Result Discussion

SNB is performed in Hadoop cluster distributed computing environment. Weka workbench [14] is adopted to run traditional multinomial Naïve Bayes algorithm (NB). To compare the two algorithms with different datasets, the datasets are partitioned into 60/40 ratio for training and testing. The accuracy measurement of each dataset is shown in Figure 4.

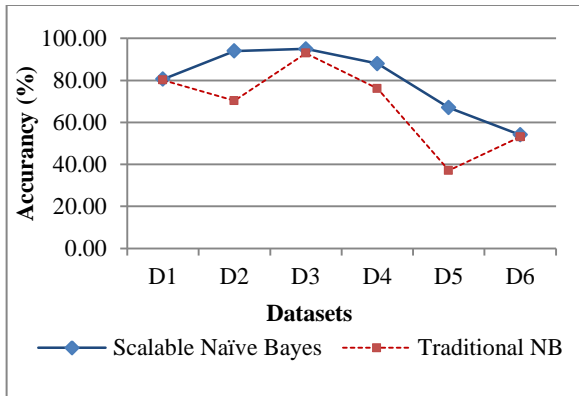


Figure 4. SNB vs. NB

According to the comparison result, the accuracies of SNB in data set D2 and D3 are almost 94% and 95% respectively, which are much higher than NB. In D1 and D4 also, its accuracies achieved beyond 80% and 88% which is 1.4% and 12% larger than NB. In D5 and D6, the two classifiers obtained equally accuracy degradation because this datasets contained more instances from one class than from the other. Therefore, the class imbalanced problem becomes a challenge for massive data mining. However, it can be noted that SNB achieved the better accuracy on six datasets than traditional Naïve Bayes classifier.

5.3. Scalability Test

SNB is parallelized on the distributed hadoop cluster environment. Mahout Spark-shell is used for unseen data during the initial vectorization of the training and testing sets. This section performs the scalability test of SNB with different computing nodes. It analyses the scalability of SNB using Enron's email spam dataset in term of accuracy and processing time. In order to verify the scalability of SNB on Beyond MapReduce, the "Enron email" datasets are enlarged according to its formats. The number of ham and spam emails in each Enron dataset is summarized in Table 2.

Table 2. Summary of Enron Email Datasets

Enron Dataset	Number of mails	
	Spam	Ham
E 1	8996	13545
E 2	12671	15045
E 3	17171	16545
Total number of mails		83973

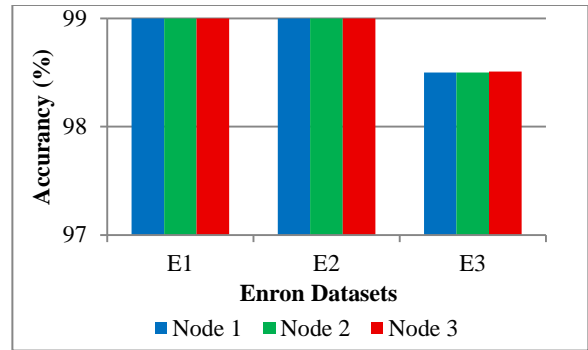


Figure 5. SNB with Different Node Number Classification Result

Figure 5 shows the accuracy level of SNB model on 1, 2 and 3 Data Nodes of hadoop cluster computing environment. According to this result, SNB provides the good accuracy results beyond the 98% with different datasets on each data node.

Figure 6 shows the total processing time of SNB on distributed hadoop cluster environment. For processing massive data, multiple data nodes are simultaneously parallelized. It reduces the processing time to minimal and provides the fast distributed computing. As a result, SNB running with 3 data nodes can provide faster computing time in most dataset classifications.

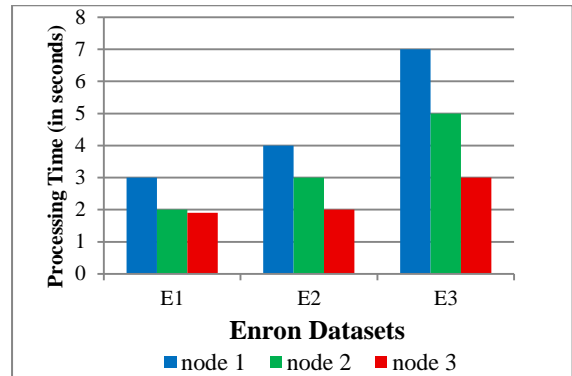


Figure 6. Comparison of Processing Time of SNB

According to the results, the scalable Naïve Bayes classifier on Beyond MapReduce (SNB) can provide a good scalability for massive data classification.

6. Conclusion

Scalability has become one of the core concept slash buzzwords of massive data. Massive data requires a scalable and parallel machine learning algorithm for efficiently processing. Analytical processing complexities are required to take minimal time to get results. In this paper, the performance of a scalable naïve Bayes classifier on Beyond MapReduce is

analyzed. SNB can process the massive data and run multiple tasks simultaneously. Our comparative study can show that SNB on Beyond MapReduce has a better accuracy and faster processing time than the traditional Naïve Bayes algorithm. It also provides the good scalable performance on distributed environment. As future work, the issues in classification of massive datasets with skew class distribution will be considered. Regarding the performance analysis of SNB on Beyond MapReduce, techniques to deal with extremely class imbalanced problem will be studied.

7. References

- [1] Big Data Flood Infographic. [Online] Available: <https://www.ibm.com/>
- [2] J. Fu, J. Sun, K. Wang, “SPARK—A Big Data Processing Platform for Machine Learning”, in 2016 IEEE International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration , 2016, pp. 48-51
- [3] X. Wang, B. Sheng, L. Xue, Z. Xiao, “Classification of Customer Requirements on Map Reduce-based Naïve Bayes”. In 2016, IEEE international Conference on Big Data Analysis (ICBDA). IEEE 2016
- [4] C. Huaixin, “An improved Naive Bayes Classifier for Large Scale Text”. In Proceedings of the 14th International Conference on Applications of Computer Engineering (ACE '15), Seoul, South Korea September 5-7, 2015
- [5] J. Han, Y. Liu, X. Sunl, “A Scalable Random Forest Algorithm Based on MapReduce”. In 2013,4th IEEE international conference on ICSESS, page 849-832. IEEE 2013
- [6] Apache Hadoop. [Online] Available: <http://hadoop.apache.org/>
- [7] Apache Spark. [Online] Available: <http://Spark.apache.org/>
- [8] Apache Mahout. [Online] Available: <http://mahout.apache.org/>
- [9] Dmitriy Lyubimov, Andrew Palumbo. *Apache Mahout: Beyond MapReduce*. 2016
- [10] A. Frank and A. Asuncion, “UCI machine learning repository”, 2010. [Online] Available: <http://archive.ics.uci.edu/ml>
- [11] Movie Review Data. [Online] Available: <https://www.kaggle.com/c/sentiment-analysis-on-movie-reviews>
- [12] Reuters-21578 collection Apte' split. [Online] Available: <http://disi.unitn.it/moschitti/corpora.htm>
- [13] Ohsumed collection. [Online] Available: <http://disi.unitn.it/moschitti/corpora.html>
- [14] Weka tool. [Online] Available: <http://www.filehorse.com/download-weka-64/>

Cloud Based Big Data Application of FP-Growth Algorithm and K-Means Clustering Algorithm Based on MapReduce Hadoop

Than Htike Aung, Nang Saing Moon Kham
thanhtikeaung@ucsy.edu.mm
moonkhamucsy@ucsy.edu.mm

Abstract

In current time large volumes of data are being produced by various modern applications at an ever increasing rate. These applications range from wireless sensors networks to social networks. The automatic analysis of such huge data volume is a challenging task since a large amount of interesting knowledge can be extracted. Association rule mining is an exploratory data analysis method able to discover interesting and hidden correlations among data. Since this data mining process is characterized by computationally intensive tasks, efficient distributed approaches are needed to increase its scalability. This paper proposes a cloud-based service, named parallel FP-growth, to efficiently mine association rules on a distributed computing model. It consists of a series of distributed MapReduce jobs run in the cloud. Each job performs a different step in the association rule mining process, followed by cloud-based parallel k-means clustering algorithm to produce similar groups. These outputs are verified and filtered by three conditional levels which results in useful rules.

As a case study, the proposed approach has been applied to the educational data scenario.

Keywords- association rule mining, distributed computing model, cloud-based service, network data analysis.

1. Introduction

Cloud computing and Data Mining are both interesting hot topics nowadays. They bring convenience to diverse fields, including education. In general, clustering and prediction are two of the most remarkable features of data mining techniques. Unlike traditional analytical methods, data mining could offer more individual-oriented results. The application of data mining techniques in the education field enables numerous possibilities such as comprehensively analyzing the characteristics of each of students, predicting success in classes, pinpointing the gifted students and their learning paths [4] etc. In the higher education field, data mining applications have been highly suggested by many researchers such as C Romero and S Ventura in [5] and Luan in [9] to modify or design the curriculum to meet the different needs of students in terms of the learning abilities and knowledge construction.

By these technologies, potentially valuable rules from educational data can be obtained for making decisions and strategies that can optimize the educational resource. In this project, the Jiaqu Yi et al.[8] proposed a cloud-based framework to generate the rules. Inside the framework, a parallel FP-growth association algorithm is adopted to generate useful rules among the students' grades, followed by reasonable analysis on the generated rules. All the analysis is based on learning skills identification for individual courses and a MapReduce-based K-means clustering algorithm on Hadoop. The experimental data come from the University of Computer Studies, Yangon in Myanmar (UCSY). The historical data can be increased as three Vs for educational schools. Thus the data are considered on the following conditions.

Volume: It is the most visible and major issue of concern referring to the fact that the amount of generated data has increased tremendously in past years. The increase in internet users has increased in the global data production. Organizations are overflowed with data, unmanaged hundreds of terabytes and petabytes of information.

Variety: With the tremendous growth in data sources there are different types of data which need analysis. Extends beyond structured data it includes semi-structured or unstructured data of all varieties, such as text, audio, video, web pages, log files and more.

Velocity: more and more data is generated and is provided to the users immediately whenever required. This aspect captures the growing data rates. Millions of connected devices i.e. Smartphone, tablets etc. increase not only volume but velocity also. Data is a rapid increase in rate of data transmission.

2. Related Work

There are many research papers about educational data mining, which are published in authoritative journals and conferences. For instance, Richard A. Huebner presents a survey of educational data mining research in [19]. In [19], Huebner describes how data mining can be utilized to analyze the data captured from course management systems. [21] Proposes a simple and efficient k-means clustering algorithm which requires a kd-tree as the only major data structure. Ramli, A.A adopts an Apriori algorithm to improve the content of learning portal [18]. Minaei-bidgoli, B Tan, P, Punch, W reveal interesting

association rules among the attributes from students and problems in order to optimize online education systems [13]. Merceron, A, Yacef, K. utilize association rules to process learning data and find out whether students use resources to enhance grade and whether their use of such resources affects their grades [12].

3. Methodology

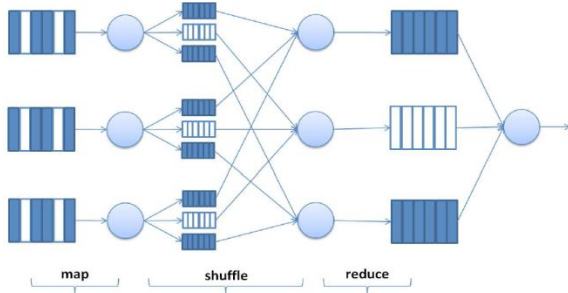


Figure 1. Typical composition of the MapReduce function.

MapReduce was originally designed and adopted by Google as a programming model for processing large data sets on a cluster with parallel processing over distributed storage.

The MapReduce paradigm now has become an industry standard and many platforms are internally built on this paradigm and support MapReduce implementation. Hadoop is an open source implementation that can be run either in-house or on cloud computing services with elastic MapReduce.

This has, at the core, the Map and Reduce functions that are capable of running in parallel across the nodes in the cluster. The Map function works on the distributed data and runs the required functionality in parallel, and the Reduce function runs a summary operation of the data.

In this section describe the data mining technique used algorithms which is MapReduce based parallel k-means algorithm and FP-Growth algorithm. These two algorithms used cloud-based framework. These two original algorithms can do in traditional data mining but also used for small and median data size.

3.1. ParallelFP-Growth on MapReduce

Nowadays, parallel distributed computing technology is quite mature, resulting in the emergence of cloud computing recent years. Cloud computing can take advantage of MapReduce and increase computing effectiveness by flexibly deploying online servers. However, the FP-Growth algorithm cannot be decomposed into multiple subtasks to facilitate distributed processes, and it also needs to scan databases twice in

order to construct an FP-Tree. Xiaoting et al. [11] proposed a parallel version of FP-Growth called the Paralleled Incremental FP-Growth, (PIFP-Growth) algorithm, shown in Figure 2. In the proposed algorithm, MapReduce executes the FP-Growth algorithm each time new data arrive, which solves the problem of incremental database increases by dynamic threshold value comparison, and also avoids the double-computing problem. However the PIFP-Growth algorithm does not solve the problem faced by the FP-Growth algorithm of not being decomposable into multiple subtasks.

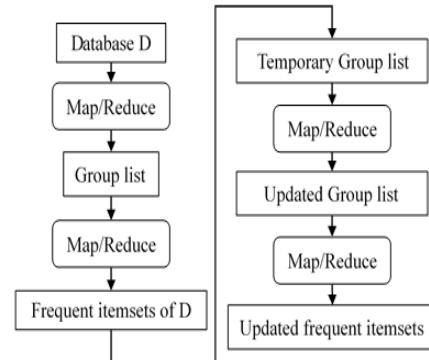


Figure 2. The PIFP-Growth algorithm

3.2. PKMeans Based on MapReduce

As the analysis above, PKMeans algorithm needs one kind of MapReduce job. The map function performs the procedure of assigning each sample to the closest center while the reduce function performs the procedure of updating the new centers. In order to decrease the cost of network communication, a combiner function is developed to deal with partial combination of the intermediate values with the same key within the same map task.

3.2.1. Map-function The input dataset is stored on HDFS[11] as a sequence file of <key, value> pairs, each of which represents a record in the dataset. The key is the off set in bytes of this record to the start point of the data file, and the value is a string of the content of this record. The dataset is split and globally broadcast to all mappers. Consequently, the distance computations are parallel executed. For each map task, PKMeans construct a global variant centers which is an array containing the information about centers of the clusters. Given the information, a mapper can compute the closest center point for each sample. The intermediate values are then composed of two parts: the index of the closest center point and the sample information. The pseudocode of map function is shown in Algorithm 1.

Algorithm 1. map (*key*, *value*)

Input: Global variable *centers*, the offset *key*, the sample *value*

Output: <*key'*, *value'*> pair, where the *key'* is the index of the closest center point and *value'* is a string comprise of sample information

1. Construct the sample *instance* from *value*;
 2. *minDis* = Double.MAX_VALUE;
 3. *index* = -1;
 4. For *i*=0 to *centers.length* do
 dis = ComputeDist(*instance*, *centers*[*i*]);
 If *dis* < *minDis* {
 minDis = *dis*;
 index = *i*;
 }
 }
 5. End For
 6. Take *index* as *key'*;
 7. Construct *value'* as a string comprise of the values of different dimensions;
 8. output <*key'*, *value'*> pair;
 9. End
-

Note that Step 2 and Step 3 initialize the auxiliary variable *minDis* and *index* ; Step 4 computes the closest center point from the sample, in which the function ComputeDist (*instance*, *centers*[*i*]) returns the distance between *instance* and the center point *centers*[*i*]; Step 8 outputs the intermediate data which is used in the subsequent procedures.

3.2.2. Combine-function. After each map task, we apply a combiner to combine the intermediate data of the same map task. Since the intermediate data is stored in local disk of the host, the procedure cannot consume the communication cost. In the combine function, we partially sum the values of the points assigned to the same cluster. In order to calculate the mean value of the objects for each cluster, we should record the number of samples in the same cluster in the same map task. The pseudocode for combine function is shown in Algorithm 2.

Algorithm 3. reduce (*key*, *V*)

Input: *key* is the index of the cluster, *V* is the list of the partial sums from different host

Output: <*key'*, *value'*> pair, where the *key'* is the index of the cluster, *value'* is a string representing the new center

1. Initialize one array record the sum of value of each dimensions of the samples contained in the same cluster, e.g. the samples in the list *V*;
 2. Initialize a counter *NUM* as 0 to record the sum of sample number in the same cluster;
 3. while(*V.hasNext*()) {
 Construct the sample *instance* from *V.next*();
 Add the values of different dimensions of *instance* to the array
 NUM += *num*;
 4. }
 5. Divide the entries of the array by *NUM* to get the new center's coordinates;
 6. Take *key* as *key'*;
 7. Construct *value'* as a string comprise of the *center*'s coordinates;
 8. output <*key'*, *value'*> pair;
 9. End
-

3.2.3. Reduce-function. The input of the reduce function is the data obtained from the combine function of each host. As described in the combine function, the data includes partial sum of the samples in the same cluster and the sample number. In reduce function, we can sum all the samples and compute the

total number of samples assigned to the same cluster. Therefore, we can get the new centers which are used for next iteration. The pseudocode for reduce function is shown in Algorithm3.

Algorithm 2. combine (*key*, *V*)

Input: *key* is the index of the cluster, *V* is the list of the samples assigned to the same cluster

Output: <*key'*, *value'*> pair, where the *key'* is the index of the cluster, *value'* is a string comprised of sum of the samples in the same cluster and the sample number

1. Initialize one array to record the sum of value of each dimensions of the samples contained in the same cluster, i.e. the samples in the list *V*;
 2. Initialize a counter *num* as 0 to record the sum of sample number in the same cluster;
 3. while(*V.hasNext*()) {
 Construct the sample *instance* from *V.next*();
 Add the values of different dimensions of *instance* to the array
 num++;
 4. }
 5. Take *key* as *key'*;
 6. Construct *value'* as a string comprised of the sum values of different dimensions and *num*;
 7. output <*key'*, *value'*> pair;
 8. End
-

4. Learning Skill Test Metrics

Table 1 shows practical and tutorial scores in each subject. Three requirement levels indicate monthly test of score for students to be became skill full students in every academic subjects and courses. This is to measure the learning needs of the students based on the level of each subjects of practical /tutorial scores.

Table1. Details of practical and tutorial scores in each subjects

Requirement Level	Practical / Tutorial Score
1	1-4
2	5-7
3	8-10

4.1. Experimental Result

Propose System design used cloud platform. Mining withParallel FP growth algorithm produced association rule from input data from cloud resources and then k-means clustering algorithm produced three clusters which is satisfy with learning skill test matrices in Table 1 . Finally rule verification and filtering on each cluster. The resulted data are useful rules shows in section 4.7.

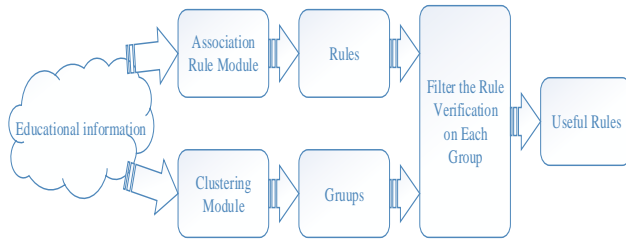


Figure 3. Research Framework

4.2. Raw Data

The raw data which contains students' grade from UCSY is shown in Figure 4. The data contains student ID, course and subject marks. All the observed students are from the same major. In total, there are 7000 lines of student information in the raw data, the five previous year data. All this data is stored in a CSV document.

	A	B	C	D	E	F	G	H
1	Student_i	Student_r	Major	Course_id	Subjects_nam	Subject_s	Grade	Address
2	10001	Ei Ei Mon	Computer M	MYANMAR	56	B	Yangon	
3	10001	Ei Ei Mon	Computer E	IELTS TEST BUI	67	B	Yangon	
4	10001	Ei Ei Mon	Computer P	PHYSICS	80	A	Yangon	
5	10001	Ei Ei Mon	Computer CST101	INTRODUCTIO	56	B	Yangon	
6	10001	Ei Ei Mon	Computer CST102	MATHEMATIC	50	B	Yangon	
7	10001	Ei Ei Mon	Computer CST103	COMPUTER AF	40	C	Yangon	
8	10001	Ei Ei Mon	Computer CST104	PROGRAMMIN	41	C	Yangon	

Figure 4. The Raw Data

4.3. Data Pre-processing

In the raw data, some attributes such as gender and hometown are unnecessary, and some records are repetitive and redundant. Hence, the data needs to be cleaned up. And stored into MySQL. After preprocessing, the clean data generated is shown in Figure 5.

	A	B	C	D	E	F
1	Student_i	Major	Course_id	Subjects_nam	Subject_s	Grade
2	10001	Computer M	MYANMAR	56	B	
3	10001	Computer E	IELTS TEST BUI	67	B	
4	10001	Computer P	PHYSICS	80	A	
5	10001	Computer CST101	INTRODUCTIO	56	B	
6	10001	Computer CST102	MATHEMATIC	50	B	
7	10001	Computer CST103	COMPUTER AF	40	C	
8	10001	Computer CST104	PROGRAMMIN	41	C	

Figure 5. The Clean Data

4.4. Data Mining

Figures stored all candidate 1-itemsets and record the support count of each 1-itemset. The minimum support condition is set to 0.1 which means 10% of the total amount (the total amount is more than 300). The minimum confidence condition is set to 0.7 which is the

conditional probability between two frequent itemsets. A part of C1 and L1 generated by parallel FP-growth algorithm is displayed in Table 2. and Table 3. respectively. Those candidates whose support counts are lower than the support condition will be pruned.

Table 2. Candidate 1-Items C1

Itemset	Amount
CST103:ICS:A	0
CST103:ICS:B	24
CST103:ICS:C	72
CST103:ICS:D	75
CST103:ICS:E	8
CST104:PL:A	0
CST104:PL:B	8
CST104:PL:C	42
CST104:PL:D	38
CST104:PL:E	14
.....

Table 3. Frequent 1-Itemstes L1

Itemset	Amount
CST103:ICS:C	72
CST103:ICS:D	75
CST104:PL:C	42
CST104:PL:D	38
CST102:Mths:A	37
CST102:Mths:B	49
P:Phy:A	41
P:Phys:B	37
E:IELTS:B	59
E:IELTS:C	87
.....

Table 4. Result of Association Rules

Rules	Confidence
CST102:Mths:C ⇒ P:Phy:C	0.73
CS202:Mths:C ⇒ CS203:DS:C	0.8
CS202:Mths:C ⇒ 203:DL:C	0.85
CST102:Mths:C ⇒ CST103:PL:C	0.73
CS206:SE:B ⇒ CS203:DS:C	0.81
CS204:SAD:B ⇒ CS201:JAVA:A	0.7
CS404:DBMS:B ⇒ CS405:UML:C	0.72
CS201:JAVA:A ⇒ CS203:DS:C	0.71
CS201:JAVA:A ⇒ CS202:Mths:C	0.71
CS304:UML:A ⇒ CS305:CAT3:C	0.71
CS206:SE:B ⇒ CS204: SAD:B	0.76
CS304:DBMS:B ⇒ CS301:CO:C	0.7
..

Significant trends can be seen from the results of FP-growth algorithm. For example, "CS204:SAD:B ⇒ CS201:JAVA:A Confidence = 0.70 ", it can be said that if a student gets B level in SAD subject, he will excel in course CS201, it means course CS204 may requires some

similar skills as course CS201. In another word, students who are able to perform well in both course CS204 and CS201, certain parts of his/her learning skills are better than the rest of the students s’ sample, vice versa. Another example, "304:DBMS:B =>301:APP:C Confidence=0.70", it tells us that a student gets C level in APP, he will probably get a bad level in course CS304:DBMS. That is very likely to say, there may be some similarity on the knowledge or skills requirement between course CS301 and course CS304. Students who cannot get a good level in both of the courses, he/she might be lacking of certain knowledge or learning skills, vice versa. According to above analysis, that can realize further.

4.5. Experimental Result of K-Means Clustering Algorithm

The learning skills test metrics presented in section 4 which are used by K-Means clustering algorithm. The courses are evaluated with this three-level test score based on the learning skill test matrices. Thus the output result produced three levels of test scores groups.

4.6. Experimental Raw Data and Results

An example of evaluated results is shown as Figure 6. The scores for the three levels range from 1 to 3. “1” indicates that this ability is highly desired, and “3” means that the corresponding ability is not required by this course. This data is input into "Clustering Module". Inside "Clustering Module" the courses are cluster via K-Means clustering algorithm. Courses which are similar according to the three level conditions will be grouped together. To demonstrate the results, the clustering result is shown visually as Figure 7. In Figure 7 is show that courses are divided into 3 groups/clusters by K-Means algorithm. In each cluster, courses are shown as a smallest circle with their course ID on it. The details in cluster 2 are shown in Figure.5. Among all these clusters, cluster 3 represents the courses which require all high level condition from those two kinds (2;3). Cluster 1 stands for the courses which not require high level condition (3).

	A	B	C	D
1	Student_i	Course_id	Require	Practical
2	10001	M	2	7
3	10001	E	2	6
4	10001	P	3	8
5	10001	CST101	2	7
6	10001	CST102	2	7
7	10001	CST103	3	9
8	10001	CST104	1	4

Figure6. Example of Practical/Tutorial Test Score

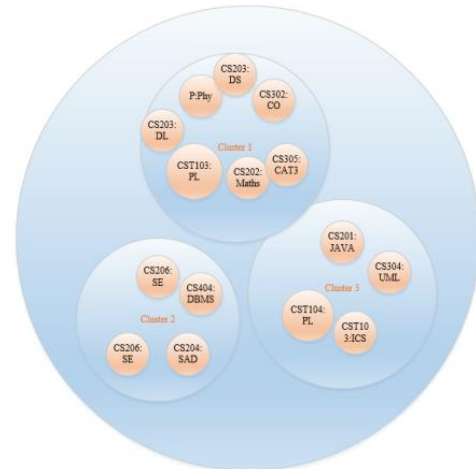


Figure7. Result of Clustering Module

In the following section that will analyze the results of “Rule Abstraction Module” by the results from “Clustering Module”.

4.7. Rules Verification and Filtering

4.7. 1. Cluster 2 and Cluster 3

If a student got a B level in course from cluster 2, he/she was likely to get a A in most of the cluster 3 courses. Courses such as “JAVA”, “SAD” etc. belong to cluster 3. Students who have level 2,3 condition he/she has good practice performance in their subjects such as JAVA and SAD. effected directly. Similar result can be found in the relationship between cluster 2 and cluster 3

4.7. 2. Cluster 1

If a student got a C in most of the cluster 1 courses, he/she also has level 1, at least who require strong level attitude level 2.

5. Conclusions

Data, classification and clustering are an important task in parallel computing and distributed computing, when the processed dataset is large-scale, it becomes even more important. In this paper introduce a parallel FP-Growth algorithm and K-Means clustering algorithms by using MapReduce and Hadoop, which improves the application of big data research in education field to produce more rules. The result also provides a good reference for education for students learning needs.

In particular, future works will aim at optimizing the MapReduce workflow and combining the workbench of the cluster and classification architecture.

5. References

- [1] Abdullah Alshwaier , Ahmed Youssef and Ahmed Emam, "A Newtrend for E-Learning in Ksa Using Educationalclouds", Advanced Computing : an International Journal, 2012,Vol.3(1), p.81
- [2] Agrawal, R.; Imieli_ski, T.; Swami, A. (1993). "Mining association rules between sets of items in large databases". Proceedings of the 1993 ACM SIGMOD international conference on Management of data – SIGMOD '93. p. 207.
- [3] American Bar Association Section of Legal Education and Admissions to the Bar, Legal Education and Professional Development –An Educational Continuum Report of The Task Force on Law Schools and the Profession: Narrowing the Gap, July, 1992.
- [4] Bael, S., S. H. Hat, and S. C. Parka. "Identifying gifted students and their learning paths using data mining techniques." Data Mining in ELearning (Advances in Management Information) 4 (2006): 191-205.
- [5] C Romero, S Ventura, Data mining in e-learning. WIT, 2006.
- [6] David Chappell, (October 2008). " Introducing the Azure Services Platform An Early look at Windows Azure, .Net Services, SQL Services, And Live Services ". Chappell & Associates.157
- [7] Fayyad, Usama; Piatetsky-Shapiro, Gregory; Smyth, Padhraic (1996)."From Data Mining to Knowledge Discovery in Databases". Retrieved 17 December 2008.
- [8] Jiaqu Yi, Sizhe Li, Maomao, Wu, H.H. Au Yeung Wilton W.T Fok, Ying Wang, Fang Liu "Apriori algorithm and K-Means Clustering algorithm based on Students' Information", 2014 IEEE Fourth International Conference on Big Data and Cloud Computing
- [9] Luan, Jing. "Data mining and its applications in higher education." Newdirections for institutional research 2002.113 (2002): 17-36.
- [10] MacKay, David (2003). "Chapter 20. An Example Inference Task: Clustering". Information Theory, Inference and Learning Algorithms. Cambridge University Press
- [11] MacQueen, J. B. (1967). "Some Methods for classification and Analysis of Multivariate Observations". Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press.1-8.
- [12] Merceron, A., Yacef, K. (2008). Interestingness Measures for Association Rules in Educational Data. In International Conference on Educational Data Mining, Montreal, Canada, 57-66.
- [13] Minaei-bidgoli,B Tan, P., Punch, W. (2004). Mining interesting contrast rulesfor a web-based educational system. In International Conference on Machine Learning Applications, Los Angeles, USA.
- [14] M.Lawanya Shri, Dr. S.Subha, "An Implementation of e-Learning System in Private Cloud", International Journal of Engineering and Technology, 2013, Vol.5(3), p.3036
- [15] Paul Pocatilu. "Cloud Computing Benefits for E-learning Solutions".Oeconomics of Knowledge, 2010, Vol.2(1), p.9
- [16] Peden, Elisabeth; Riley, Joellen, "Law Graduates Skills A Pilot Study into Employers Perspectives" [2005] LegEdRev 5; (2005) 15(1&2)Legal Education Review 87.
- [17] Rakesh Agrawal and Ramakrishnan Srikant Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, pages 487- 499, Santiago, Chile, September 1994]
- [18] Ramli, A.A. (2005). Web usage mining using apriori algorithm: UUM learning care portal case. In International conference on knowledge management, Malaysia, 1-19.
- [19] Richard A. Huebner, "A survey of educational data-mining research",Research in Higher Education Journal, Retrieved 30 March 2014
- [20] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D.Piatko, Ruth Silverman, Angela Y. Wu, "An Efficient k-MeansClustering Algorithm: Analysis and Implementation", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 24, NO. 7, JULY 2002
- [21] "The NIST Definition of Cloud Computing". National Institute ofStandards and Technology. Retrieved 24 July 2011.
- [22] University of Computer Studies , Yangon in Myanmar <http://www.ucsy.edu.mm>
- [23] http://en.wikipedia.org/wiki/Microsoft_Azure
- [24] <http://azure.microsoft.com/enus/documentation/articles/fundamentalsintroduction-to-azure/>
- [25] http://en.wikipedia.org/wiki/Platform_as_a_service
- [26] [http://en.wikipedia.org/wiki/IaaS#Infrastructure as_a_service_28IaaS.29](http://en.wikipedia.org/wiki/IaaS#Infrastructure_as_a_service_28IaaS.29)

Analytics of Reliability for Real-Time Big Data Pipeline Architecture

Thandar Aung, Aung Htein Maw
University of Information Technology, Yangon, Myanmar
thandaraung@uit.edu.mm, ahmaw@uit.edu.mm

Abstract

Nowadays, many applications need high reliability pipeline architecture to get faster process and reliable data within short time. Kafka has emerged as one of the important components of real-time processing pipelines in combination with Storm. This paper focuses to develop the real-time big data analytics pipeline architecture for reliability. Real-time data pipelines can be implemented in many ways and it will look different for every business. To develop the pipeline architecture, we create real time big data pipeline by using Apache Kafka and Apache Storm. Kafka and Storm naturally complement each other and their powerful cooperation enables real-time streaming analytics for fast-moving big data. Then, the experiment will be conducted how the processing time decreases with the same messages on the different partitions.

Keywords- Messaging, Real-time processing, Apache Kafka, Apache Storm

1. Introduction

In the present big data era, the very first challenge is to collect the data as it is a huge amount of data and the second challenge is to analyze it. This analysis typically includes User behavior data, Application performance tracing, Activity data in the form of logs and Event messages. Processing or analyzing the huge amount of data is a challenging task. It requires a new infrastructure and a new way of thinking about the way business and IT industry works. Today, organizations have a huge amount of data and at the same time, they have the need to derive value from it. Considering the huge volume and the incredible rate at which data is being collected, the need arises for an efficient analytic system which processes this data and provides value in real time.

Real-time processing is a fast and prompt data processing technology that combines data capturing, data processing and data exportation together. Real-time analytics is an iterative process involving multiple tools and systems. It consists of dynamic analysis and reporting, based on data entered into a system less than one minute before the actual time of use [1]. In contrast to traditional data analytical systems that collect and periodically process huge –static –volumes of data, streaming analytics systems avoid putting data at rest and

process it as it becomes available, thus minimizing the time a single data item spends in the processing pipeline[2]. The main purpose of Big Data real-time processing is to realize an entire system that can process such mesh data in a short time[4]. Real-time information is continuously getting generated by applications (business, social, or any other type), and this information needs easy ways to be reliably and quickly routed to multiple types of receivers. Most of the time, applications that are producing information and applications that are consuming this information are well apart and inaccessible to each other. This, at times, leads to redevelopment of information producers or consumers to provide an integration point between them. Therefore, a mechanism is required for seamless integration of information of producers and consumers to avoid any kind of rewriting of an application at each end.

Real-time usage of these multiple sets of data collected from production systems has become a challenge because of the volume of data collected and processed. Kafka has high throughput, built-in partitioning, replication, and fault-tolerance, which makes it a good solution for large scale message processing applications [8]. In this paper, we propose to develop real time big data analytics pipeline architecture by using Apache Kafka and Apache Storm.

The remainder of this paper is organized as follows: section 2 reviews the related work of this paper. Section 3 presents the proposed system architecture. In Section 4, we describe the architecture of Kafka, the zookeeper which needs to run Kafka. The process of Apache Storm shows in Section 5. Section 6 describes the framework of our system and testing results for this proposed system. Then, Ring Election Algorithm is intended to enhance the pipeline architecture in the future. Section 7 describes conclusion and future work.

2. Related Work

Khin Me Me Thein [1] has proposed to provide the secure big data pipeline architecture for the scalability and security. The author used Sticky policies and AES Algorithm for secure big data pipeline for real time streaming applications.

Steffen FriedWolfram Wingerath, FelixGessert,rich, and Norbert Ritter [2] have also proposed qualitative comparison of the most popular distributed stream

processing systems. The author gives an overview over the state of stream processors for low-latency Big Data analytics and conduct a qualitative comparison of the most popular contenders, namely Storm and its abstraction layer Trident, Samza and Spark Streaming. In their paper, Streaming processing system is high availability, fault-tolerance and horizontal scalability.

Mohit Maske, Dr. Prakash Prasad, International Journal of Advanced [3] intends to ensure the practical and high efficiency in simulation system that is established and shown acceptable performance in various expressions using data sheet. It proved that data analysis system for stream and real time processing based on storm can be used in various computing environment.

Wenjie Yang, Xingang Liu and Lan Zhang [4] have also proposed to ensure the practical applicability and high efficiency, to establish and shows acceptable performance in simulation. In their paper, an entire system RabbitMQ, NoSQL and JSP are proposed based on Storm, which is a novel distribution real-time computing system. The paper organized a big data real-time processing system based on Storm and other tools, and according to the simulation experiment, the system can be easily applied in practical situation.

Martin Kleppmann [5] explains the reasoning behind the design of Kafka and Samza, which allow complex applications to be built by composing a small number of simple primitives – replicated logs and stream operators. We draw parallels between the design of Kafka and Samza, batch processing pipelines, database architecture and design philosophy of UNIX.

P Beulah Soundarabai, ThriveniJ, K R Venugopal, L M Patnaik [6] describes the process of ring Election Algorithm and presents a modified version of ring algorithm. Their paper involves substantial modifications of the existing ring election algorithm and the comparison of message complexity with the original algorithm. Simulation results show that our algorithm minimizes the number of messages being exchanged in electing the coordinator. Each of Election Algorithms gives better performance in terms of time and messages.

Seema Balhara, Kavita Khanna[7] has proposed to maintain coordination between the nodes and leader node have to be selected. Their paper contains the information about the various existing leader election mechanisms which is used for selecting the leader in different problem. The author discusses about several election algorithm in Distributed system.

Jiangyong Cai, Zhengping Jin[12] has proposed a real-time processing scheme for the self-health data from a variety of wearable devices by using storm. Their designs a framework using Apache Storm, distributed framework for handling stream data, and making decisions without any delay. Their framework has improved more efficient than the old method of using regular task with DB cluster.

3. Proposed System Architecture

In this section, we focus on the design and architecture of big data real-time pipeline as our proposed system architecture in Figure 1.

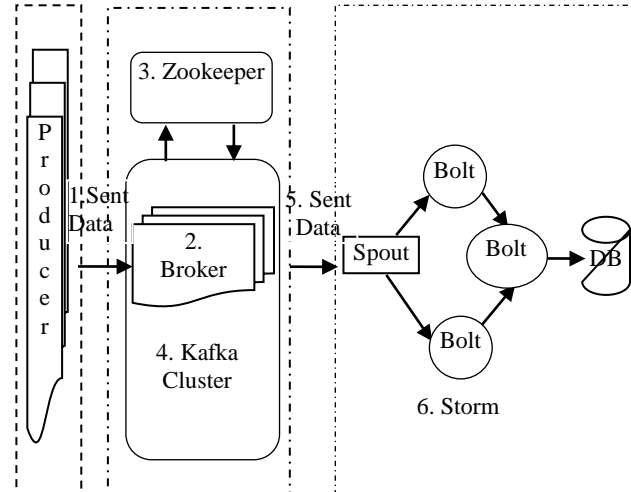


Figure 1. Proposed System Architecture

The processes of proposed system architecture are as follows:

1. In Apache Kafka, Producer send messages to consumers. Brokers can divide messages in many partitions.
2. Each partition is optionally replicated across a configurable number of servers for fault tolerance. Each partition available on either of the servers acts as the leader and has zero or more servers acting as followers.
3. If one of the followers fails, the system can choose follower in-sync replicas (ISR) list. If the leader fails, the system can elect leader randomly in processing.
4. Kafka is a high-performance publisher-subscriber-based messaging system .Kafka spout is available for integrating Storm with Kafka clusters.
5. The Kafka spout is a regular spout implementation that reads the data from a Kafka cluster. Kafka has emerged as one of the important components of real-time processing pipelines in combination with Storm.
6. Kafka can act as a buffer or feeder for messages that need to be processed by Storm. Kafka can also be used as the output sink for results emitted from the Storm topologies. By constructing real time pipeline architecture, two processes can run concurrently. When a process is running in storm, another process can run in Kafka. So, real time message processes faster and faster. It can process high performance in message parsing system.

4. Apache Kafka architecture

Kafka[8] is an open source, distributed publish subscribe messaging system, mainly designed with the following characteristics:

Persistent messaging: To derive the real value from big data, any kind of information loss cannot be afforded. Apache Kafka is designed with O(1) disk structures that provide constant-time performance even with very large volumes of stored messages, which is in order of TB.

High throughput: Keeping big data in mind, Kafka is designed to work on commodity hardware and to support millions of messages per second.

Distributed: Apache Kafka explicitly supports messages partitioning over Kafka servers and distributing consumption over a cluster of consumer machines while maintaining per-partition ordering semantics.

Multiple client support: Apache Kafka system supports easy integration of clients from different platforms such as Java, .NET, PHP, Ruby, and Python.

Real time: Messages produced by the producer threads should be immediately visible to consumer threads; this feature is critical to event-based systems such as Complex Event Processing (CEP) systems. Kafka which provides a real-time publish-subscribe solution for overcoming the challenges of consuming the real-time and batch data volumes that may grow in order of magnitude to be larger than the real data.

Table 1. Characteristics of Kafka

Feature	Description
Scalability	Distributed system scales easily with no downtime
Durability	Persists messages on disk, and provides intra-cluster replication
Reliability	Replicates data, supports multiple subscribers, and automatically balances consumers in case of failure
Performance	High throughput for both publishing and subscribing, with disk structures that provide constant performance even with many terabytes of stored messages

Apache Kafka is a real time, fault tolerant, scalable messaging system for moving data in real time. Kafka maintains feeds of messages in categories called topics. We'll call processes that publish messages to a Kafka topic are producers. And we'll call processes that subscribe to topics and process the feed of published messages are consumers. Kafka is run as a cluster comprised of one or more servers each of which is called a broker. Producers send messages over the network to the Kafka cluster which in turn serves them up to consumers. A producer publishes messages to a Kafka

topic. Kafka topic is also considered as a message category or feed name to which messages are published. Kafka topics are created on a Kafka broker acting as a Kafka server. Processes that subscribe to topics and process the feed of published messages are called consumers. Brokers and consumers use Zookeeper to get the state information and to track message offsets, respectively. In figure 2, single node-multiple broker architecture is shown with a topic having four partitions. There are five components of the Kafka cluster: Zookeeper, Broker, Topic, Producer, and Consumer.

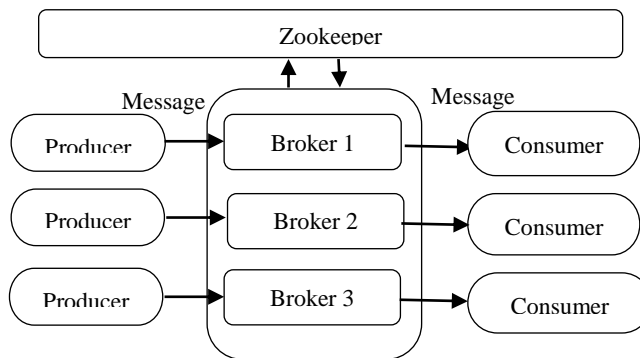


Figure 2. A single node-multiple broker architecture

All the message partitions are assigned a unique sequential number called the offset, which is used to identify each message within the partition. Each partition is optionally replicated across a configurable number of servers for fault tolerance. Each partition available on either of the servers acts as the leader and has zero or more servers acting as followers. Here the leader is responsible for handling all read and write requests for the partition while the followers asynchronously replicate data from the leader. Kafka dynamically maintains a set of in-sync replicas (ISR) that is caught-up to the leader and always persist the latest ISR set to Zookeeper. In a Kafka cluster, each server plays a dual role; it acts as a leader for some of its partitions and also a follower for other partitions. If any of the follower in-sync replicas fail, the leader drops the failed follower from its ISR list. After the configured timeout period and writes will continue on the remaining replicas in ISRs. Whenever the failed follower comes back, it truncates its log to the last checkpoint and then starts to catch up with all messages from the leader, starting from the checkpoint. As soon as the follower becomes fully synced with the leader, the leader adds it back to the current ISR list.

If the leader fails, the process of choosing the new lead replica involves all the followers' ISRs registering themselves with Zookeeper. The very first registered replica becomes the new lead replica and its log end offset (LEO) becomes the offset of the last committed. The rest of the registered replicas become the followers of the

newly elected leader. The system occurs the problem in leader Election because Kafka dynamically maintains a set of in-sync replicas. So the replica is not reliable in processing. Figure 3 explain replication in Kafka:

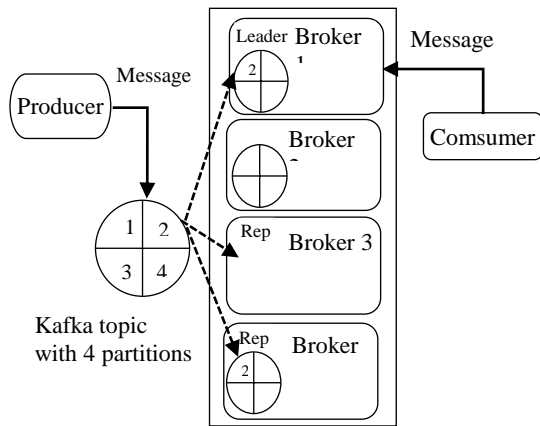


Figure 3. Replication in Kafka

4.1. Zookeeper

Zookeeper [10] is a centralized service for maintaining configuration information, naming, providing distributed synchronization, and providing group services. Zookeeper is also a high-performance coordination service for distributed applications. Each time they are implemented there is a lot of work that goes into fixing the bugs and race conditions that are inevitable. Because of the difficulty of implementing these kinds of services, applications initially usually skimp on them, which make them brittle in the presence of change and difficult to manage. When it works correctly, different implementations of these services lead to management complexity when the applications are deployed. The service itself is distributed and highly reliable.

Kafka uses Zookeeper for the following tasks: Detecting the addition and the removal of brokers and consumers. Triggering a rebalance process in each consumer when the above events happen, and Maintaining the consumption relationship and keeping track of the consumed offset of each partition. Specifically, when each broker or consumer starts up, it stores its information in a broker or consumer registry in Zookeeper. The broker registry contains the broker's host name and port, and the set of topics and the partitions stored on it.

5. Apache Storm

Storm [9] is also an open source, distributed, reliable, and fault-tolerant system for processing streams of large volumes of data in real-time. It supports many use cases, such as real-time analytics, online machine learning, continuous computation, and the Extract Transformation

Load (ETL) paradigm. Storm can be used for the following use cases:

Stream processing: Storm is used to process a stream of data and update a variety of databases in real time. This processing occurs in real time and the processing speed needs to match the input data speed.

Continuous computation: Storm can do continuous computation on data streams and stream the results into clients in real time. This might require processing each message as it comes or creates small batches over a little time. An example of continuous computation is streaming trending topics on Twitter into browsers.

Distributed RPC: Storm can parallelize an intense query so that you can compute it in real time.

Real-time analytics: Storm can analyze and respond to data that comes from different data sources as they happen in real time. A Storm cluster follows a master-slave model where the master and slave processes are coordinated through Zookeeper. The Storm Cluster is made up of a main node and several working nodes [4].

5.1. Nimbus

The Nimbus node is the master in a Storm cluster. A daemon process called "Nimbus" is running on main node, in order to allocate codes, arrange tasks and detect errors.

5.2. Supervisor

Supervisor nodes are the worker nodes in a Storm cluster. Each working node has a daemon process called "Supervisor" to monitor, start and stop working process. The coordination work between Nimbus and Supervisor is handled by "Zookeeper" as shown in Fig 4. Zookeeper is the subproject of Hadoop, and it aims at coordinate works in large-scale distribution system. The Storm Cluster is similar with Hadoop, where Nimbus corresponds to Job Tracker, and Supervisors correspond to Task Trackers.

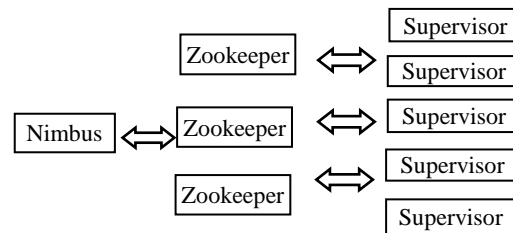


Figure 4. Storm Cluster's Architecture

In Storm terminology, [9] a topology is an abstraction that defines the graph of the computation. A topology can be represented by a direct acyclic graph, where each node does some kind of processing and forwards it to the next node(s) in the flow. The followings are the components of a Storm topology:

Stream: A stream is an unbounded sequence of tuples that can be processed in parallel by Storm. Each stream can be processed by a single or multiple types of bolts.

Spout: A spout is the source of tuples in a Storm topology. Spout is the input stream source which can read from external data source. [12] For example, by reading from a log file or listening for new messages in a queue and publishing them-emitting, in Storm terminology-into streams.

Bolt: The spout passes the data to a component called bolt. Bolts [12] are processor units which can process any number of streams and produce output streams. A bolt is responsible for transforming a stream. Each bolt in the topology should be doing a simple transformation of the tuples, and many such bolts can coordinate with each other to exhibit a complex transformation. There is an example of one topology in Figure 5.

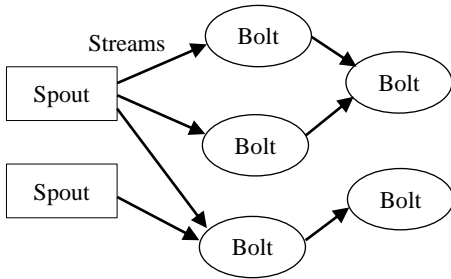


Figure 5. Storm Topology

6. Experimental Setup and Results

The experimental set up is performed by using two open source frameworks Apache Kafka 0.8.1.1 and Apache Storm 0.9.3 as the main pipeline architecture. JAVA/jre 1.4 is running on underlying pipeline architecture. The Apache Marven 3.11 is used as in Kafka-Storm integration.

The processes of overall pipeline architecture are as follows:

1. Start zookeeper server for processing.
2. Start Kafka local server to define *broker_id*, *port* and *log dir*.
3. Create topic to show a successful creation message.
4. Producer publishes them as a message to the Kafka cluster.
5. The consumer consumes messages.
6. Get some message in our Apache Kafka cluster.
7. Execute by using command to verify whether topic created.
8. Recall producer and write messages again.
9. Get some messages and run Kafka-Storm integration pipeline.

Table 2 shows testing data in pipeline architecture which is one broker and five partitions using text

messages. The purpose of our experiment is to show the comparison of the processing time on the same messages with different partitions. We tested four different numbers of messages; they were 18, 22, 24 and 29 messages with five partitions. It shows the faster processing time on various messages and partitions.

Table 2. Testing Data in pipeline

No. of partitions	Number of messages with processing time			
	18 Msg	22 Msg	24 Msg	29 Msg
1	1.7 sec	1.9 sec	2 sec	2.4 sec
2	0.9 sec	1.1 sec	1.4 sec	1.5 sec
3	0.6 sec	0.7 sec	0.8 sec	0.9 sec
4	0.4 sec	0.5 sec	0.6 sec	0.7 sec
5	0.2 sec	0.3 sec	0.4 sec	0.5 sec

Msg=Messages
Sec=seconds

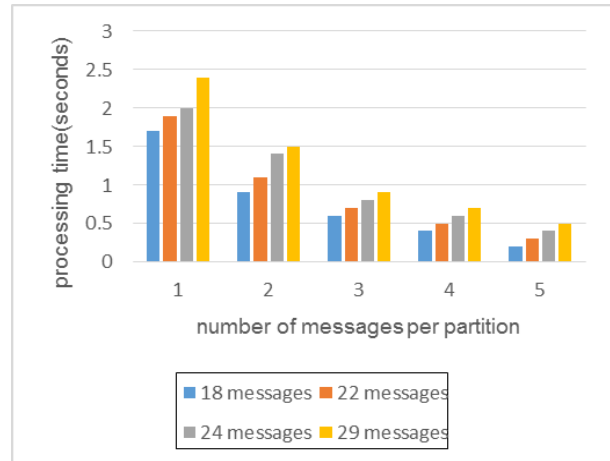


Figure 6. Testing Result of different numbers of messages per partition in pipeline Architecture

Figure 6 shows the process of pipeline by testing on various partitions. According to the experiment, processing time decreases with more partitions are used and the comparison of number of partitions based on various amount of messages. So, this pipeline architecture effects on parallel processing in real time.

6.1 Ring Election Algorithm

In the future, we intend to propose ring election Algorithm in replication process. In Kafka framework, we face any problem in leader Election in processing. In processing, if one of the followers fails, the system can

choose follower in-sync replicas (ISR) list. If the leader fails, the system can elect leader by replacing ring election Algorithm in processing. By using Ring Election Algorithm, we can get more reliable data in Kafka-storm pipeline architecture.

The goal of Ring Election Algorithm is to choose and declare one and only process as the leader even if all processes participate in the election. And at the end of the election, all the processes should agree upon the new leader process with the largest process identifier without any confusion. Ring Election Algorithm can elect new leader without wasting of time and number of messages which are exchanged. Depending on a network topology, many algorithms have been presented for electing leader in distributed systems. The Ring Election Algorithm is based on the ring topology with the processes ordered logically and each process knows its successor in a unidirectional way, either clockwise or anticlockwise. The process of Ring Election Algorithm [10] describes in Figures 7.

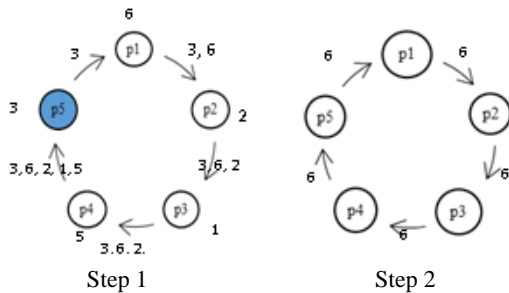


Figure 7. Processes of Ring Election

Step 1. A process is a leader as it has highest id number. When a leader process fails, it starts leader election. It sends message with its id to next node in the ring. The next process passes the message on adding its own id to the message again and again.

Step 2. When starting process receives the message back, it knows the message has gone around the ring, as its own id is in the list. Picking the highest id in the list, it starts the coordinator message as the leader around the ring.

7. Conclusion

In this paper we have implemented a real time framework using Apache Kafka and Apache storm. This pipeline has the reliability to deliver the streaming data. We use Apache Kafka and Apache Storm to develop high performance big data pipeline architecture for real time streaming applications. Using the proposed pipeline architecture, the completion time decrease although a number of messages increase. So, this factor is reliable for the proposed pipeline architecture. We emphasize to be

reliable message in real time big data pipeline architecture.

As future direction, we intend to propose Ring election Algorithm in replication process. By using Ring Election Algorithm, we can get more reliable data in Kafka-storm pipeline architecture.

8. References

- [1] Khin Me Me Thein, "Security of Real-time Big Data Analytics Pipeline", International Journal of Advances in Electronics and Computer Sciences, Feb, 2017.
- [2] Wolfram Wingerath*, Felix Gessert, Steffen Friedrich, and Norbert Ritter, "Real-time stream processing for big data", May 2016.
- [3] Mohit Maske, Dr. Prakash Prasad, "A Real Time Processing and Streaming of Wireless Network Data using Storm ", International Journal of Advanced Research in Computer Science and Software Engineering, January 2015.
- [4] Wenjie Yang, Xingang Liu and Lan Zhang, "Big Data Real-time Processing Based on Storm", 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2013.
- [5] Martin Kleppmann, "Kafka, Semza and the Unix Philosophy of Distributed Data" Bulletin of the IEEE computer Society Technical Committee on Data Engineering.
- [6] P Beulah Soundarabai, Thriveni J, KR Venugopal, L M Patnaik, "An Improved Leader Election Algorithm for Distributed System", International Journal of Next-Generation Networks (IJNGN), March 2013.
- [7] Seema Balhara, Kavita Khanna, "Leader Election Algorithms in Distributed System", International journal of Computer Science and Mobile Computing, June 2014.
- [8] Nishant Garg, "Apache Kafka", PACKT Publishing UK, 2015.
- [9] <http://zookeeper.apache.org/>.
- [10] Tanenbaum Andrew, Tanenbaum-Distributed operating system, Wikipedia, p-100, 1994
- [11] Shay Kuten, Shlomo Moran, Leader election, Wikipedia, 24 August 2017
- [12] Jiangyong Cai, Zhengping Jin, "Real-time Calculating Over Self-Health Data Using Storm", 4th International Conference on Mechatronics, Materials, Chemistry and Computer Engineering (ICMMCCE 2015).
- [13] Ankit Jain, Anand Nalya, "Learning Storm" PACKT Publishing UK, 2015.

Optimum Checkpoint Interval for MapReduce Fault-Tolerance

Naychi Nway Nway, Julia Myint

University of Information Technology, Yangon, Myanmar
naychinwaynway@uit.edu.mm, juliamyint@uit.edu.mm

Abstract

MapReduce is the efficient framework for parallel processing of distributed big data in cluster environment. In such a cluster, task failures can impact on performance of applications. Although MapReduce automatically reschedules the failed tasks, it takes long completion time because it starts from scratch. The checkpointing mechanism is the valuable technique to avoid re-execution of failed tasks in MapReduce. However, defining incorrect checkpoint interval can still decrease the performance of MapReduce applications and job completion time. In this paper, the optimum checkpoint interval is proposed to reduce MapReduce job completion time when failures occur. The proposed system defines checkpoint interval that is based on five parameters: expected job completion time without checkpointing, checkpoint overhead time, rework time, down time and restart time. Therefore, because of proposed checkpoint interval, MapReduce does not need to re-execute the failed tasks, so it reduces job completion time when failures occur. The proposed system reduces job completion time even though the number of failures increases and the performance of this system can be improved 4 times better than the original MapReduce.

Keywords- MapReduce, big data, task failures, completion time, checkpoint interval

1. Introduction

Data-intensive applications process vast amounts of data with special-purpose programs. Even though the computations behind these applications are conceptually simple, the size of input datasets requires them to be run over thousands of computing nodes. For this, Google developed the MapReduce framework, which allows non-expert users to run complex tasks easily over very large datasets on large clusters. The large datasets are often messy, containing data inconsistencies and missing value (bad records). This may, in turn, cause a task or even an application to crash. Google reports 5 average worker deaths per MapReduce job in March 2006 [8], and at least one disk failure in every run of a 6 hour MapReduce job with 4,000 machines [16].

The impact of task failures can be considerable in terms of performance [7]. In MapReduce process, after map stages the intermediate data is produced and it is the

input for reduce stages. So, intermediate data is important to be successful MapReduce process. Although MapReduce can restart the process and produce intermediate data again when task failures occur, it can prolong job completion time.

Fault-tolerance is, in fact, an important aspect in large clusters because the probabilities of task failures increase with the growing of computing nodes. It allows a computation in progress in spite of having individual failures in system. Checkpoint saves the system state in stable storage so it can reduce the amount of lost computation. The performance of defining correct checkpoint interval can reduce job completion time when failures occur.

Therefore, in this paper, checkpoint-based fault-tolerance with optimum checkpoint interval is proposed to reduce the job completion time when task failures occur in Hadoop MapReduce. The proposed system addresses the surveys of related work in Section 2. Section 3 describes the basic flow and built-in fault-tolerance of MapReduce. The checkpoint interval and implementation of proposed system are described in Section 4 and 5. Section 6 proposes the experimental results and finally, the conclusion of this paper is presented in Section 7.

2. Related Work

MapReduce [1] is a parallel programming model which is originally proposed by Google in 2004 to deal with the rapidly increasing demand of processing mass data concurrently. Through well-defined interfaces and runtime support library, MapReduce can automatically perform the large-scale computing tasks in parallel, hide the underlying implementation details, and reduce the difficulty of parallel programming, which makes MapReduce become one of the most widely used parallel programming models in the concurrent processing vast amount of data. MapReduce considers task and worker failures as characteristic rather than exception. As a result, it comes with fault tolerance strategies. However, applications can experience significant performance downgrade in case of failures. According to a recent study [11], a single failure on a Hadoop job could cause a 50% increase in completion time.

RAFTing MapReduce presented in [9] tries to create several kinds of checkpoint to handle different failures. RAFT-LC is a local checkpointing algorithm that allows a

map task to store progress metadata on local disk and later restore based on this in case of failures. RAFTing mappers push data to reducers instead of the opposite way and make the intermediate data replicated without bringing much overhead.

To prevent task failures in MapReduce, CROFT [13] proposed a checkpoint and replication oriented fault tolerant scheduling algorithm, which uses a checkpoint based active replication method. It also creates a local checkpoint file which is responsible for recording the progress of the current task and a global index file which is responsible for recording the characteristics of the current execution.

In paper [14], the author introduced two checkpoint algorithms to eliminate the costs of re-reading, re-copying, and re-computing the partial processed data. It makes an input checkpoint to record the location of unprocessed input data, while the output checkpoint consists of spilled files and their index information. Young proposed a first-order model that defines the optimal checkpoint interval in terms of checkpoint overhead and mean time to interrupt (MTTI). Young's model does not consider failures during checkpointing and recovery [12].

Given the checkpointing parameters such as checkpoint latency and MTTI, Daly's model [3] provides a method for computing the optimal checkpoint which is associated with the optimal execution time. The choice of a checkpoint interval influences the number of checkpoint operations performed during an application's execution. Checkpoints are created when the progress reaches 0.5 (or) 0.25 by calculation progress rate and estimated task execution time [2]. When the checkpoints are created in 50% of execution time, the failed tasks before 50% cannot be recovered. The checkpointing mechanism for 25% of progress score can cause the network traffic.

To ensure that checkpoints can be used effectively, the proposed system introduces optimum checkpoint interval that aims to recover from task failures and to improve performance as the main goal. Unlike original MapReduce, the proposed system reschedules the failed tasks without starting again. The experiments show that the proposed system outperforms original MapReduce with a 20% increasing of performance.

3. The MapReduce Framework

3.1. Execution Flow of MapReduce

MapReduce [5] adopted a two-stage and shared-nothing design. In the first stage, the map stage, MapReduce takes a list of key value pairs as input, and applies a map function on each of the pair to generate arbitrary number of intermediate key value pairs. In the second stage, all the intermediate values associated with the same keys are grouped together as a list, and a reduce function takes each of the groups as input to generate

another arbitrary number of final output key value pairs. The paradigm behind MapReduce is a quite simple behavior because a map or reduce function call on a key value pair shall depend neither on other pairs nor on the processing order. This makes it easy to split the whole job into smaller independent subtasks that can run in parallel.

The input data files of MapReduce are usually stored on a DFS (distributed file system) such as HDFS, an open-source implementation of GFS. The data files are split into small pieces logically, every one of which will be fed to a map task. Map tasks, also known as mappers, parse raw input data splits into $k_1 v_1$ pairs, and invoke the map function on every single pair, the generated k_2 and v_2 pairs are written to a memory buffer. When the buffer verges to overflow, the mapper flushes it to a local disk file, which is called a spill. A mapper may create several spill files, however, it will merge the spill files into a single output file on local disk after all input records are processed. There are usually several reduce tasks, or reducers, key value pairs with the same key hash value go to the same reducer. As a result, the single map output file shall be logically spilt into R parts, each part will be fed to a reducer. A reduce task can be summarized to 3 main phases: shuffle, sort and reduce. During the shuffle phase, reducers copy outputs from each mapper, and merge the outputs into less amount of files in the sort phase. The shuffle phase and sort phase often overlap in practice, but the reduce phase shall not start until shuffle phase finishes, which is limited by the MapReduce semantics. The execution flow of MapReduce is shown in Figure. 1.

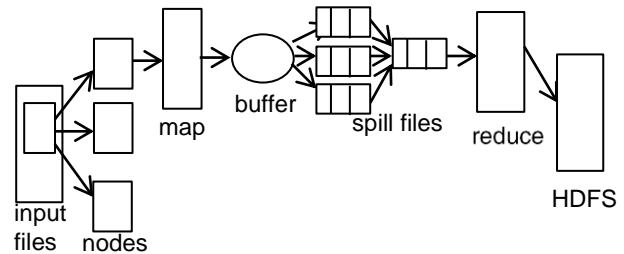


Figure 1. Execution Flow of MapReduce

3.2. Fault-Tolerance in MapReduce

Hadoop has been built with some level of faults tolerance [10]. MapReduce adopted a centralized design, an instance of Hadoop MapReduce deployment basically consists of a master and several slaves [4]. The master keeps several data structures, like the state and the identity of the worker machines [15]. Slaves execute the task on master's request, and each execution of a task is called a task attempt. A task attempt periodically informs the master about its latest status information [5]. Once the master receives status report from a task attempt indicating failure, or a task attempt fails to contact the master for a certain amount of time, the task attempt is considered to have failed and the master will schedule

another attempt for the same task. The new attempt will recompute the whole input split of the task regardless of the progress of last attempt. Task attempt failures may result from bad records, such as invalid or inconsistent field values, which is common in big data analysis. In the worst case, the last record of an input split is corrupted and it will result in a second task attempt processing the exact same input and doubles the task execution time at least. In Hadoop, the bad record will be skipped in a third attempt, and apparently the delay caused by the single bad record is too high and not tolerable.

While checkpointing is one of the most widely used techniques in fault tolerance [12], a naïve implementation of checkpointing in Hadoop may downgrade the performance. Due to the fact that a MapReduce job often processes vast amount of input data, the intermediate data generated is usually also very large. Checkpointing requires the intermediate data to be replicated among several nodes, which involves huge amount of disk IO and network IO, the two most critical resources in MapReduce. Checkpointing strategy in MapReduce needs to be carefully designed.

4. Checkpoint Interval

A checkpoint interval [3] is defined as the duration between the establishments of two consecutive checkpoints. That is, an interval begins when one checkpoint is established, the interval ends when the next checkpoint is established. Figure 2 shows how to define checkpoint interval and T is the amount of useful computation in each interval, C is checkpoint overhead and L means the duration of time needed to save the checkpoint [6].

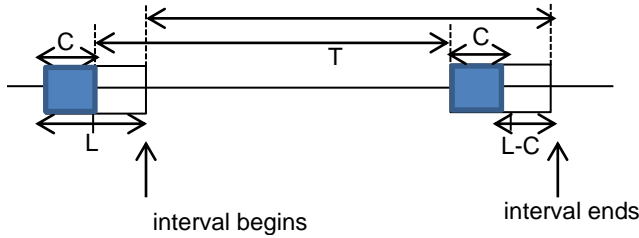


Figure 2. Checkpoint Interval

5. Proposed System Design

The proposed system aims to minimize job completion time due to failures in MapReduce by determining checkpoint interval that is based on task failures. Before calculating checkpoint interval, the system calculates the expected job completion time [5] without checkpoint using equation (1)

$$Tc = \left(\frac{Tn}{w}\right) * \left(Jt + \frac{Dsize}{Jp}\right) \quad (1)$$

where Tc means job completion time, Tn means the number of tasks, w means the number of workers, Jt means time to take JVM, $Dsize$ means input data size and Jp means processing size of JVM per second.

After that, based on job completion time, the system calculates interval between checkpoint files that minimizes the time lost when failures occur using equation (2)

$$T = \text{Completion Time} + \text{Overhead Time} + \text{Rework Time} + \text{Down Time} + \text{Restart Time} \quad (2)$$

Completion Time is defined as actual completion time without checkpoints. Overhead Time is overhead for writing checkpoint files, Rework Time is time lost due to failures, Down Time is time lost when an application cannot reach current running state and Restart Time is time required before an application resumes to current work. Completion Time will be Tc and Overhead Time will be $\beta(C(\tau) - 1)$ where $C(\tau)$ is number of checkpoint taken and one is subtracted because there is no need to write checkpoint files in last segment. For Rework Time, it will be described by $\frac{1}{2}(\tau + \beta)N(\tau)$ where $N(\tau)$ is expected number of interrupts. Down Time is used as $DN(\tau)$ and finally, Restart Time is $RN(\tau)$, the amount of time required to restart times total number of failures. So, the system constructs the formula as equation (3)

$$T = Tc + (C(\tau) - 1)\beta + \frac{1}{2}(\tau + \beta)N(\tau) + DN(\tau) + RN(\tau) \quad (3)$$

Next, system determines the number of interrupts $N(\tau)$ and numbers of checkpoints are calculated by dividing completion time by checkpoint interval. The expected number of interrupts can be calculated by the product of numbers of checkpoints required to complete calculation and the probability of each segment failing as in equation (4)

$$N(\tau) = \frac{Tc}{\tau} \left(e^{\frac{\tau+\beta}{M}} - 1 \right) \cong \frac{Tc}{\tau} \left(\frac{\tau+\beta}{M} \right) \quad (4)$$

Then, $N(\tau)$ is substituted in equation 3:

$$T = Tc + \left(\frac{Tc}{\tau} - 1\right)\beta + \left[\frac{1}{2}(\tau + \beta) + D + R\right] \frac{Tc}{\tau} \left(\frac{\tau + \beta}{M}\right) \quad (5)$$

Using equation 5, the system finds the minima with respect to τ that sets the derivation to zero.

$$e^{\frac{\tau+\beta}{M}} [\tau^2 + (\beta + 2R + 2D)\tau - (\beta + 2R + 2D)M] + 2RM - \beta M = 0 \quad (6)$$

Instead of expanding the exponential term, recast equation 6 as follows:

$$\frac{\tau + \beta}{M} = \ln \left[\frac{(\beta - 2R)M}{\tau^2 + (\beta + 2R + 2D)\tau - (\beta + 2R + 2D)M} \right] = \ln[g(\tau)] \quad (7)$$

The system which calculates a Taylor series expansion for natural logarithm of $g(\tau)$ is as follows:

$$\frac{\tau + \beta}{M} = \frac{g(\tau) - 1}{g(\tau)} + \frac{1}{2!} \left(\frac{g(\tau) - 1}{g(\tau)} \right)^2 + \frac{1}{3!} \left(\frac{g(\tau) - 1}{g(\tau)} \right)^3 + \dots$$

$$= \left(1 - \frac{1}{g(\tau)} \right) + \frac{1}{2} \left(1 - \frac{1}{g(\tau)} \right)^2 + \frac{1}{3} \left(1 - \frac{1}{g(\tau)} \right)^3 + \dots \quad (8)$$

Reduce the equation 8 to quadratic form as in (9)

$$\tau^2 + 2D\tau + (\beta^2 - 2\beta(R + M) - 2DM) = 0 \quad (9)$$

Finally, the value of τ which minimize equation 5 as follows:

$$\tau = -\beta + \sqrt{2\beta(R + M) + 2DM} \quad (10)$$

The proposed system defines checkpoint interval (τ) after processing 50 seconds. After calculating checkpoint interval, the system creates a checkpoint file in local disk with three checkpoint information: taskID, a unique task identifier and offset that specify the last byte of input data processed by map tasks.

6. Experiment

We analyze the performance of the proposed system in this section. Experiments are designed to measure the job completion time in the case of task failures. The implementation of the proposed system is based on Hadoop 2.7.1, Java 1.8 and Hadoop Distributed File System (HDFS) with data size of 400MB, 500MB and 600MB. The jobs for experiments are word count over user-submitted comments on StackOverflow.

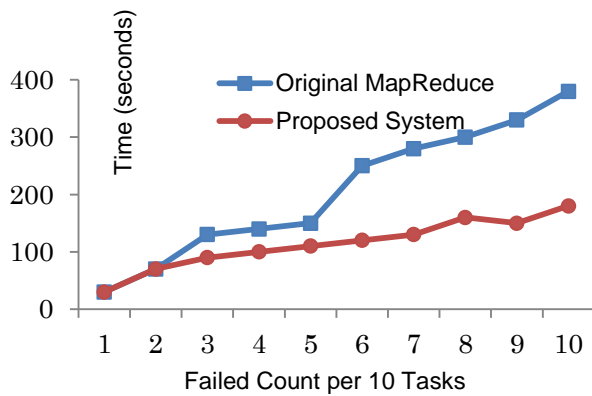


Figure 3. Comparison of Completion Time of 10 Tasks with Task Failure

Figure 3 shows the comparison of MapReduce job completion time between original MapReduce and

proposed system with 400MB. The x-axis is the number of task errors per 10 tasks and y-axis is the total completion time. According to the experiment, if a number of errors increase, the completion time of the job will take 4 times less than the original Hadoop. When failures occur, the proposed system reads checkpoint files more frequently so it saves job completion time. The experiment of Figure 4 with 500MB and Figure 5 with 600MB also show that the performance of proposed system is better than original MapReduce when the number of failures increases.

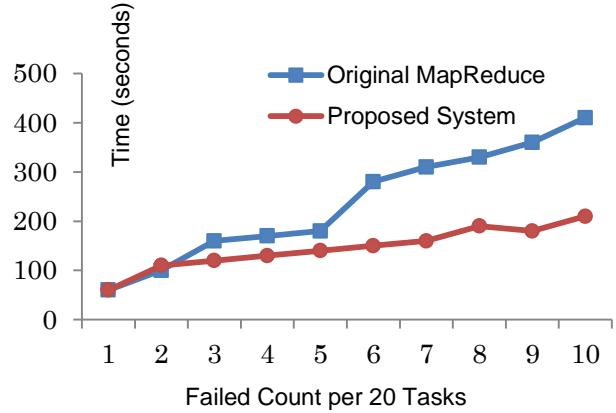


Figure 4. Comparison of Completion Time of 20 Tasks with Task Failure

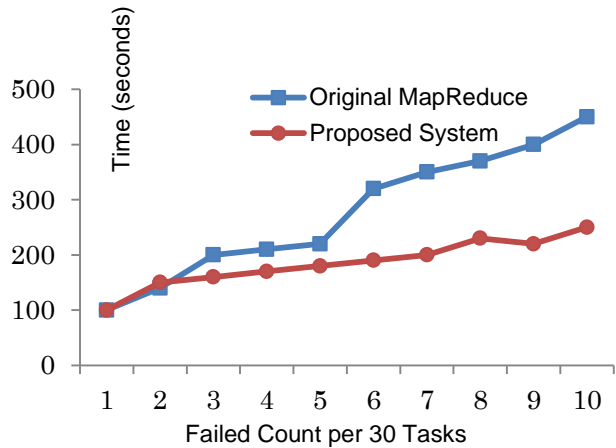


Figure 5. Comparison of Completion Time of 30 Tasks with Task Failure

7. Conclusion

MapReduce is a popular programming model that allows the user with simple APIs and is able to run big data applications. MapReduce is also able to retry the failure tasks but it performs poorly because of start from scratch. Although the original MapReduce facilitates fault-tolerance with re-executing of failed tasks, it can

prolong job completion time when failures occur. The proposed system presents checkpointing mechanism not to re-execute failed tasks from start. In order not to delay long job completion because of checkpointing, the proposed system defines optimum checkpoint interval that has the advantageous of reducing job completion time when failures occur.

As future direction, we intend to propose a task migration technique for slow tasks in MapReduce. The main causes of slow tasks in MapReduce are (i) a slow node and (ii) input data skew. Slow tasks in MapReduce also threaten the job completion so we will combine checkpointing and task migration techniques to solve the problem of slow tasks in MapReduce.

8. References

- [1] B.Cho, and I.Gupta, "Making cloud intermediate data fault-tolerant", ACM symposium on Cloud Computing,2010.
- [2] C.Lin, T.Chen, and Y. Cheng. "On Improving Fault Tolerance for Heterogeneous Hadoop MapReduce Clusters", IEEE International Conference on Cloud Computing and Big Data, 2014.
- [3] D.John."Future Generation Computer Systems", Volume 22, Issue 3, February 2006, pp. 303-312.
- [4] H.Wang, H.Chen, and F.Hu. "ReCT: Improving MapReduce Performance under Failures with Resilient Checkpoint Tactics", IEEE International Conference on Big Data,2014.
- [5] H.Wang, H.Chen, and F.Hu, "BeTL: MapReduce Checkpoint Tactics Beneath the Task Level", IEEE Transactions on Services Computing,2016.
- [6] H.Nitin, "On Checkpoint Latency", Technical Report,1995.
- [7] J.Dean and S Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", In 6th symposium on Operating System Design and Implementation (OSDI), San Francisco, December 2004.
- [8] J.Dean, "Experiences with MapReduce: an Abstraction for Large-Scale Computation", In Keynote I: PACT 2006.
- [9] J.Quiane Ruiz, C. Pinkel, J. Schad, and J. Dittrich, "RAFTing MapReduce :Fast Recovery on the RAFT", IEEE International Conference on Data Engineering, 2011.
- [10]P. Costa, M. Pasin, "Byzantine Fault-Tolerant MapReduce: Faults are Not Just Crashes", IEEE International Conference on Cloud Computing Technology and Science, 2011.
- [11]Q.Zheng. "Improving MapReduce Fault Tolerance in the Cloud", IEEE International Symposium on Parallel & Distributed Processing and Phd Forum(IPDPSW), 2010.
- [12] W. Yong, "A first order approximation to the optimum checkpoint interval", Communication of the ACM, 1974.
- [13] W. Wei, Y. Liu, and Y. Zhang, " Checkpoint and Replication Oriented Fault Tolerant Mechanism for MapReduce", IEEE International Conference on Data Engineering .,2011.
- [14] Y.Wang, W.Lu, R.Lou, B. Wei, "Journal of Grid Computing",Volume 13,Issue 4, December 2015, pp. 587-604.
- [15] M.Bunjamin, I.Shadi, P. Maria, A. Gabriel, "Resource Management for Big Data Platforms", Springer, 2016.
- [16] Sorting 1PB with MapReduce: [http:// googleblog.blogspot.com/2008/11/sorting-1pb-with-mapreduce.html](http://googleblog.blogspot.com/2008/11/sorting-1pb-with-mapreduce.html).

Forensic Analysis of Residual Artifacts on CDH Storage

Myat Nandar Oo¹, Thandar Thein²

University of Computer Studies, Yangon¹, University of Computer Studies, Maubin²
myatnandaroo@gmail.com¹, thandartheinn@gmail.com²

Abstract

Hadoop Storage is increasingly used by consumers, business, and government, and can potentially store and process large amounts of data. With the maturing and wide usage of Hadoop Storage, there are more and more crimes in this environment. The retrieval of digital evidence from Hadoop Storage can be a challenge in forensic investigation, due to its complex infrastructure and, lack of location knowledge about digital evidences. As a consequence, forensic researchers are moving towards the investigation researches of locating and documenting the residual artifacts to trace the criminal activities of Hadoop Storage. The Cloudera Distribution Hadoop (CDH) is a popular Hadoop Storage Platform, providing users a cost-effective, and in some cases free with the ability to access, store, and process data. This paper proposes a forensic investigation framework for locating and discovering the residual artifacts that remain on the CDH Storage Server and attached client machine. The residual artifacts can provide the potential evidences for forensic examiners to extract the evidences, and reconstruct the crime scene.

Keywords- CDH, crime, forensics investigation framework, residual artifacts

1. Introduction

Hadoop Storage is increasingly used by government, businesses, and consumers to store and access a large amount of information. In Statista report [12], the Hadoop market was valued at 6 billion U.S. dollars worldwide in the year 2015. A number of companies became bundle Hadoop and related technologies into their own Hadoop distributions as the Hadoop Platforms. The three prominent Hadoop Platforms are MapR, Cloudera, and Hortonworks [10]. CDH, Cloudera's open source Hadoop Platform, is the most popular distribution of Hadoop and related projects [17].

The popularity of Hadoop Storage enables the criminal to conduct their activities on it for exploitation. With the growing use of Hadoop to tackle processing of sensitive data, a Hadoop could be a target for data exfiltration, corruption, or modification [11].

Hadoop is subject to exploitation by criminals, who may be able to use storage for criminal purposes, thus

adding to the challenge of growing volumes of digital evidence in cases under investigation.

Overcoming these investigation challenges, it is important to have a contemporary understanding of the location and type of residual artifacts left behind by file operations of storage service. The identification of potential data stores is an area that can impede an investigation. The paper [18] found out to identify potential artifacts that remain on the client devices and servers involving the use of Syncany as a private cloud storage solution supporting the Big Data Platform. The forensic researchers discovered the artifacts on client devices to identify the usage of Google Drive [6], Skydrive [7] and Dropbox [5].

It is important to have a rigorous methodology and a set of procedures for conducting forensic research on the emerging technical environments (such as Hadoop Platform and cloud computing). The forensic work of locating and documenting the forensically residual artifacts is also required to trace the criminal activities. These residual artifacts can provide the forensic practitioners in extracting the effective evidences for future forensics works.

Martin and Choo [1] presented an integrated conceptual methodology of digital forensic framework for cloud computing that consists of (i) Evidence source identification and preservation, (ii) Collection, (iii) Examination and presentation, and (iv) Reporting and presentation phases. This paper discovered the data remnants on client devices to identify the usage of cloud storage by applying their proposed forensic framework.

As far as I know, there are no publications concerned with the forensic investigation work on CDH Storage. This paper discusses the need for Hadoop Platform forensics and proposes the forensic investigation framework for CDH with the aim to discover what artifacts can be gathered from CDH. Organization of the paper is as follows: CDH Storage is explained in Section 2. The research questions and methodology for digital forensics are defined in Section 3. The CDH Storage Forensics Framework is presented in Section 4. Section 5 draws conclusions and describes future works.

2. CDH Storage

Cloudera was the first vendor to offer Hadoop as a package and continues to be a leader in the industry. Its Cloudera CDH distribution, which contains all the open source components, is the most popular Hadoop distribution. Cloudera is the best known player and market leader in the Hadoop space to release the first commercial Hadoop distribution. Cloudera, the global provider of the fastest, easiest, and most secure data management and analytics platform built on Apache Hadoop and the latest open source technologies, today announced that it is positioned as a leader in The Forrester Wave™: Big Data Hadoop Distributions, Q1 2016 report [10]. The Hadoop backlogs of CDH are useful to trace illegal usages and embody the crime scene. Obtaining these artifacts from log files could provide forensic examiners with valuable evidence.

The architecture of CDH Storage is shown in Figure 1. The targeted CDH Storage utilizes Hadoop 2.x architecture. The Resource Manager manages resources and allots the resources to the application. It has Scheduler and Application Manager Components. The Scheduler executes the scheduling function using the client applications resource requirements. The application Manager employs to accept job-submissions, exchanging-container to execute the specific Application Master and provides the service for restarting the Application Master container on failure. The Application Master has the responsibility of negotiating suitable resource containers from the Scheduler, tracking their status and monitoring. For launching containers, the Node Manager is engaged, where each can house a map or reduce task. Cloudera Manager is an end-to-end application for managing agent on each cluster.

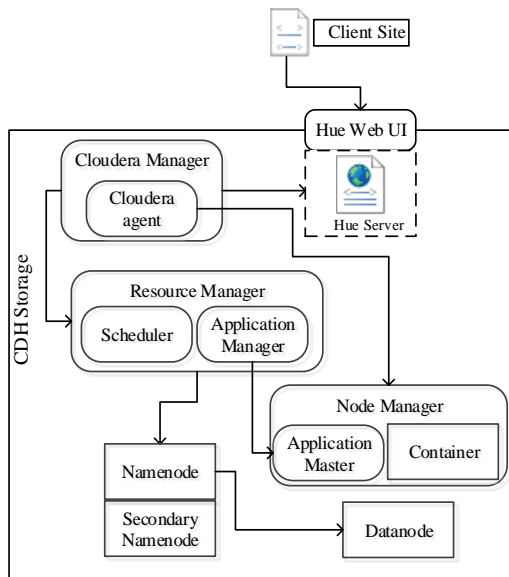


Figure 1: Architecture of CDH Storage

Conventional digital forensic methods are insufficient for investigating such composite infrastructure. Therefore, this paper proposes a forensic investigation framework for undertaking forensic research on CDH Storage. The resulting residual artifacts can provide effective evidences to the forensic examiners for future real-world CDH forensics.

3. Research Questions and Research Methodology

The identification of data remnants provides a better understanding of the types of artifacts that are likely to remain, and the access point(s) for digital forensics examiners to assist in the ‘identification’ stage of an investigation, which then follows with preservation and analysis. The ability to identify relevant data in a timely fashion can impact on an investigation by not including data that may be crucial to accurate findings in relation to circumstances; as such, the identification of data is an important part of the digital investigation process. The focus of this paper is to discover the residual artifacts left on CDH Storage and client machine.

3.1. Research Questions for Forensic Investigation on CDH Storage

For undertaking forensic research on the CDH storage environment, the following questions are examined:

- Q 1. What data remnants are likely to remain after the use of CDH?
- Q 2. What artifacts are created during the file operation on CDH storage?
- Q 3. What data are remained on client machine that are resulting from the use of CDH to identify its use?

3.2. Forensic Investigation Framework for CDH Storage

The proposed forensic investigation framework is based on the National Institute of Standards and Technology [3]. It comprises five phases; preparation, collection, analysis, and documentation and presentation, and closing as shown in Figure 2.

In the framework, the analysis phase can be cyclic and iterative as it is common that during an investigation a forensic examiner may need to return to a previous step.

1. Preparation: concerns with preparation of tools, techniques, research methodology, training, acquisition, and management support

2. Collection: includes collection and acquisition of data from identified sources and preserving the crime scene and data
3. Analysis: concerns with an in-depth systematic search, focuses on identifying and locating potential evidence
4. Presentation: concerns with completely and accurately documenting of findings and the residual artifacts
5. Closing: retains all related documentation recorded at each phase and review them to learn lesson for future real-world forensics

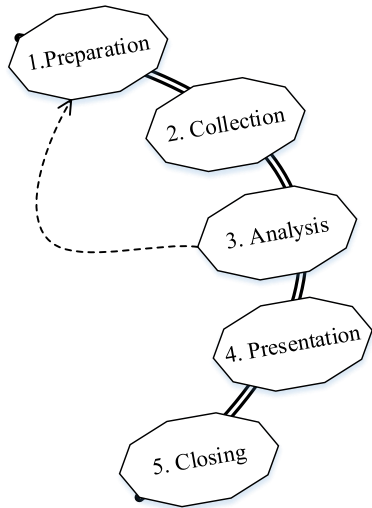


Figure 2: Forensic Investigation Framework for CDH Storage

4. Forensic Investigation Research on CDH Storage

This proposed framework finds out residual artifacts that are likely to remain after the use of CDH. These residual artifacts are useful to identify the action of a criminal. The remained artifacts can provide the potential evidences for forensic examiners to extract the evidences in exploring the illegal usages; what did the criminal do with CDH.

The investigation scope of this paper is to trace whether the suspected person connects the CDH server, and operate the primary file operations (upload, read and download) on the confidential data. Therefore, this section locates and discovers the residual artifacts on CDH Storage Server and attached the client machine to trace the file operations.

4.1. Preparation Phase

The targeted infrastructure is implemented and forensic tools are prepared to do the forensic research for gathering the artifacts that are likely to remain after the usage of CDH.

Table 1: Tools Prepared for Forensic Investigation on CDH Storage

Tool	Usage
FTK Imager Version 3.2.0.0 [9]	To create a forensic image of the .VMDK files.
dcfldd, dd version 1.3.4-1 [4]	To produce a bit-for-bit image of the .VMEM files.
Autopsy 3.1.1 [13]	To parse the file system, produce directory listings, as well as extracting/analysing the files, Windows registry, swap file/partition, and unallocated space from the forensic images.
SQLite Browser Version 3.4.0 [14]	To view the contents of SQLite database files.
Browser History Spy V-3.0 [15]	all-in-one software to instantly recover or view the browsing history from popular web browsers
WebBrowserPassView v1.56 [16]	the password recovery tool that reveals the passwords stored by the web browsers
File Viewer Plus 2 [8]	View, edit, save, and convert over the Hex files

The forensic tools for investigation both CDH Storage Server and the client machine are prepared as shown in Table 1. Testing environment and summary configurations of server and client are described in Table 2.

Table 2: System Configuration in Testing Environment

(CDH Storage)Server Configuration	Client Configurations
Operating System - Cent OS 6 Virtual memory size - 2GB Virtual HD size - 16GB Cloudera Version - Cloudera- quick-start VM - CDH 5.7.0 [2] IP address/ URL - http://hostname/8888	Operating System - Windows 7 64 bit Virtual memory size - 2GB Virtual HD size - 16GB Browsers - Mozilla Firefox 33.0.2 - IE 9.10.9200.16384, - Google Chrome 38.0.2125.111 m

4.2. Collection Phase

A digital forensic investigation is the ability to conduct analysis on a forensic copy, rather than interacting with or altering the original source. With the aim to adapt the nature of CDH, live forensic data collection is also needed. Data is copied in a forensic manner, using write-protection and creating a bit-for-bit copy. Secure collection of evidence is important to guarantee the evidential integrity and security of information. Files were identified which would contain the information needed to conduct the analysis; the virtual hard drives (VMDK files) in each Virtual Machine (VM) folder, each memory instance (VMEM files). These were identified for each of the VMs.

For the volatile data collection, imaging the memory of the server

“dd if=/dev/mem of=media/usb/memory.image”

The non-volatile data are collected by imaging the hard drive of the machine.

“dd if=/dev/sda | /media/usb/disk.image”

MD5 hash value of each file is calculated and verified with each forensic copy to ensure the integrity of the duplicated file.

4.3. Analysis Phase

For this research, each of the forensic copies of the hard drives, memory and VM image captures were examined using the prepared forensic analysis tools. This paper locates the forensically valuable files and extracts the residual artifacts from the large amount of backlogs,

metadata, registry and image. The residual artifacts on server and client machine can trace the criminal activities.

4.3.1. Evidential Analysis on CDH Storage Server

By analyzing the collected data, this investigation can track the footage on the CDH Storage Server to identify the usages. The usages include the primary file operations; upload, read and download. In this research, the residual artifacts related to each file operations are reported in Tables 3 through 5.

Among the large amount of backlogs and metadata of CDH Storage Server, the most important artifacts which can completely explore the primary file operation are discovered in the hdfs-audit.log and access.log files. The residual artifacts are the operated date, user name, file name, source IP, destination IP and file operation name.

Table 3: Residual Artifacts (File Uploading)

File name - hdfs-audit.log	Path - /var/log/hadoop-hdfs/ Artifacts - ¹ 2017-6-20 00:18:48,allowed = true ugi=admin ² src = /home/file.pdf ³ cmd=create ⁴ Type of artifacts - ¹ Date - ² User name - ³ File name - ⁴ Operation (create)
File name - access.log	Path - /var/log/hue Artifacts - ¹ 20/Jul/2017 20:41:27 ² 172.16.38.24 ³ admin – POST ⁴ ⁵ /filebrowser//dirBb/file.pdf ⁶ Type of artifacts - ¹ Date - ² Source IP - ³ User name - ⁴ Access method (POST for upload) - ⁵ File Path - ⁶ File name

Table 4: Residual Artifacts (File Reading)

File name - hdfs-audit.log	Path - /var/log/hadoop-hdfs/ Artifacts - ¹ 2017-6-20 00:18:48,allowed = true ugi=admin ² src = /home/file.pdf ³ cmd=getfileinfo ⁴ Type of artifacts - ¹ Date - ² User name - ³ File name - ⁴ Operation (getfileinfo)
File name - access.log	Path - /var/log/hue Artifacts - ¹ 20/Jul/2017 20:41:27 ² 172.16.38.24 ³ admin – GET ⁴ ⁵ /filebrowser//dirBb/file. pdf ⁶ Type of artifacts - ¹ Date - ² Source IP - ³ User name - ⁴ Access method (GET for read) - ⁵ File Path - ⁶ File name

Table 5: Residual Artifacts (File Downloading)

File name - hdfs-audit.log	Path - /var/log/hadoop-hdfs/ Artifacts - ¹ 2017-6-20 00:18:48,allowed = true ugi=admin ² src = /home/file.pdf ³ cmd=open ⁴ Type of artifacts - ¹ Date - ² User name - ³ File name - ⁴ Operation (open)
-------------------------------	--

File name - access.log	Path - /var/log/hue Artifacts - ¹ 20/Jul/2017 20:41:27 ² 172.16.38.24 ³ admin – GET ⁴ ⁵ /filebrowser//dirBb/ file.pdf ⁶ Type of artifacts - ¹ Date - ² Source IP - ³ User name - ⁴ Access method (GET for download) - ⁵ File Path - ⁶ File name
---------------------------	---

4.3.2. Evidential Analysis on Client Machine.

In order to analyze the vmdk image and collected data on the client machine, the prepared forensic analysis and recover tools are tested and applied. The aim of analysis is to explore what residual artifacts are left to identify whether CDH Storage Server was accessed via the web browser on the client machine.

The artifacts found on the client machine are URL, date, time, and file name. Moreover, the browser, the log files, the accessed web URL, the title of the website, the visited date and time are also identified. The web address of the server machine is also found in the browser cache file entries on the client machine. The forensically important files and residual artifacts of the popular web browsers; Mozilla Firefox, IE and Google Chrome are shown in Table 6.

Table 6: Important files and paths of Web Browsers

Mozilla Firefox 33.0.2	
Data	Path
Cache	%LocalAppData%\Mozilla\Firefox\profile\xxxx.default\cache2\entries
History	%AppData%\Mozilla\Firefox\profile\xxxx.default\places.sqlite %AppData%\Mozilla\Firefox\profile\xxxx.default\formhistory.sqlite
Cookie	%AppData%\Mozilla\Firefox\profile\xxxx.default\cookies.sqlite %AppData%\Mozilla\Firefox\profile\xxxx.default\permissions.sqlite
IE 9.10.9200.16384	
Cache	%LocalAppData%\Microsoft\Windows\TemporaryInternet Files\Low

History	%LocalAppData%\Microsoft\Internet Explorer
Cookie	%LocalAppData%\Microsoft\Windows\cookies
Google Chrome 38.0.2125.111 m	
Cache	%LocalAppData%\Google\Chrome\user data\default\cache
History	%LocalAppData%\Google\Chrome\user data\default\history
Cookie	%LocalAppData%\Google\Chrome\user data\default\cookie

4.4. Presentation Phase

According to the experiment, we found that a variety of data remnants were located when the user makes file operations on CDH.

The important files, paths, artifacts of storage server and attached client machine are documented. This information enables a practitioner to conduct forensic analysis and will assist to embody the criminal activity.

4.5. Closing Phase

The whole documentations are organized for later use. The collected data are stored in archived format. The forensic researcher reviews the tasks of each phase to extract which factors should be notice for the next investigation. The difficulties, solutions, usage of tools and all experiences of each step are reviewed for the preparation phase of the next investigations.

5. Conclusion and Future Works

The usage of Hadoop Storage is becoming more widespread. It is possible for malicious users to handle the illegal usages and the number of crimes on them has increased rapidly. This paper proposes a forensic investigation framework for locating and discovering the residual artifacts on CDH Storage Server and attached client machine. The residual artifacts can identify the use of CDH, trace the file operations and explore the illegal usages. The remained artifacts can provide forensic examiners in generating the effective evidences, and embody the criminal activity. The in-depth forensic analysis with crime scenario on CDH will also be presented in our later research. Future research opportunities also include conducting forensic research and exploring forensic methodologies for other Hadoop Distributions and Big Data solutions. And then, the development of log analysis model for forensic investigation of Hadoop Storage Platforms will also be a future work.

6. References

- [1] B.Martini and K.K.R. Choo, An integrated conceptual digital forensic framework for cloud computing. Elsevier-Digital Investigation, volume 9(2), pp. 71-80, 2012.
- [2] Cloudera Hadoop Distribution, Available: www.Cloudera.com, Accessed: September 3, 2017.
- [3] K. Kent, "Guide to Integrating Forensic Techniques into Incident Response," Special Publication 800-86, Computer Security Division Information Technology Laboratory National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.
- [4] Dcfldd Available: [http:// stefanoprenna.com/blog/2014/03/02/tutorial-how-to-use-dcfldd-instead-of-dd/](http://stefanoprenna.com/blog/2014/03/02/tutorial-how-to-use-dcfldd-instead-of-dd/) Accessed: September 3, 2017.
- [5] D.Quick and K.K.R.Choo, "Dropbox Analysis: Data Remnants on User Machines," Digital Investigation, vol. 10, no. 1, pp. 3–18, 2013.
- [6] D.Quick and K.K.R.Choo, "Google Drive: Forensic Analysis of Data Remnants," J Netw Comput Appl, vol. 40, pp. 179–193, 2014.
- [7] D.Quick and K.K.R.Choo, "Digital Droplets: Microsoft SkyDrive Forensic Data Remnants," Future Gener. Comput. Syst., vol. 29, no. 6, pp. 1378–1394, 2013.
- [8] FileViewerPlus Available: <http://fileviewerplus.com.siterankd.com/>. Accessed: Sept. 8, 2017.
- [9] FTK Imager, Available: accessdata.com/product-download/ftk-imager-version-3.2.0, Accessed: Sept. 8, 2017.
- [10] Report of Hadoop Big Data Distribution, Available: [https:// www.forrester.com/report/The+Forrester+Wave+Big+Data+Hadoop+Distributions+Q1+2016](https://www.forrester.com/report/The+Forrester+Wave+Big+Data+Hadoop+Distributions+Q1+2016). Accessed: September 3, 2017.
- [11] S.Acharya, J.Cohen "Towards a More Secure Apache Hadoop HDFS Infrastructure," in Network and System Security, Lecture Notes in Computer Science Volume 7873, 2013, pp 735-741.
- [12] S. V. President, "Hadoop/Big Data Market Size Worldwide 2015-2020 | Statistic," Statista, 2016. Available: <https://www.statista.com/statistics/587051/worldwide-Hadoop-bigdata-market/>. Accessed: Nov. 8, 2016.
- [13] Sleuthkit Autopsy, Available: [https:// www.sleuthkit.org/autopsy/download.php](https://www.sleuthkit.org/autopsy/download.php), Accessed: Sept. 8, 2017.
- [14] SQLite Database Recover, Available: <https://www.stellarinfo.com/sqlite-repair.php> , Accessed: Sept. 8, 2017.

[15] Web Browser Pass View, [https:// www securityxploded.com/ browser-password-decryptor.php](https://www.securityxploded.com/browser-password-decryptor.php), Accessed: Sept. 8, 2017.

[16] Webbrowserhistoryspy: Availabe: [http:// www.nirsoft.net](http://www.nirsoft.net), Accessed: August. 4, 2017.

[17] Top 6 Hadoop Vendors providing Big Data Solutions in Open Data Platform Available:<https://www.dezyre.com/article/top-6-hadoop->

[vendors-providing-big-data-solutions-in-open-data-platform/93](#), Accessed: August. 4, 2017.

[18] Y.Y.Teing, A.Deqhantan, K.K.R.Choo, Z.Muda, M.T.Abdullah and W.C.Chai, "A, Closer Look at Syncany Windows and Ubuntu Clients' Residual Artifacts ", in Security, Pravity and Anonymity in Computation, Communication and Storage, Springer International Publishing, 2016, pp.342-357.

Networking and Network Security

Stateful Firewall Application on Software Defined Networking

Nan Haymarn Oo, Aung Htein Maw

University of Computer Studies, Yangon , University of Information Technology
nanhaymanoo@ucsy.edu.mm, ahmaw@uit.edu.mm

Abstract

Software defined networking(SDN), one of the advanced networking technology, is more flexible, faster and more secure than traditional network technology because of using OpenFlow in separating the network control operations from the forwarding devices to develop networks. However, many security issues still exist in the SDN architecture. To solve these issues, firewall is one of the most important SDN security applications. Firewall can be generally categorized into stateless and stateful depending on the capability of connection state tracking. Stateless firewall does not check the states of the connection session. As a result, it has some limitations in solving the security issues. But this limitation can be overcome by stateful firewall. An acl application has been implemented in Open Network Operating System (ONOS) as a stateless firewall. In order to increase the security level of acl application in ONOS, this paper proposes a stateful firewall. The firewall application is implemented by taking into account both security and performance perspectives in order to be a proper SDN security application. Finally, the experiment will be conducted how proposed stateful firewall tackle the security issues and affect the performance by comparing with acl application.

Keywords- Stateless Firewall, Stateful Firewall, SDN, ONOS

1. Introduction

SDN is a network with three-tier architecture: data plane, control plane, and application plane. SDN switches existing in data plane do not have intelligent - i.e., how to transmit packet among them. Thus they pass the packet to the controller in control plane. The controller passes the packet again to the respective application. The application produces flow rules for the packet and installs them into the switches via controller.

Open Vswitch is commonly used as forwarding device. For firewall application, it can act as a stateless firewall because it has OpenFlow table including the fields such as source MAC address, destination MAC address, source IP address, destination IP address, source Port, destination Port, action and count. But it does not have state field and inspect the state of the packet.

Therefore, packets are needed to send to the controller in order to check the state of the packet in implementation of the stateful firewall application.

Firewall application sets the flow rules depending on the firewall rules set. It can be implemented to install necessary flow rules either by itself or by using the help of forwarding application. The forwarding application can be differentiated into two types: forwarding packets within the same network, and routing packets among the different networks. This paper only emphasizes on the packet forwarding within the same network. Hence, the stateful firewall application proposed in this paper uses the reactive forwarding application as its assistant in installation of flow rules.

According to the stateless and stateful firewall, flow rule installation depends on whether it inspects the connection state or not. For stateless firewall application, it makes decision of flow rules installation by considering only the firewall rules set. But, stateful firewall application takes into account both its firewall rules set and state of each packet when setting the flow rules into switches. Therefore, stateful firewall can be protected the attacks more than stateless one.

The acl application in ONOS controller installs flow rules with different manners according to the action of acl rule as shown in figure 1. The application solely installs permanent drop flow rules for deny action acl rules in the source switch. For allow action acl rules, the application installs permanent flow rules with To Controller output action. Such flow rules installation means that the acl application delegates the packet forwarding to the default fwd application running in controller. The forwarding application installs temporary forwarding flow rules. It removes the installed flow rules after flow timeout value expire. It does not consider any state of the connection session during the flow rules installation and removing.

Therefore, in order to overcome the main limitation of acl application - lack of connection state inspection, this paper proposes stateful firewall application by adding connection state inspection into the acl application. State-aware application has to send packets to controller to check state of the connection session, though, the proposed stateful firewall application only send necessary packet according to both connection tracking and SDN nature in order not to reduce performance while increasing security level.

As ONOS controller[1] has distributed and topology-aware nature, its acl application has tackled the problems such as single point of failure occurs in centralized firewall, less sensitive to topology change, complicated firewall configuration, additional cost in rule maintaining, and longer rule matching time that may occur in simple distributed firewall.

The rest of this paper is organized as follows: section 2 lists the related work of this paper. Section 3 presents stateless vs stateful firewall. And section 4 introduces the process of proposed stateful firewall application. Section 5 describes the testbed for experiment. Section 6 evaluates the results of the proposed stateful firewall application and existing acl application. Section 7 describes conclusion and future works.

2. Related Work

Software-Defined Network (SDN), new modern network, has many security issues. In order to countermeasure them, firewall application has been implemented on every type of controller. Two types of firewall with inefficient and efficient ways has been created on Ryu[2]. [3] proposed reactive stateful firewall with a global orchestrator on Ryu controller. Stateless firewall application also has been implemented on Floodlight[4]. And its stateful firewall application has been researched[5]. Likewise, acl application is written in ONOS controller. It can be said this application as stateless firewall because it filters the packet by using the rules that are the same as stateless firewall rules including source IP, destination IP addresses, source port, destination port and action. Stateful firewall on ONOS controller has proposed and researched to be able to early block the DDOS attack especially for the Internet Service Provider Network[6].

Early firewall applications are implemented as centralized applications according to the SDN architecture. But the centralized applications have weaknesses such as single point of failure, controller overhead, and overloaded communication between control plane and data plane. In order to overcome those weaknesses, firstly, distributed firewall applications have been proposed[7]. Although these applications can tackle the single point of failure problem, they can cause further problems: less sensitive to topology change, complicated firewall configuration, additional cost in rule maintaining, and longer rule matching time. Therefore, later firewall applications are implemented as a topology-aware selectively distributed firewall[8-9].

In this paper, stateful firewall application is mainly implemented based on the ONOS acl application in order to reduce communication between control plane and data plane, and controller overhead while adding connection state inspection.

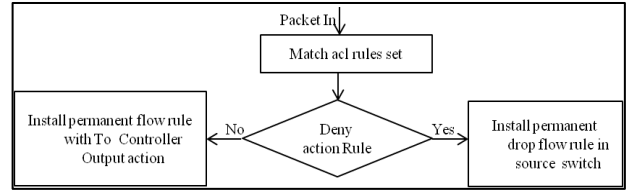


Figure 1. Flow chart of overall acl application

3. Stateless Firewall vs. Stateful Firewall

Stateless firewall or packet filtering firewall allows or denies the packets according to the action defining in firewall policy or firewall rules set. The filtering rule in firewall policy includes layer-3 and layer-4 information such as source IP, destination IP, destination port, protocol type and action. It does not inspect the state information including in the packet. Consequently, it may allow the SYN_ACK packet without sending its prior SYN packet.

Stateful firewall itself can filter such packet because it not only filters the incoming packets by matching them with firewall policy like stateless firewall but also checks the state of connection session. The connection state is described by TCP flag value which represents as the code bit in TCP header. Table 1 and table 2 show the binary code bit order and list of TCP flag respectively[10]. Those tables only present the flag types used in this paper.

Normally, the TCP packet is transmitted sequentially from SYN packet, SYN_ACK packet, ACK packet vice versa between source and destination. Sending the three packets establishes connection by three-way handshake. Finally, FIN_ACK packet is sent to the destination to close the connection formally. FIN flag cannot be seen in the flow because the packet with ACK flag is needed while sending FIN flag to the destination for informing both confirmation of the previous received packet and termination of the connection as in Figure 2.

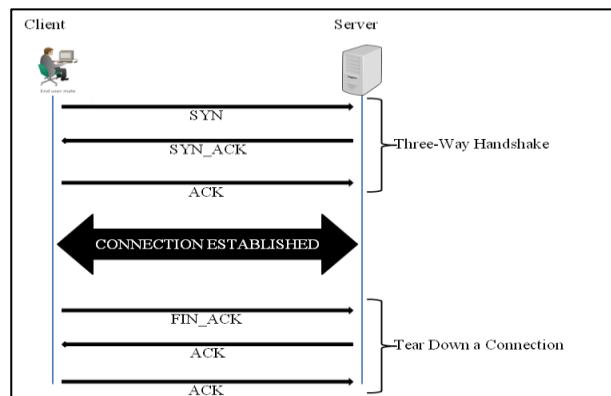


Figure 2. TCP Packet Sequence

Table 1. Binary code bit order for TCP flag

Code	1	2	3	4	5	6
Name	Urgent Pointer	ACK	PUSH	RST	SYN	FIN

Table 2. TCP flag

Flag	Code bit			Description
	Binary	Hexa	Dec	
SYN	000010	0x002	2	First segment of a new TCP connection
SYN_ACK	010010	0x012	18	Return acknowledgement for received SYN
ACK	010000	0x010	16	Acknowledge the successful receipt of packets
FIN_ACK	010001	0x011	17	Tear down the created virtual connection
RST	000100	0x004	4	Reject the request that is not intended for current connection

4. Proposed Stateful Firewall

The proposed stateful firewall application is implemented by adding connection state inspection into the packet forwarding operation of acl application. As state inspection cannot be done by forwarding devices such as Open Vswitch, all packets must be sent to the controller. Consequently, overloaded communication between control layer and data layer, and controller overhead problems are caused. To solve the consequence problems, the application sends only the essential packets to the controller by considering both the nature of connection tracking and SDN security.

In SDN environment, only the first packet of a flow is sent to the controller in order to get the flow rule. But it is also important to know the last packet or the packet contained FIN flag of TCP protocol for checking the teardown of connection session. Moreover, according to the connection tracking nature of TCP protocol, both initialization (SYN) and termination (FIN) of a connection session trigger from the source host. Thus, only packets come from source host have to be inspected carefully. By considering the combination of these factors, the stateful firewall application installs flow rule with group action in source switch and single output action in other switches.

Group action in this paper is the combination of output port action and To Controller action. The flow rule with this action sends packet to both destination and

controller concurrently so that the controller can receive all packets including the last packet with FIN flag and terminate the connection by removing installed flow rules from the switches and deleting added state records in the state table immediately. As the flow rule with group action overloads the communication between control plane and data plane, it is only applied in essential source switch where the FIN packet comes from.

In addition, this application installs flow rules temporarily with timeout value like acl application but removes them as soon as the connection is terminated. When the connection is tearing down improperly, the controller will not receive the FIN packet and it will remove the flow rules after the timeout value expire. This fact is the main difference between acl application and this stateful firewall application. While acl application installs and removes flow rules temporarily without checking any state of the connection session, this application installs and removes them depending on the connection state.

From the SDN security point of view, attack packets can enter via the host connected switches and they must be protected tightly at those type of switches so that the attack cannot enter into the remaining network area. The host connected switches are source switch and destination switch. However, this application inspects the state of connection session at the source switch because it is safe enough to check one packet once at the entrance. Thus, firstly, the application in controller checks where the packet comes from. If it comes from the destination switch or intermediate switch, the application installs flow rule without checking any state of the connection session. Otherwise, connection state of the packet is checked by the application in order to make decision whether it installs flow rules or not for this packet in source switch.

If the packet does not pass through the controller as a TCP packet sequence, the application installs drop rule for this packet in the source switch. By this way, the firewall protects the possible attacks that breach the TCP protocol.

The connection state inspection process of proposed firewall application is shown in figure 3. For the packet with SYN flag or SYN_ACK flag comes from the destination switch or intermediate switches, the controller installs forwarding rule for it without checking the connection state. Thus, this figure does not consider the packet comes from those switches and it is especially for the inspection of incoming packet from source switch.

In this paper, default forwarding application in ONOS is used for forwarding packet. Hence, connection state inspection function is added into it. This application can be used to forward packet within the same network. Therefore, MAC address is used in state table construction instead of IP address. In addition, since only

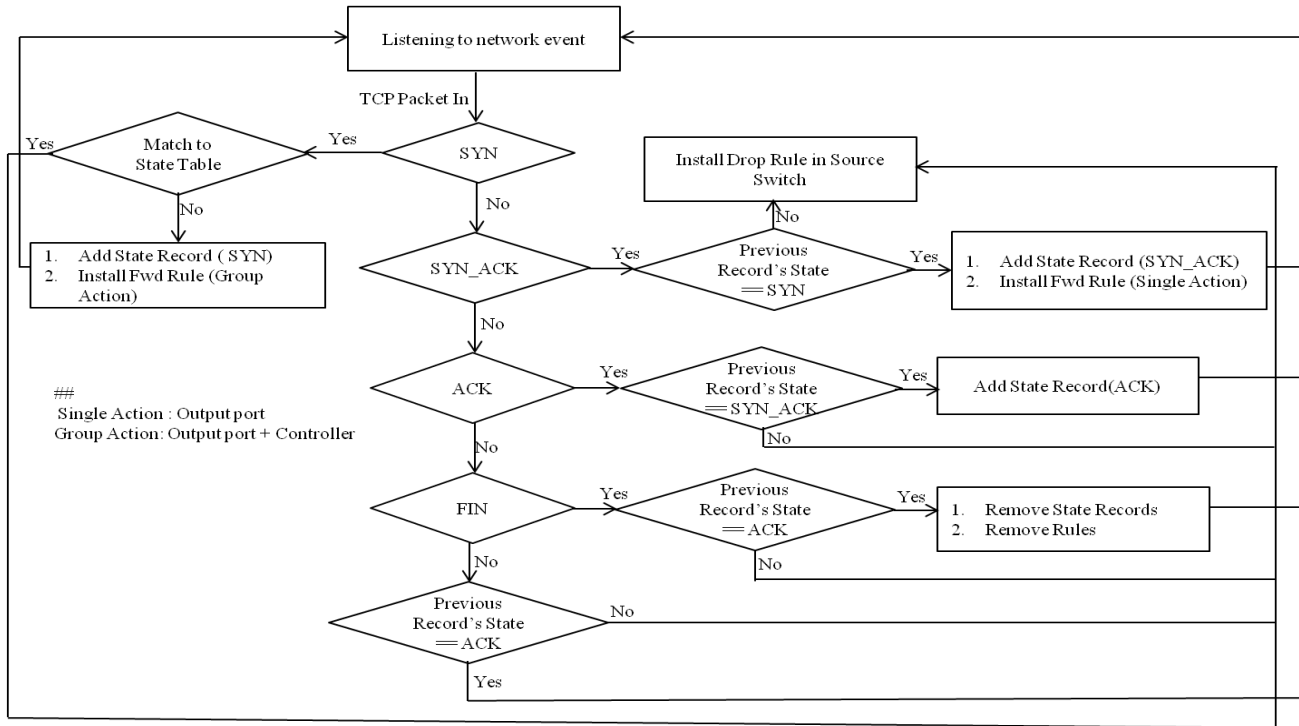


Figure 3. Flow chart of connection state inspection

the TCP protocol is stateful protocol, this paper presents the tracking of connection state for solely TCP when implementing stateful firewall.

5. Experimental Testbed

The testing is performed by using mininet emulator[11] with OpenFlow version 1.3[12] and ONOS controller. Both of them are running on Dell Desktop PC with Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz, 64 bits and 4 GB memory. The performance is measured on the linear topology of open virtual switch (OVS) with one host per switch. In order to compare the performance level, the acl application and stateful firewall application use the same linear topology.

6. Evaluation

As every security application has trade-off between security level and performance, this paper evaluates the two applications with two parts. Security level is measured by showing the filtering result of stateful firewall. Since stateful firewall tracks the network connection when filtering packets, the filtering results are shown by the log of filtered accessing information together with its state records. In addition, performance level is compared by taking the concurrent downloading time from web servers.

6.1 Filtering Result of Stateful Firewall

In this section, we experiment using a linear topology with three switches and three hosts in mininet and define

the acl rules set as shown in figure 4 and table 3 respectively. As this paper emphasizes solely on TCP protocol, the acl rules set is defined for only TCP. According to the rules set, all hosts are not allowed to access TCP protocol with port number 80 except those between host 1 and host 3.

Table 4 shows the flow rules of switch s1 installed by acl application together with its assistant, reactive forwarding application. According to the acl rules set in Table 3, acl application installs one drop rule and two allow rules with action To Controller.

Thus, the forwarding application takes responsible for installation of flow rules for allowed packets. It installs forwarding rule with output action to the destination port and group action to both controller and destination port concurrently in order to catch the connection termination packet with FIN flag from controller while sending packets of a flow to their destination especially in source switch.

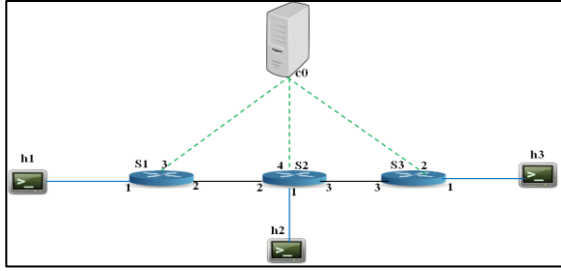


Figure 4. Linear Topology

Table 3. Acl rules set

Src Mac	Dst Mac	Proto	Dst Port	Action
10.0.0.1	10.0.0.3	TCP	80	Allow
10.0.0.3	10.0.0.1	TCP	80	Allow
10.0.0.0/24	10.0.0.0/24	TCP	80	Deny

Table 4. Flow rules in switch s1

Src Mac	Dst Mac	Proto	Dst Port	Action
10.0.0.1	10.0.0.3	TCP	80	Controller:65535
10.0.0.3	10.0.0.1	TCP	80	Controller:65535
10.0.0.1	10.0.0.3	TCP	80	group :1
10.0.0.3	10.0.0.1	TCP	80	output:1
10.0.0.0/24	10.0.0.0/24	TCP	80	drop

In figure 5, since only host 1 and host 3 are allowed to use TCP traffic with port 80 in defined acl rules set, host 1 can access web server in host 3 and cannot connect another web server in host 2.

Figure 6 shows the log of both connection state records for connection establishment and its termination from client host 1 to web server host 3. For each TCP flow, the application inspects the incoming packets according to the inspection process as shown in the flow chart of figure 3 and recognizes three records for SYN(2), SYN_ACK(18), ACK(16) after it has established a connection session. The fields included in each state record are source Mac, destination Mac, source port, destination port, and TCP flag. The SYN_ACK record is stored after swapping the source and destination of Mac and port in order to map the records easily. When the connection termination packet with FIN flag comes to the controller, the application removes the installed flow rules and deletes the state records of the respective connection.

The stateful firewall application drops an abnormal TCP flow as shown in figure 7. An abnormal TCP packet is sent from client host 1 to web server host 3 by using hping3[15]. TCP traffic with port 80 between host 3 and host 1 is allowed in acl rule, but the application drops the

packet with SYN_ACK flag from host 1 to host 3 because of its abnormal state. The packet with SYN_ACK flag comes without its prior packet with SYN flag. Thus, the application assumes its abnormal packet and installs drop rule to block it as shown in figure 8.

6.2 Latency Result

Measurement of latency for TCP is performed on the increasing number of simultaneous connection (10 to 100) by setting up web servers depending on the number of TCP connection and one host accesses the servers at the same time. The web servers are created by using SimpleHTTPServer in mininet hosts and parallel download HTTP requests are sent from the client host with combination of xargs[13] and wget[14] command. This command uses web server url list while sending parallel downloading requests to web servers.

In order to get a thorough latency result, we conduct our experiment with two types of flow: long lived flow and short lived flow. We download a 277MB file from servers for long lived flow. For short lived flow, we only access the web page from web servers and the size of the web page is 2.4KB.

We can examine the latency for long lived flow from figure 9 that the download time differences between acl application and stateful firewall application vary from 0.2s to 7.87s for the number of simultaneous connection from ten to one hundred. The average times recorded from stateful firewall application are in the range of 0.26s to 37.93s while these of acl application are 0.06s to 34.1s.

```

"Node: h3"
root@hayman:~/TCPTest# python -m SimpleHTTPServer 80 &
[1] 12646
root@hayman:~/TCPTest# Serving HTTP on 0.0.0.0 port 80 ...
10.0.0.1 - - [12/Sep/2017 11:40:01] "GET / HTTP/1.1" 200 -

"Node: h1"
root@hayman:~/TCPTest# # wget -O - 10.0.0.3
--2017-09-12 11:40:01-- http://10.0.0.3/
Connecting to 10.0.0.3:80... connected.
HTTP request sent, awaiting response... 200 OK

"Node: h2"
root@hayman:~/TCPTest# python -m SimpleHTTPServer 80 &
[1] 12645
root@hayman:~/TCPTest# Serving HTTP on 0.0.0.0 port 80 ...

"Node: h1"
root@hayman:~/TCPTest# # wget -O - 10.0.0.2
--2017-09-12 11:44:44-- http://10.0.0.2/
Connecting to 10.0.0.2:80... failed: connection timed out.

```

Figure 5. Accessing information

```

2017-09-12 11:40:01,575 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | TCP Flag :16
2017-09-12 11:40:01,575 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | Connection state record: 00:00:00:00:01 00:00:00:00:03 60674 80 2
2017-09-12 11:40:01,575 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | Connection state record: 00:00:00:00:01 00:00:00:00:03 60674 80 18
2017-09-12 11:40:01,575 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | Connection state record: 00:00:00:00:01 00:00:00:00:03 60674 80 16
2017-09-12 11:40:01,576 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | TCP Flag :24
2017-09-12 11:40:01,578 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | TCP Flag :16
2017-09-12 11:40:01,578 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | TCP Flag :17
2017-09-12 11:40:01,579 | INFO | ew I/O worker #6 | ReactiveForwarding | 176 - org.onosproject.on
s-app-fw - 1.8.0.SNAPSHOT | Successful Flows have been removed

```

Figure 6. Log for connection state

```

"Node: h1"
root@hauaman:~# sudo hping3 -d 120 -S -a -u 64 -p 80 -i 1 10.0.0.3 -c 1
HPING 10.0.0.3 (hi-eth0 10.0.0.3): [SA set], 40 headers + 120 data bytes
--- 10.0.0.3 being statistic ---
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

```

Figure 7. Abnormal TCP packet

```

s-app-fw - 1.8.8.SNAPSHOT | TCP Flag :18
2017-09-12 12:12:02,665 | INFO | ew I/O worker #6 | ReactiveForwarding | 181 - org.onosproject.on
s-app-fw - 1.8.8.SNAPSHOT | SYN ACK comes without SYN
2017-09-12 12:12:02,665 | INFO | ew I/O worker #6 | ReactiveForwarding | 181 - org.onosproject.on
s-app-fw - 1.8.8.SNAPSHOT | Block the traffic from 00:00:00:00:00:00 to 00:00:00:00:00:00

```

Figure 8. Log for blocking abnormal TCP packet

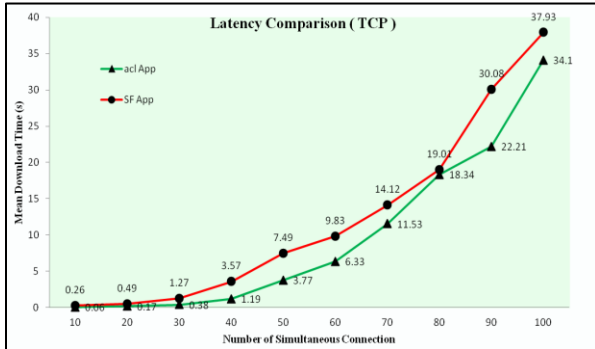


Figure 9. Latency result for long lived flows

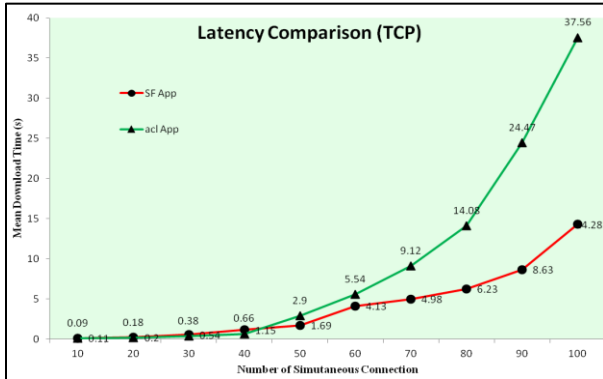


Figure 10. Latency result for short lived flows

In order to get the average delay time between these two applications, we calculate the time difference for each measurement step of their downloading time. Finally, we get the mean delay time, 2.6% by stateful firewall application.

Figure 10 shows the latency results for concurrent short lived flows. The acl application's average downloading time is more increase than stateful firewall's one when the number of simultaneous connections is above 40 because stateful firewall application uninstalls flow rules as soon as their respective connection is terminated while acl application is maintaining many flow rules without expiring their timeout value.

7. Conclusion

In order to keep higher security level in SDN, this paper proposes the stateful firewall application based on acl application in ONOS. Hence, the acl application becomes more secure due to the fact that it is enhanced from stateless firewall application into stateful firewall. However, the performance of stateful firewall is affected by added security operations. On the other hand, it is effected by removing flow rules immediately after terminating connection. By observing the latency comparisons in this paper, the average latency increased by stateful firewall application is 2.6% in long lived flow, though, the application increases the performance up to 8.5% when short lived flows pass through it. Therefore, we believe the proposed stateful firewall application is a proper SDN security application. We will implement and test the stateful firewall for not only TCP but also UDP and ICMP that can filter among different networks and improve the performance while preserving the security.

8. References

- [1] ONOS controller, <http://www.onosproject.org>.
- [2] Cornelius Diekmann and Florian Wohlfart, "Implementation and Performance Analysis of Firewall on Open vSwitch". Master's thesis. Technische Universitat Munchen, 2015.
- [3] S. Morzhov, I. Alekseev, and M. Nikitinskiy, "Firewall application for Floodlight SDN controller", International Siberian Conference on Control and Communications (SIBCON), 2016- May-12, pp. 1-5.
- [4] S. Vasudevan, "Firewall A New Approach to Slove Issues in Software Define Networking", 6th International Conference on Emerging trends in Engineering and Technology (ICETET'16), ISSN:2248-9622, pp.14-19.
- [5] Salaheddine Zerkane, David Espes, Philippe Le Parc, Frederic Cuppen,. "Software Defined Networking Reactive Stateful Firewall", 31st International Conference on ICT Systems Security and Privacy Protection. Springer, Ghent, Belgium. 471, May 2016, pp.119-132.
- [6] Andis Arins, Riga, and Lativa, "Firewall as a service in SDN OpenFlow network, 2015.
- [7] Justin Gregory V. Pwna nd William Emmanuel Yu, "Development of a Distributed Firewall Using Software Defined Networking Technology", 2014.
- [8] Thuy Vinh Tran, Heejune Ahn, "A Network Topology-aware Selectively Distributed Firewall Control in SDN", 2015.

[9] Thuy Vinh Tran, Heejune Ahn, "FlowTracker: A SDN Stateful Firewall Solution with Adaptive Connection Tracking and Minimized Controller Processing", 2016.

[10] Administrator, "TCP Flag Options - Section 4."Internet: <https://www.firewall.cx/networking-topics/protocols/tcp/136-tcp-flag-options.htm>.

[11] Mininet Network Emulator, <http://mininet.org>.

[12] OpenFlow Switch Specification, version 1.3.5, <https://www.opennetworking.org/>, 2015-March-26.

[13] Xargs command, Internet: <http://man7.org/linux/man-pages/man1/xargs.1.html>.

[14] GNU Wget 1.18 Manual, Internet: <https://www.gnu.org/software/wget/manual/wget.html>.

[15] Hping3 Security Tool, Internet: <http://www.hping.org/hping3.html>.

An Analysis of Decision Tree Based Intrusion Detection System

Yi Yi Aung, Myat Myat Min
University of Computer Studies, Mandalay
yiyiaung123@gmail.com, myatiimin@gmail.com

Abstract

Intrusion detection is the process called indentifying intrusions. The action of entering to a system without permission is called intrusion. With the improving advanced technology of mobile devices such as smart phones, tablet, smart devices, other computing devices, the number of network users are increasing more and more. Hence, security on network is very important for all net consumers. IDS are fundamental part of security boundary. So, they are now considered as a mandatory safety mechanism for critical networks. There are many traditional techniques of intrusion detection. In the research of traditional intrusion detection technology analysis, the statistical model for the establishment of the regulatory basis, management and aggression capability and so on there are still some disadvantages and disabilities, because actual test results cannot meet the requirements. Current methods used in IDS are many. Each method has advantage and disadvantage. Intrusion detection can also be seen as a classification problem. In this research we use K-means and C4.5 algorithms. This paper presents the comparison of intrusion detection by using hybrid data mining methods and a single method. The purpose of this paper is to show the differences of time complexity between hybrid data mining method and a single method. This model is verified using KDD'99 data set. Experimental result clearly shows hybrid methods can reduce model training time while maintaining the higher detection rates than using single method.

Keywords- Intrusion Detection System, KDD'99 dataset, K-means, C4.5

1. Introduction

People want to keep their possessions secure. So they consider many ways to secure their possessions and then invented many software and hardware devices to protect their belongings. There is no secure system in world but we must consider and protect our system as much as we can.

The computer system controls a large amount of data over the network, so data communications should be secure enough for data transceivers. In the previous day's firewall, data encryption, antivirus is used to prevent unauthorized access to the network system. There is a

technique known as the intrusion detection system that will be used to monitor unwanted user data over a network. [4]

Cyber security is a critical topic for both researchers and practitioners as successful cyber attacks can result in severe costs due to losses of confidentiality, integrity or availability. Various security mechanisms have been suggested for detecting cyber attacks such as intrusion detection system. Intrusion detection system is sometimes called classification problems.

Intrusions in a computer system are the activities that infringe the system security policy. The process of monitoring and analyzing the activities occurring in the process of Intrusion Detection process is in a deep way. [1]

Detection of intrusions identifies an unauthorized individual to use a computer, and identifies an authorized individual who abuse their power. The intrusion Detection System is an important defense tool for network security. By analyzing audit data, the Intrusion Detection System tells the administrator to take appropriate action to protect the application system from further attacks when the system is in unsafe condition. Because network intrusion is a set of human behavior and user behavior, it can be divided into normal and abnormal behavior. [2]

Many security experts and researchers have proposed and implemented different strategies to defend the computer system from attacks. Among them the intrusion detection systems (IDSs) can be specifically designed to identify attacks that can target computers or networks and resources. IDS have two main components: data audit component, sensors or log files, which monitors / collects data on system behavior; a component detection method that analyzes data that is observed/collected to detect malicious activity. In terms of audit components, IDS is classified as host-based (HIDS) or network-based (NIDS). Host-based IDS detects attacks on computer system by monitoring mainly operating system activity. Network-based IDS detects nodes connected to the network by monitoring TCP/IP events. In terms of detection methods, IDS is further classified as signature or anomaly tracking system. Signature-based systems monitor traffic to known attack patterns (signatures), similar to virus scanners that protect personal computers. Signature-based IDSs efficiently detect existing threats but always miss new threads.

Finally, KDD CUP 1999 dataset is used to verify the effectiveness of our method. The experimental results

show that the collaborative intrusion detection and customization methods proposed in this paper are superior to C4.5 detection in tracking accuracy and tracking efficiency.

2. Literature Survey

Many data mining algorithms are applied to intrusion detection. This is divided into general offline algorithms and inline incremental algorithms. Most researchers have focused on off-line intrusion detection using a well-known KDD99 dataset and verified the development of IDS. The KDD99 dataset is a statistically preprocessed dataset that has been available since 1999. [17]

Classification is one of the important functions of data mining. It performs its task by classifying the data into different categories using the algorithm type. This classification has wide range of applications in the field of network intrusion detection. It categorizes network patterns as normal or attack to identify malicious activities occurring in the network. In this paper, researcher is analyzing classification algorithms using NSL KDD 99 dataset. [9]

Intrusion detection is one of the difficult problems encountered by the modern network security industry. Data mining can play an important role in system development. Data mining is a technique for extracting important information from a huge data repository. In order to detect intrusion, the traffic created in the network can be broadly categorized into following two categories-normal and anomalous. In this proposed paper, several classification techniques and machine learning algorithms have been considered to categorize the network traffic. The comparison of data mining algorithms has been performed using WEKA tool and listed below according to certain performance metrics. Simulation of these classification models has been performed using 10-fold cross validation. For this simulation, they used NSL-KDD based data set in WEKA. [3]

With the rapid development of computer networks during the past decade, security has become a key issue for computer systems. It is the IDS which protect our computer network. Different classification and clustering algorithms have been proposed in recent year for the implementation of intrusion detection systems. In this paper, multiple algorithms are analyzed to find the optimal algorithm. At last the optimal algorithms Random Forest and DB Scan are occurred. [5]

In this paper, the various intrusion-detection-system techniques and their application on the basis of decision trees also discussed that are available and on which various researches have made. Some detection approaches that are applied for the intrusion detection are focused on some specific methods. In this paper various intrusion detection approaches are analyzed for detection of intrusion by the use of decision tree algorithm. [6]

Intrusion is an act that violates the security policy of the system. The purpose of this research paper is to explain the method / technique used for intrusion detection based on the concept of data mining and the structure designed for it. This survey document outlines the intrusion detection process and the data mining methods and methods to facilitate the frames developed using these concepts. [10]

Since the ready-made data mining algorithms is offered, intrusion detection based on the data mining has improved rapidly. It advances in the ability to hold enormous data, but it also has troubles like, for instance, searching for more helpful data mining algorithms, how to progress the correct rate of intrusion detection, and etc. These can be the topic for future study; meanwhile they also need lots of effort and experiments to develop a system that is more effective and more suitable. There are many types of approaches in intrusion detection, in which that based on the data mining becomes the hot topic in the current intrusion detection methodology. However, data mining is still in its developing stage, so more thorough study needs to be done. A brief survey of the IDS in the data mining field is given in this paper. [12]

In this paper, they present an intrusion detection system using J-48 and Naïve Bayes for classification. To implement and classify of the system they used KDD 99 dataset and their University's traffic. The principal challenge in intrusion detection is to obtain high detection rate. From this paper's experimental result shown as single classifier is not sufficient to obtain the high result and feature selection is the most important to detection ratio also showed that the effectiveness of J-48 is comparable to the Naïve Bayes. [13]

This paper draws the conclusions on the foundation of implementations performed using various data mining algorithms. Combining more than one data mining algorithms may be used to eliminate disadvantages of one another. Thus a combining approach has to be made while selecting a mode to apply intrusion detection system. Combining a number of qualified classifiers lead to a better performance than any single classifier. [15]

3. Intrusion Detection System and Data Mining

Monitoring user activity on the network and categorizing malicious and normal activity is called intrusion detection. The system used for this purpose is called Intrusion Detection System (IDS). Intrusion detection systems are combinations that perform software or hardware, or automated processes that track and analyze events. In general, IDS monitors and records computer system events, performs to determine if an event is a security incident, alerts potential threat to security employees, and generates event reports. Every

time an intruder attempts to compromise the confidentiality, integrity or availability of the network or system, IDS monitors and detect illegal activities, prohibit legitimate users to access resources or computer to system services. It also takes appropriate predefined expectations into account and performs appropriate actions.

The intrusion detection techniques can be defined as a system that identifies and deals with malicious use of computer and network resources. The IDS using its detection techniques tracks the user available on the network and traces the activities being carried out. The audit data after being traced are compared to the known awful records and an alarm is set whether the similarities of the two are above some predefined threshold. The user is then accordingly distinguished as normal or illicit user.

The IDS techniques on the basis of their detection process can be categorized into two methodologies:

Misuse Detection technique:

The misuse detection technique uncovers the intruder activities based on the extensive knowledge of known patterns provided by human experts. The detection process involves matching features through the attacking feature library and confirming the attack incidents. The key advantage of misuse detection system is that once the patterns of known intrusions are stored, future instances of these intrusions can be detected effectively and efficiently. The detection process though can even catch the negligible intrusive activities and generates the much fewer false alarms but still is unable to detect novel or unknown attack. [11]

Anomaly Detection technique:

Anomaly detection technique analyses the intrusive activity and identifies the new intrusion types according to the deviation of a computer from its normal usage. If the divergence is much enough then the user activity is considered as abnormal. The key advantages of anomaly detection systems are that they can detect unknown intrusion since they require no a priori knowledge about specific intrusions. Anomaly detection although reveals the new trends of intrusions still lacks the detection of negligible intruder activities and also generates higher false alarm rate.

After focusing on the IDS techniques, the IDS based on its analysis and audit data storage unit are of two types:

Host-based IDS (HIDS):

HIDS is a host based detection approach in which a system collects the data as the records of various activities of host including event logs, system logs etc. As the system monitors only the host or agent it determines the awfulness more accurately. As everything is on the host there is no need of installing additional hardware or software but still redundancy is one of the important issue especially when we desire to install the system for a network and we should have a HIDS for each host. Since individual monitoring system for each host is needed, the

efficiency in terms of speed decreases and the system cost increases.

Network-based IDS (NIDS):

NIDS is a network based approach in which the system in place of collecting data from a particular host/agent directly collects it from the network monitored in form of packets. It provides better security against DoS attacks as compare to HIDS. Mostly NIDS are operating system independent and are easy to deploy. The system does not need to be installed on multiple monitoring systems hence are less expensive but lacks accuracy due to loosing of some data during the detection process. Also for large scale network scalability is still a problem. [7] [14]

Data mining (DM), also called Knowledge-Discovery and Data Mining, is the process of automatically searching large volumes of data for patterns using association rules. It is a fairly recent topic in computer science but utilizes many older computational techniques from statistics, information retrieval, machine learning and pattern recognition. Data mining techniques can be differentiated by their different model functions and representation, preference criterion, and algorithms. The main function of the model that we are interested in is classification, as normal, or malicious, or as a particular type of attack.

The use of data mining techniques in IDSs usually implies analysis of the collected data in an offline environment. There are important advantages in performing intrusion detection in an offline environment, in addition to the real-time detection tasks typically employed.

There are several reasons why data mining approaches plays a role in these three domains. First of all, for the classification of security incidents, a vast amount of data has to be analyzed containing historical data. It is difficult for human beings to find a pattern in such an enormous amount of data. Data mining, however, seems well-suited to overcome this problem and can therefore be used to discover those patterns. [8]

4. Methodology

This paper involves discussion of the two algorithms of data mining classification approaches, K-means and C4.5.

K-means algorithm:

The K-means algorithm is one of the most popular methods of clustering analysis that aims to partition 'n' data objects into 'k' clusters in which each data object belongs to the cluster with the nearest mean. It uses Euclidean metric as a similarity measure. The important properties of k-means algorithms are efficient in processing large datasets and it can work only on numerical values.

The basic algorithm of k-means is:

1. Select k objects as initial centroids
2. Assign each object to the closest centroids.

3. Recalculate the centroid of each cluster.
4. Repeat steps 2 and 3 until centroids do not change.

C4.5 algorithm:

Decision trees can be used as misuse intrusion detection as they can learn a model based on the training data and can predict the future data as one of the attack types or normal based on the learned model. It works well with large data sets. It constructs easily interpretable models, which is useful for a security officer to inspect and edit. It can also be used in the rule-based models with minimum processing.

The system flow diagram of this paper by using the K-means and C4.5 can be seen in figure 1.

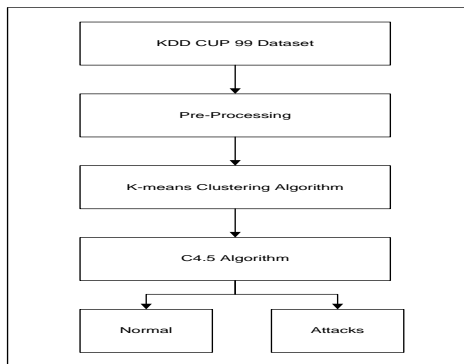


Figure 1. System Flow Diagram

5. Experiment and Result Analysis

To facilitate the experiments, we used Eclipse Java and Weka tool to implement the algorithms on a PC with 64-bit window 7 operating system, 4GB RAM and a CPU of Intel core i3-4010U CPU with 1.70GHz.

5.1. Data Selection

The experimental analysis is done by considering the typical dataset for intrusion detection named as KDD CUP'99. It is the most widely preferred dataset especially formulated for examining the newly implemented intrusion detection models. This database contains a standard set of data to be audited, which includes a wide variety of intrusions simulated in a military network environment. The data set includes 42 attributes classifying the data records into normal or a type of attack. The Table 1 notify about the 41 types of attributes in the KDD CUP'99 dataset categorized into 5 major attack classes under which they fall. [16]

Table 1: Various attacks and their respective categories

Class	Known Attacks Subclass
DoS	back, land, Neptune, pod, smurf, teardrop
Probe	ipsweep, nmap, portsweep, satan
U2R	buffer_overflow, loadmodule, perl, rootkit
R2L	ftp_write, guess_passwd, imap, multihop, phf, spy, warezclient, warezmaster

Kddcup'99 dataset have two variations of training dataset; one is a full training set having 5 million connections and the other is 10% of this training set having 494021 connections. Since the whole dataset is vast, the experiment has been performed on its smaller version that is 10% of KDD.

5.2. Result and Analysis

The analysis is performed by using K-means and C4.5 algorithms. We use K-means algorithm to generate heterogeneous dataset to nearly homogeneous dataset. Then we apply C4.5 algorithm to know the intrusions and normal traffic. For the experiment, we have run the simulation with the five different sizes of partition by using k-means algorithm and then use C4.5 algorithm for detection. And also we have run the dataset without using k-means algorithm. Then, we can compare the accuracy of two approaches. One approach is using k-means and C4.5 algorithm and the other approach is using only C4.5 algorithm. The comparison of these two approaches can be seen from table 2 to table 5.

The training time of C4.5 algorithm based on K-means is 3546.66 seconds, while that of only C4.5 algorithm is 7969.19 seconds in 10 fold cross validation. The training time of C4.5 algorithm based on K-means is 3198.51 seconds, while that of only C4.5 algorithm is 7181.49 seconds in 66-34 percentage validation. This indicates that the collaborative intrusion detection based on K-means and C4.5 is better than a single C4.5 algorithm in the training time.

Table 2. Comparison Testing Results of Two Approaches for 10 fold

dataset	k-means	C4.5	Correct instances	Correct percent	Incorrect instances	Incorrect percent
10% P1	Y	Y	108826	99.9825	19	0.0175
10% P2	Y	Y	23495	99.8597	33	0.1403
10% P3	Y	Y	280798	100	0	0
10% P4	Y	Y	78656	99.8718	101	0.1282
10% P5	Y	Y	2066	98.71	27	1.29
Total			493841		180	
10%kdd	N	Y	493823	99.9599	198	0.0401

Table 3. Comparison Testing results of Two Approaches for 10 fold with time complexity

dataset	k-means	C4.5	Total instances	Time to build model (sec)
10% P1	Y	Y	108845	907.86
10% P2	Y	Y	23528	37.79
10% P3	Y	Y	280798	2036.96
10% P4	Y	Y	78757	563.91
10% P5	Y	Y	2093	0.14
Total				3546.66
10%kdd	N	Y	494021	7969.19

Table 4. Comparison Testing Results of Two Approaches for 66-34 percentage

dataset	k-means	C4.5	Correct instances	Correct percent	Incorrect instances	Incorrect percent
10% P1	Y	Y	37000	99.9811	7	0.0189
10% P2	Y	Y	7984	99.8	16	0.2
10% P3	Y	Y	95471	100	0	0
10% P4	Y	Y	26736	99.8469	41	0.1531
10% P5	Y	Y	702	98.5955	10	1.4045
Total			167893		74	
10%kdd	N	Y	167876	99.9458	91	0.0542

Table 5. Comparison Testing results of Two Approaches for 66-34 with time complexity

dataset	k-means	C4.5	Total instances	Time to build model (sec)
10% P1	Y	Y	37007	864.42
10% P2	Y	Y	8000	35.89
10% P3	Y	Y	95471	1722.84
10% P4	Y	Y	26777	575.23
10% P5	Y	Y	712	0.13
Total				3198.51
10%kdd	N	Y	167967	7181.49

The total correctly classified instances based on K-means and C4.5 are 493841 while that of instances based on only C4.5 is 493823 in 10 fold cross validation. The total correctly classified instances based on K-means and C4.5 are 167893 while that of instances based on only C4.5 is 167876 in 66-34 percentage validation. This shows that hybrid method can correctly classify than a single method.

6. Conclusion

Experimental results show that the optimized and adaptive collaboration intrusion detection model based on K-means and C4.5 is superior to the detection system with a single C.5 algorithm in the detection accuracy and time efficiency.

Further work will be directed to experimental research of the data mining methods, approaches and algorithms by using real network data.

7. References

- [1] K.A. Al-Enezi, I.F. Al-shaikhli, A.R. Al-kandari, L. Z. All-Tayyar, "A Survey of Intrusion Detection System using Case Study Kuwait Governments Entiteis ", 3rd International Conference on Advanced Computer Science Applications and Technologies, 2014.
- [2] L. Teng, S. Teng, F. Tang, H. Zhu, W. Zhang, D. Lin and L. Liang, "A Collaborative and Adaptive Intrusion Detection Based on SVMs and Decision Trees", IEEE International Conference on Data Mining Workshop, 2014.
- [3] S. Choudhury and A. Bhowal, "Comparative Analysis of Machine Learning Algorithms along with Classifiers for Network Intrusion Detection", International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015,pp.89-95.

- [4] S.H. Vasudeo, P.P. Patil and R.V. Kumar, "IMMIX-Intrusion Detection and Prevention System", International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015, pp.96-101
- [5] P.S. Rath, M. Hohanty, S. Acharya and M. Aich, "Optimization of IDS Algorithms Using Data Mining Technique", Proceeding of 53rd IRF International Conference, Pune, India, 2016, ISBN:978-93-86083-01-2.
- [6] S. Singh and S. Jain, "A Comparative Analysis of Decision Tree Based Intrusion Detection System", International Journal of Modern Trends in Engineering and Research, Scientific Journal Impact Factor (SJIF), 2016, ISSN (Online):2349-9745.
- [7] M. Dhakar, N. Chaurasia and A. Tiwari, "Analysis of K2 based Intrusion Detection System", Current Research in Engineering, Science and Technology (CREST) Journals, 2013, ISSN 2320-706X.
- [8] R. Patel, A. Thakkar and A. Ganatra, "A Survey and Comparative Analysis of Data Mining Techniques for Network Intrusion Detection Systems", International Journal of Soft Computing and Engineering (IJSCE), March 2012, ISSN: 2231-2307, Volume-2, Issue-1.
- [9] M.P. Bhoria and Dr.K. Garg, "An Imperial learning of Data Mining Classification Algorithms in Intrusion Detection Dataset", International Journal of Scientific & Engineering Research, June-2013, Volume 4, Issue 6, ISSN 2229-5518.
- [10] R. Venkatesan, R. Ganesan and A.A.L. Selvakumar, "A Comprehensive Study in Data Mining Frameworks for Intrusion Detection", International Journal of Advanced Computer Research, December-2012, Volume-2 Number-4 Issue-7, ISSN (print): 2249-7277 ISSN (online): 2277-7970.
- [11] J. Cannady and J. Harrell, "A Comparative Analysis of Current Intrusion Detection Technologies".
- [12] L.S. Parihar and A. Tiwari, "Survey on Intrusion Detection Usingn Data Mining Methods", IJSART, January-2016, Volume-2 Issue-1 ISSN (online): 2395-1052.
- [13] Ugtakhybayar.N , Usukhybayar.B and Nyamjav.J, "An approach to detect TCP/IP based attack", IJCSNS International Journal of Computer Science and Network Security, April-2016, Vol-16 No-4.
- [14] E. Bloedorn, A.D. Christiansen, W. Hill, C. Skorupka, L.M. Talbot and J. Tivel, " Data Mining for Network Intrusion Detection: How to Get Started".
- [15] TR. Patel, A. Thakkar and A. Ganatra, "A Survey and Comparative Analysis of Data Mining Techniques for Network Intrusion Detection Systems", International Journal of Soft Computing and Engineering (IUSCE), March-2012, Vol-2, Issue-1, ISSN: 2231-2307.
- [16] M. Dhakar and A. Tiwari, "A New Model for Intrusion Detection based on Reduced Error Pruning Technique", I.J. Computer Network and Information Security, 2013,22,51-57.
- [17] A.A. Nasr, M.M. Ezz and M.Z. Abdulmageed, "Use of Decision Trees and Attributional Rules in Incremental Learning of an Intrusion Detection Moedl", International Journal of Computer Networks and Communications Security, July-2014, Vol-2, No-7, 216-224, ISSN 2308-9830.

Delay Controlled Elephant Flow Rerouting in Software Defined Network

Hnin Thiri Zaw, Aung Htein Maw

University of Computer Studies, Yangon, University of Information Technology
h.thirizawucsy@ucsy.edu.mm, ahmaw@uit.edu.mm

Abstract

As the network has limited resources, the traditional network with a single path routing mechanism will make the inefficient network resource utilization. Because the competitive utilization along the single path causes the performance degradation of traffic flows. Multipath routing is a good approach for inefficient resource utilization problem. It distributes the traffic load among parallel paths instead of single path. Moreover, the long flows which called elephant flows are needed to detect and handle in order to avoid traffic congestion. This paper proposes an effective solution by combining the elephant flow detection and rerouting based on end-to-end delays of available paths in Software Defined Network (SDN). The proposed method is implemented by using ONOS controller and Mininet emulator. The experimental results prove that 16.57%~78.03% throughput improvement and 26.18%~171.68% flow completion time (FCT) reduction for multiple elephant flows compared with the single path routing approach.

Keywords- Multipath, Elephant flow, Software defined network, SDN

1. Introduction

As the growth of deployment in online end-user applications such as VoIP, online gaming, video conferencing and so on, maintaining high throughput and low latency issues is important challenges for networks. To be efficient network resource utilization and fulfilled with quality of service (QoS) requirements, more and more traffic engineering (TE) applications are needed to innovate for better route decisions by measuring and controlling traffic flows. Reactive forwarding application of ONOS controller is a default single path application to make forwarding decisions whenever a new flow arrives at the switch [2]. The main process is that the switch sends a copy of the first packet header from new flow to the controller and then the controller installs forwarding rule to the switch. The major drawback of the reactive forwarding method is that it makes route decisions without awareness traffic condition and QoS parameters (such as bandwidth, delay, packet loss and jitter), resulting in throughput degradation. The aim of the proposed method is to solve the drawback of the reactive forwarding method. This

paper presents the overview of the proposed architecture to generate better route decision by considering traffic conditions (measuring elephant flows) and end-to-end delays of paths. The proposed method includes three main folds: (1) monitoring and detecting the elephant flow periodically (2) measuring end-to-end delays of available paths between source and destination where large flow happens and (3) rerouting the elephant flow to the least delay path. The proposed method implementation uses ONOS [10] controller and Mininet [8] emulated. sFlow [7] analyzer is used to monitor elephant flow by using packet sampling technology. The single path routing is used for mice flows by default. Once the elephant flow is detected from sFlow analyzer, end-to-end delays of all paths between source and destination nodes are measured and elephant flow is shifted to the least delay path.

The remainder of this paper is organized as follows. Section 2 presents related work overview. Section 3 gives the explanation about the overall architecture of the proposed method with three main tasks: large flow detection, end-to-end delay estimation and rerouting elephant flow. In Section 4, performance evaluation describes experiment scenario with throughput and packet loss. Section 5 presents the conclusion of this paper.

2. Related Work

The existing traffic rerouting models implement different strategies in the multipath forwarding mechanism. The authors in [3] propose the routing algorithm splits the elephant traffic into mice and distributes them across multiple paths based on source routing (label based forwarding) with round-robin manner. The limitation of their method is that it requires overhead bytes to implement policy in packet header increases linearly with path length. The difference is that their approach uses round-robin to split traffic load and our method is based on estimated delays of each path. Hedera [4] is a flow scheduling scheme to solve the hash collision problem of Equal Cost Multipathing (ECMP). It reduces large flow completion time (FCT) caused by network congestion and utilizes the path diversity of data center network topologies. The difference is that Hedera uses per flow statistics for large flow detection, which has poor scalability and our

method uses packet sampling. DiffFlow [5] differentiate short flow and long flow by using a packet sampling method. It applies ECMP to short flows and Random Packet Spraying (RPS) method to long flows. Their method causes packet reordering problem while transferring each packet to random egress ports because of different packet delivery time of available paths between source and destination. Our proposed method can avoid reordering problem since it is flow-based rerouting. Another work of traffic rerouting in [6] monitors congested path by collecting port statistics of each switch by using OpenFlow protocol. When congestion occurs, it computes the least loaded path and reroutes some traffic flows from the congested path. TinyFlow [9] presents large flow detection and random rerouting method. Once an elephant is identified, the edge switch adds a new rule to the flow table and collects byte count statistics periodically. When the byte count exceeds a limit, the switch picks an alternate egress port out of the equivalent cost paths randomly for elephant, reinstalls the new flow entry, and resets the byte count. The drawback of TinyFlow is the elephant flow collision problem at the random egress ports at aggregate switches, resulting in poor bandwidth utilization.

In this paper, the proposed rerouting method is mainly based on large flow identification and end-to-end delay estimation. As soon as large flow is detected, the controller computes delays of parallel multiple paths between source and destination and reroutes the large flow to the path with the least delay path in order to improve throughput.

3. Overall architecture of proposed method

The overall architecture of the proposed method (see in Figure 1) is to reroute elephant flow based on average end-to-end delays of parallel paths between source and destination. The sFlow real time analyzer is used for monitoring and detecting elephant flows. In order to access the elephant flow information from our proposed method, the sFlow REST API is called in every 1 second. The new elephant flow event can be defined in the proposed rerouting method by comparing the timestamp values of elephant flow events since sFlow REST API provides flow information with time stamp values. According to the flow chart of Figur 1, as soon as the elephant flow is found, firstly it finds an available shortest path list in terms of hop counts between source and destination nodes. Then end-to-end delay of each path from path list is measured by sending out probe packets from the controller. From delay measurement module, the

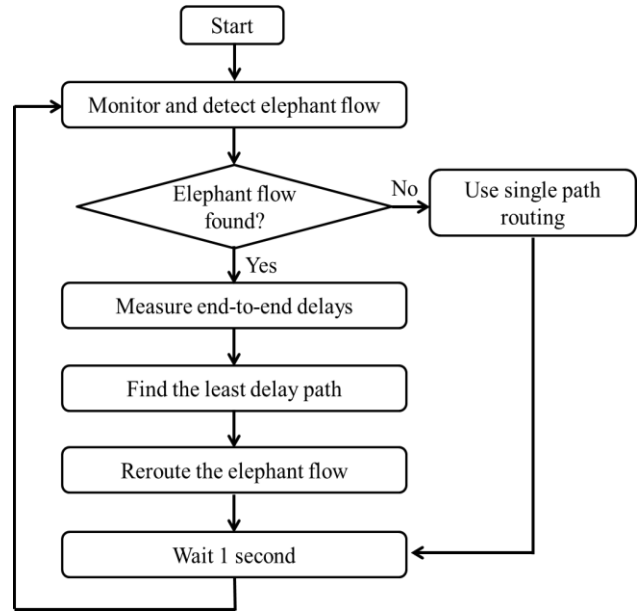


Figure 1. Flow chart of proposed method

average delays of each path can be calculated. After comparing the average end-to-end delays of available paths, the elephant flow is shifted to the least delay path to optimize throughput performance. For TCP traffic flow, the least and second least delay path are selected. In general, three main modules: monitoring and detecting elephant flows, measuring end-to-end delays and flow rerouting are developed for ONOS application.

3.1. Monitoring and detecting elephant flows

The proposed method uses sFlow analyzer for elephant flow monitoring and detection. This analyzer is a real time traffic analyzer for software-defined networking. It makes network traffic visibility in both physical and virtual devices (eg. Open vSwitch). sFlow uses packet sampling technology to analyze traffic statistics and it is based on the collector and agent architecture (see in Figure 2). The analyzer (or) collector receives a continuous stream of sFlow datagrams periodically from its agents which are embedded in network devices such as routers and switches. Therefore, sFlow solutions consist of two components (1) network equipments equipped with sFlow agents which monitor network traffic and generate sFlow data, and (2) sFlow application that receives and analyzes the sFlow data. Then the collector analyzes the utilization statistics of every traffic flow on all ports of devices. sFlow agents do very little processing. They simply package data into sFlow datagrams that are immediately sent to the sFlow collector. Once the utilization of traffic flow exceeds the specified threshold value, the collector

converts them into metrics which are specified in keys of

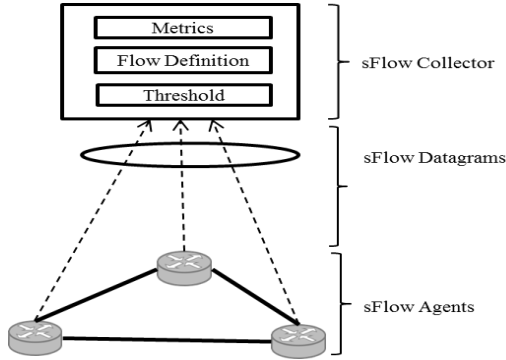


Figure 2. The Flow architecture

```
SetFlow('tcpflow',
  {keys: 'macsource, macdestination,
        ipsource, ipdestination,
        tcpsourceport, tcpdestinationport,
        link:inputifindex, link:outputifindex',
   value: 'bytes'});
```

Figure 3. Flow definition of sFlow collector

flow definition. The output metrics are represented by JSON format which consisting of attribute-value pairs. According to the flow definition in Figure 3, the output information of elephant flow includes source and destination MAC addresses, IP addresses, TCP port numbers and names associated with the ports of a link. The elephant flow events of sFlow collector are queried by the proposed rerouting method via calling REST API: /events/json which is used to filter the threshold exceed events. Here the REST API calling interval is set from delay based rerouting application to sFlow analyzer is 1 second (less than 1 second affects the accuracy of delay estimation).

3.2. Measuring end-to-end delay

In delay measurement, three probe packets are needed to send for one path. Probe packet includes two parts (see in Figure 4): header and payload. The header field includes faked source/ destination MAC addresses and Ethernet type value (0x5577). The payload field includes a time stamp (sent time) value instead of traditional packet encapsulation. According to Figure 5, let's assume to find end-to-end delay between source S1 and destination S2. Firstly, the flow entries are needed to install to each device along the path proactively before sending the first probe.

The matching fields of flow entries are source/destination MAC addresses and Ethernet Type. The action output ports for flow entries are based on

links of the path. For example, the action output port for S1 is 2 (source port of link) and the output port for S2 is default controller port c0 because S2 is the last device and there is no next link in the path. Table 1 and Table 2 show the flow entries for S1 and S2. Here, faked source MAC and destination MAC are assumed as 11:11:11:11:11:11 and 22:22:22:22:22:22 respectively.

6 bytes	6 bytes	2 bytes	8 bytes
Src Mac address	Dst MAC address	Type	Payload

Figure 4. Frame format of probe

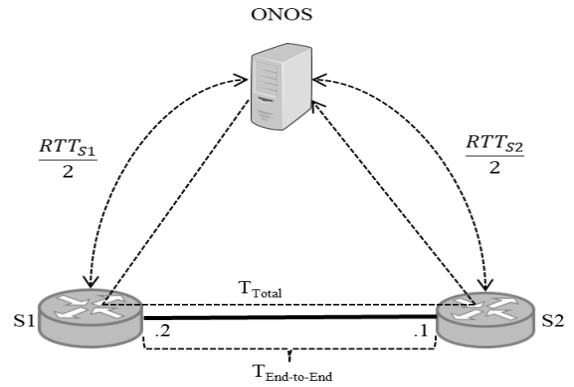


Figure 5. Delay measurement architecture

Table 1. Flow entry for S1

Source MAC	Destination MAC	EtherType	Action
11:11:11:11:11:11	22:22:22:22:22:22	0x5577	Port 2

Table 2. Flow entry for S2

Source MAC	Destination MAC	EtherType	Action
11:11:11:11:11:11	22:22:22:22:22:22	0x5577	C0

After flow entries installation, the first probe is sent through source switch to the destination switch along the path and back to the controller. When the first probe is received back, the controller records the packet arrival time $T_{arrival}$. Then the header information and payload are extracted to get packet sent time T_{sent} . From the first probe, the total delay time T_{total} (including $T_{arrival}$ and T_{sent}) can be learned. After the first probe, the next two probe packets are also generated from the controller to source switch S1 and destination switch S2 respectively like the first probe. From these two probes, the two round-trip-time between the controller and switches (RTT_{S1} and RTT_{S2}) can be found. It can be summarized as follow:

- 1st probe packet: measure T_{total} ($T_{arrival} - T_{sent}$),

- 2nd probe packet: measure RTT_{S1} , and
 - 3rd probe packet: measure RTT_{S2} .
- Therefore, the equation for end-to-end delay $T_{end-to-end}$ cost can be derived following:

$$T_{end-to-end} = T_{total} - \frac{RTT_{S1}}{2} - \frac{RTT_{S2}}{2} \quad (1)$$

Since the delay estimation method is based on end-to-end delay, the half round-trip-time is assumed as the one-way delay in the calculation.

3.3. Rerouting flows

After delay estimation of available paths between source and destination where large flow occurs, the least delay path is selected and new flow entries are injected to respective devices through this path by using FlowRuleService which is provided from ONOS controller. For TCP traffic flow, the least delay path and second least delay path are chosen to optimize TCP throughput. The traffic selection fields of each flow entry address, destination MAC address, and TCP ports. When the traffic flow does not exceed the threshold, the route decision and flow entries are made by using the single path routing method. After utilization exceeds, the route decision and new flow entries are made by delay based elephant flow management. The old entries which are injected from single path mechanism will be removed automatically after 10 seconds, which is identified in idle-timeout. The idle-timeout is A flow table entry is removed if no packet matches the rule within a certain amount of time.

4. Performance evaluation

We evaluate the proposed delay aware rerouting method using emulated testbed as shown in Figure 6. Two laptop PCs are used for evaluating the performance results. The first PC (i.e., Core i5-5200U CPU @ 2.20GHZ with RAM 4GB, Ubuntu 14.04 on Oracle VM VirtualBox) serves as ONOS controller. The second Laptop PC (i.e., Core i5-5200U CPU @ 2.20GHZ with RAM 4GB, Ubuntu 14.04) serves as mininet emulator and sFlow-rt collector.

4.1. Testbed Emulation

The network topology as shown in Figure 6 is created by using Mininet emulator (version 2.2.1) which can create the virtual network and provide hundreds and even thousands of virtual hosts. The topology is inspired by leaf-sine topology which is one of modern data center architectures. In leaf-spine topology, all leaf switches form access layer and meshed to range of spine switches. ONOS controller (version 1.8) is used among other kinds

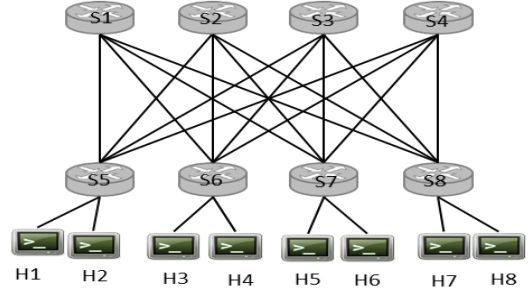


Figure 6. Emulated leaf-spine topology

of SDN controllers because of its performance, high-level abstractions and API. ONOS is distributed system which is designed for scalability and high availability. Iperf [12] tool is also used to generate TCP traffic and evaluate throughput and flow completion time (FCT). The FCT of a flow is the time difference between the time when the first packet of a flow leaves the source and the time when the last packet of the same flow arrives at the destination [5].

4.2. Parameter settings and evaluation results

Experimental scenarios are based on the two different parameter settings for the testbed topology (see in Figure 6). The proposed method is evaluated by generating four different numbers of TCP elephant flows to stress the network as shown in Table 5. In the first settings, the up-link speed is 10 Mbps and the down-link speed is 60 Mbps. The window size (or) socket buffer size at the receiver is 65535 bytes and sender is used default window size. The threshold value of elephant flow is greater than (or) equal 1 Mbps and packet sampling rate is 1 in 10 packets. In the second settings, the up-link speed is 2 Mbps and the down-link speed is 12 Mbps. The window size (or) socket buffer size at the receiver is the same as first parameter setting. The threshold value of elephant flow is greater than (or) equal 0.2 Mbps and packet sampling rate is 1 in 2 packets. In both parameter settings, the amount of data transfer for Iperf testing is 150 MB. There are four paths (P1, P2, P3, P4) between every source and destination in Figure 5. Different delays are used to test in both settings. Table 3 and Table 4 show the summarized parameter settings in details.

Table 3. Parameter setting I

Parameter	Value
Link speed	Up:10 Mbps, Down:60 Mbps
Threshold	1 Mbps
Sampling rate	1 in 10
Window size	65535 Bytes
Latency	P1, P2, P3, P4 : [20, 50, 80, 110] ms

Table 4. Parameter setting II

Parameter	Value
Link speed	Up:2 Mbps, Down:12 Mbps
Threshold	0.2 Mbps
Sampling rate	1 in 2
Window size	65535 Bytes
Latency	P1,P2,P3,P4:[2.78,20.2,24.6,6.8] ms

Table 5. Multiple elephant flow information

Number of flows	Source Host→Destination Host
1	H8→H1
2	H3→H1, H4→H2
4	H3→H1, H4→H2, H5→H1, H6→H2
6	H3→H1, H4→H2, H5→H1, H6→H2, H7→H5, H8→H6
8	H5→H1, H5→H3, H6→H2, H6→H4, H7→H1, H7→H3, H8→H3, H8→H4
10	H3→H1, H4→H2, H5→H1, H5→H3, H6→H2, H6→H4, H7→H1, H7→H3, H8→H3, H8→H4
12	H1→H3, H2→H4, H3→H1, H4→H2, H5→H1, H5→H3, H6→H2, H6→H4, H7→H1, H7→H3, H8→H3, H8→H4

The results of the proposed method are compared with the single path method. In Figure 7, the delay based rerouting method has the throughput improvement 16.57%~78.03% . This is because the proposed method reroutes the elephant flows to the least delay path while the single path method only uses the shortest paths for all traffic flows. In Figure 9, although the throughput improvement is 49.84%~79.03% for 2 and above 6 TCP elephant flows, the proposed method has the same result with the single path method for 1 and 4 TCP flows. The same results occur when the single path routing chooses the least delay path. In Figure 8 and 10, the results show that 26.18%~171.68% FCT reduction of proposed method. Therefore, it has been studied that the more elephant traffic flows in the network, the proposed scheme still outperforms evidently. According to throughput improvement, the proposed method is more outperformed when the link speed is 2 Mbps. The proposed rerouting scheme can reduce the performance degradation problem (in terms of throughput) of single path routing, i.e. poor bandwidth utilization without awareness of path condition and traffic types. The delay based traffic rerouting method is presented in software-defined network by emulating layer 2 topology. The proposed method leverages an SDN infrastructure to support delay estimation and traffic rerouting. Unlike the traditional single path routing method, the proposed

method includes: differentiation elephant flows, estimation end-to-end delay of available

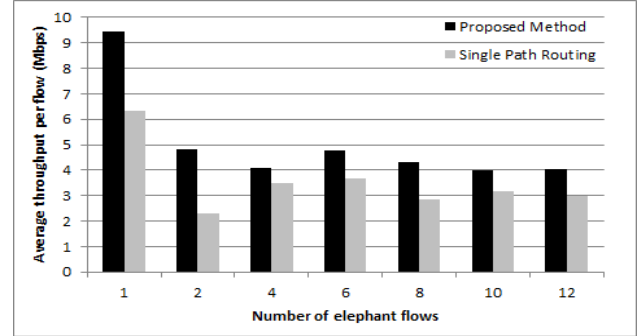


Figure 7. Throughput results for parameter setting I

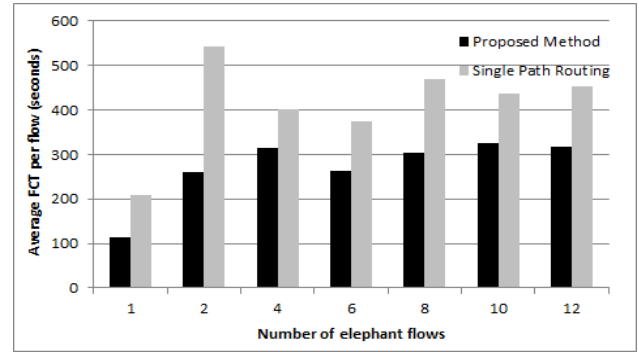


Figure 8. FCT results for parameter setting I

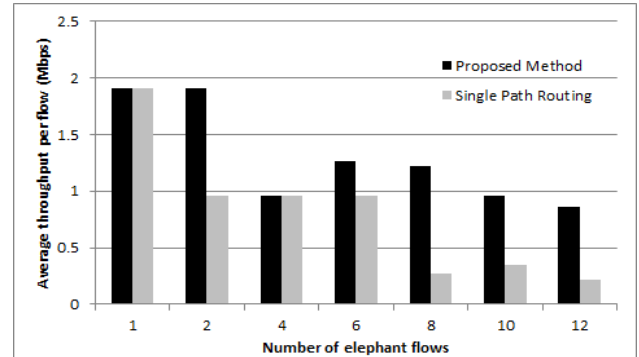


Figure 9. Throughput results for parameter setting II

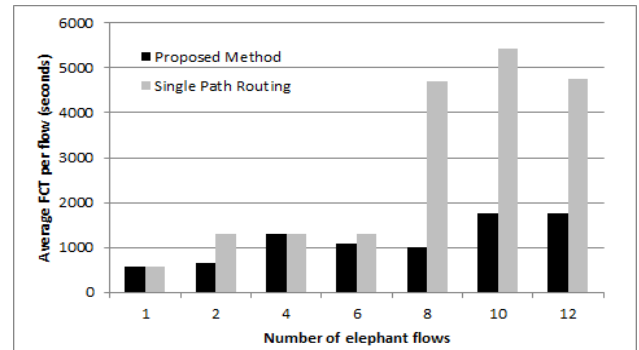


Figure 10. Throughput results for parameter setting II

paths between specified source and destination and reroute the elephant flows to the least delay path. The objective of proposed method is to improve network performance by measuring and managing traffic dynamically. The experimental results show throughput and FCT between the single path routing and delay based rerouting method as shown in Figure 7, 8, 9 and 10. The delay based rerouting scheme effectively uses least delay paths with the available bandwidth to avoid the congestion link. Hence, it has been obtained the better throughput and FCT after using the path delay based rerouting method.

5. Conclusion

The proposed delay based elephant flow rerouting method is implemented by using OpenFlow version 1.0 and it works on layer 2. Consideration for available bandwidth utilization is beyond the scope of paper and the future work will be considered it. The difference from traditional single path routing method is that the proposed method differentiates types of flows and reroute the elephant flow to least delay path in order to optimize throughput. According to experimental results, the proposed method improves the throughput results 16.57%~78.03% and 26.18%~171.68% FCT reduction as compared with the single path routing approach.

6. References

[1] O. M. E. Committee, "Software-defined networking: The new norm for networks", ONF White Paper, 2012, pp. 2--6.

[2] A. Bianco, P. Giaccone, , R. Mashayekhi, M. Ullio, V. Vercellone, "Scalability of ONOS reactive forwarding applications in ISP networks", *Computer Communications*, 2017, pp. 130--138.

[3] S. Hegde, S. G. Koolagudi, S. Bhattacharya, "Scalable and fair forwarding of elephant and mice traffic in software defined networks", *Computer Network*, 2015, pp. 330--340.

[4] M. Al-Fares, S. Radhakrishnan , B. Raghavan, N. Huang, A. Vahdat, "Hedera: Dynamic Flow Scheduling for Data Center Networks", In *NSDI*, 2010, pp. 19--19.

[5] F. Carpi, A. Engelmann, A. Jukan, "DiffFlow: Differentiating Short and Long Flows for Load Balancing in Data Center Networks", In *Global Communications Conference (GLOBECOM)*, 2016, pp. 1--6.

[6] M. Gholami, B. Akbari, "Congestion control in software defined data center networks through flow rerouting", In *Electrical Engineering (ICEE), 23rd Iranian Conference on*, 2015, pp. 654--657.

[7] *Peter Phaal, March 2013 [Online]. Available from: <http://blog.sflow.com/2013/03/ecmp-load-balancing.html>*

[8] B. Lantz, B. Heller, N. McKeown, " A network in a laptop: rapid prototyping for software-defined networks", in *SIGCOMM*, 2010, pp. 19 .

Bandwidth Allocation Scheme using Segment Routing on Software-Defined Network

Ohmmar Min Mon, Myat Thida Mon

University of Information Technology, Yangon, Myanmar

ommm@uit.edu.mm,myattmon@uit.edu.mm

Abstract

Nowadays, Internet is the main communication medium but traditional network approach has been difficult to adapt to the fast growing Internet. The emergence of Software Defined Network (SDN) has brought to predict for the solution of this problem. Methods for measuring Quality of Service (QoS) parameters such as bandwidth utilization, bandwidth allocation and delay have been introduced not only for traditional network but also for SDN-based scenarios. Bandwidth allocation is a great significance in various areas of networking. We propose a low-complexity bandwidth allocation algorithm based segment routing for real-time and non-real-time service requests to allocate network resource to ensure the request of services. In this paper, we present available bandwidth allocation scheme based on SDN to provide Quality of Service (QoS) support for various services via experiments under our SDN testbed.

Key Words-Quality of Service, Bandwidth Allocation, Software Defined Networking, Segment Routing

1. Introduction

With the development of the Internet, there are more and more types of services carried by the Internet competing for network resource, most of which require performance guarantees. Nowadays IP networks are very complex to build and manage. Software Defined Networking (SDN) offers a solution for this problem mainly through the following features:

- (1) data and control planes are decoupled;
- (2) control logic is moved out of SDN switches to an external the SDN controller and
- (3) external applications can program the network using the abstraction mechanisms provided by the SDN controller. Controller communicates networking devices to collect information from them and also to push configuration information to them.

Segment Routing (SR) is a new emerging routing technique and can be used in MPLS network that enables source routing. In the segment routing domain, nodes and links are assigned Segment Identifiers (SIDs). SR uses a sequence of segments to compose the desired end-to-end connection path. The segment labels are

carried in the packet header and so per-flow state is maintained only at the ingress node. SR allows better control of the routing paths and can be used to distribute traffic for better network utilization. A central controller can take advantage of the possible segment routing by choosing segments based on the traffic. The major advantage of SR is to eliminate the per-flow states from the service provider's core routers. In fact, a path is directly usable by any router; no prior setup/signalization is required, unlike MPLS-TE where a tunnel has to be signaled and maintained using protocols such as the Resource Reservation Protocol Traffic Engineering (RSVP-TE) or Label Distribution Protocol (LDP). It allows the network operator to be forwarded a path from ingress node to egress node. When MPLS is used to instantiate SR tunnels, the MPLS forwarding plane does not change. There are many advantages when we integrate SR in the data plane and in SDN-based control layer technologies. As for scalability and agility, SR avoids the requirement for many labels to be stored in each network device along the path.

In the most recent years, there have been several proposals for monitoring Quality of Service (QoS) parameters in SDN networks. On the other hand, the paper in literature that deals with the problem of bandwidth allocation using SR in SDN. Service providers use this parameter for network management and traffic engineering purposes. In this case, users can send bandwidth allocation requests to the controller if necessary.

The controller receives the requests using the algorithm to calculate routes that can satisfy the demands of those users and allocate the bandwidth for them. This paper presents a solution for bandwidth allocation by monitoring traffic statistics from network devices in an SDN environment, we want to allocate bandwidth on each link in the network and find the highest available bandwidth between two points in the network. We also validate our method on a test environment using the Mininet network emulation tool and the ONOS SDN controller.

Internet Service Providers demand for enhanced network performance, increase the need for network resources. The main challenge is how to achieve optimal path for real time traffic. The second challenge is to

allocate bandwidth on each link for large networks. The major contribution of this system is to present Bandwidth Allocation Algorithm for the traffic that requests for the path with available bandwidth.

The rest of the paper is structured as follows:

Section 2 briefly reviews the related work. Section 3 explains segment routing operations. Bandwidth Allocation Algorithm is demonstrated in Section 4. Section 5 conducts the performance of our mechanism with experimental results on SDN testbed. Finally, section 6 concludes the paper and points out the future research directions.

2. Related Work

The SDN controller can communicate with the switch via the southbound API, where the most used standard is OpenFlow. For NOS platform there is much available open software such as NOX, POX, Floodlight or Ryu. Moreover, there are ongoing industrial projects for controller platforms specialized for data centers, for e.g. OpenDayLight or ONOS [1].

Slavica Tomovic, et.al[2] analyzed performance of bandwidth constrained and bandwidth-delay constrained centralized routing algorithms in large-scale SDN backbone networks. Bandwidth rejection ratio (BRR) and the average route length are used as the two main performance metrics to evaluate the algorithms. The best results of BRR performance were obtained when BWP (Bandwidth Propotion) parameter was set to 0.9.

Hyunhun Cho, et.al[3] constructed an application that calculates optimal paths for data transmission and real-time audio and video transmission. This paper showed that the average of the optimal path is about 100Mbps faster than that of other path.

In [4], the authors discussed the performance of the network by separating application into bandwidth-oriented and latency-oriented application. This system discussed the use case for the bandwidth and latency aware network.

Trong-Tien Nguyen and Dong-Seong Kim [5] present a novel routing scheme called Accumulative-Load Aware Routing for Software-Defined Networks. ALAR algorithm is implemented routing decisions based on Dijkstra algorithm to calculate the link cost values.

In [6], P. Megyesi, A. Botta, G. Aceto, A. Pescapè, S. Molnár presented an approach to measure end-to-end available bandwidth (ABW) in Software Defined Networks (SDN). This system reported the results obtained in different network configurations.

Péter Megyesi, et.al[7] proposed the use of a passive technique for the Available Bandwidth estimation, taking advantage of the NOS in the architecture of SDN. This system analyzed the source of errors introduced by SDN and OpenFlow and considered the possible

implementations of ABW estimation based on meter statistics.

Cihat Cetinkaya, Erdem Karayer, MugeSayit, Cornelius Hellge [8] proposed a model to find the most suitable path for DASH services over SDN. It discussed how to add the different types of clients such as standard, HDTV and mobile users and improved the optimization model by considering the client properties.

In [9] Diyar Jamal Hamad, et.al proposed how to get traffic measurement statistics from network devices in an SDN environment. This system shows that the rate received from ports is a little bit higher than the generated traffic because there is background traffic in the network.

In [10], Rakesh Kumar, et.al proposed mechanisms that provide end-to-end delays for critical traffic in real-time systems using COTS SDN switches. This system shows that increasing the number of flows slightly decreases end-to-end delays.

This paper presents to allocate bandwidth on each link in SDN. The results show that bandwidth allocation algorithm in the SDN can optimize the routing paths.

3. Segment Routing Operation

In order to make the current IP and MPLS network more service-oriented and efficient, in 2013, IETF proposed Segmentation Routing (SR) technology[11]. It enables to use non-shortest paths by specifying alternative routes. It is typically associated with a centralized control plane implementation.

SR controller can support the global information of network resources and the global deployment and optimization of resources according to the service requirements of the source nodes, such as traffic engineering, load balancing, etc. The controller is in charge of setting up the edge-to-edge services, by configuring the ingress and egress PE nodes for a given flow.

In segment routing, nodes and links are assigned Segment Identifiers called segments. In the case of a link (i.e. adjacency) segment, the shortest path to the upstream node is taken and then that link is crossed.

There are three actions that are performed on segments by SR-capable nodes [1]. They are associated with operations performed on MPLS labels in MPLS networks. Segment Routing operations are: (a). PUSH (MPLS PUSH) – a segment is pushed on the top of segment stack (b) NEXT (MPLS POP) – an active segment is completed and it is removed from the stack (c). CONTINUE (MPLS SWAP) – active segment is not completed yet and it remains active.

Table 1 shows mapping of each SR control plane operation with a corresponding MPLS operation. The

basic functionality of each corresponding operation in both MPLS and SR are the same.

Table 1. SR operations mapping to MPLS operations

Segment Routing	MPLS
SR Header	Label Stack
Active Segment	Topmost Label
PUSH Operation	Label Push
NEXT Operation	Label POP
CONTINUE	Label Swap

In this paper, the switches update their forwarding tables according to the commands taken from the controller. In the forwarding tables of each switch, the output ports are kept according to the source address, destination address and input port. The switch informs the controller about the requested flow. The controller selects a path considering the requested bandwidth. After selecting an optimal path for the requested flow, the controller sends flow information to the corresponding switches along the selected path via FlowMod message which is defined in OpenFlow protocol. In order to determine the path, the controller needs to calculate the available bandwidth of the paths. For this purpose, the controller queries the switches periodically via sending OFPC_PORT_STATS messages which is defined in OpenFlow protocol to obtain information about available bandwidth on the links.

Figure 1 illustrates a simple network topology to describe for segment routing operation. When a traffic flow has to be routed along the shortest path to its destination, a segment list including only one label can be used (i.e., the SID of the destination node). In Figure 1, if the target path for an incoming traffic flow is $p = \{R1, R2, R3, R4\}$ the segment list is $SL = \{R4\}$. In this section, this system presents an example of how segment routing works.

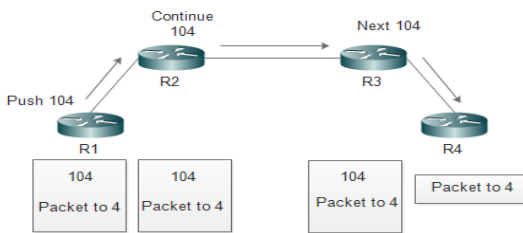


Figure 1. Segment Routing Operations

4. Bandwidth Allocation in SDN

Software Defined Networking offers the chance to speed-up the adoption of QoS control in the Internet.

When a user needs certain bandwidth, it sends a bandwidth allocation request packet to the controller. Request packet contains information such as the identification of the user, the destination, how much bandwidth it needs, when it needs the service. When switch received a packet that has no matched flow-entry, it encapsulates the packet and sends the request packet to the controller as a Packet-in message. When the controller received a packet-in message, it checks whether it is a bandwidth allocation request or not. If it is, the controller uses the topology and bandwidth information to calculate the route for bandwidth allocation. Finally the controller responds the bandwidth allocation result to the switch.

4.1. Bandwidth Allocation in Controller and switch

End-to-end bandwidth allocation can be achieved by setting the flow to transmit packets from a host to another with QoS. Firstly the controller tries to route the higher bandwidth flow via a path that can accommodate the required rate. If such a path is not presented, the flow will be dropped. Best-effort flows skip the admission procedure and will always use the shortest path between the end hosts. The controller keeps track of how much bandwidth is allocated in the network. Using this information, the controller decides where to route the path.

Firstly, our application uses this information to build up the network topology graph $G(V, E)$ as shown in Figure 2, where the node set V corresponds to the switches and the edge set E corresponds to the links (for further notations see Table 2). Link states of edge E are indicated by vector: (C, u) where 'C' is the link capacity and 'u' is the link bandwidth utilization. The utilization of a link can be calculated by Equation. (1). In this case, bw_u is the bandwidth usage which can be obtained by monitoring the traffic flow of the link.

$$u_E = \frac{bw_u}{C_E} \quad (1)$$

The available bandwidth of each path can be computed by Equation. (2):

$$bw_p = \min_{1 \leq i \leq n} (C_E - bw_u) \quad (2)$$

Our mechanism attempts to avoid traffic congestion when new flow is added to the link. Here, we have to get the path P between source and destination in the network where the available bandwidth is the largest. This can be calculated through the following Equation. (3):

$$abw_{S \rightarrow D} = \max_{P \in P_{S \rightarrow D}} \min_{1 \leq i \leq n} (C_E - bw_u) \quad (3)$$

To get the path P with largest available bandwidth between source and destination, we use Modified Dijkstra algorithm. In this algorithm, the cost of the path, denoted C_P , is measured by the minimum bandwidth instead of the sum of edge of capacities used in traditional Dijkstra algorithm according to Equation (4).

$$C_P = \sum_{i=1}^{n-1} C(bw_u, C_E) \quad (4)$$

In this algorithm, a path from source to destination in G with the maximum bandwidth constructed. Here the array element P records the father of the node v in the maximum bandwidth tree and the array element P records the bandwidth of the path from source node to another node in the maximum bandwidth tree. This algorithm takes $O(m+n \log n)$ times to run where m and n are links and nodes respectively in the network G.

Table 2. Notation list

Notation	Description
$G(V, E)$	The directed graph representation of the topology with node V and edge E
u_E	Utilization of link in the topology graph
C_E	The capacity of link
bw_u	The link usage bandwidth
bw_P	The available bandwidth of each path
$P_{S \rightarrow D}$	The set of all available paths from S to D

4.2. Bandwidth Allocation Algorithm

The general idea to measure the bandwidth of the path in the network is to provide end to end communication between switches and hosts. This section mentions Bandwidth Allocation Algorithm for the traffic that request for the path with available bandwidth. To explain Bandwidth Allocation Algorithm, consider a network with n nodes. To find the available bandwidth, the controller needs to know the current network topology and links reserved bandwidth.

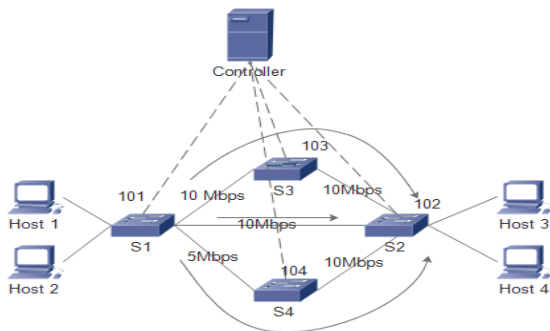


Figure 2. Network topology used in the experiment

This system uses the network topology that describes how bandwidth allocation can be implemented with SR. In the topology, source node is S1 and destination node is S2. When the flows entered the network, for the case of higher bandwidth application traffic S1-S3-S2, S1 would push a SR header with segment list {101,103,102}, and forward it to S2 as shown in Figure 2.

Best-effort traffic should be steered over a shortest path, which is S1 to S2. S1 would place the segment list {101,102} in the SR header, and forward to S2.

The algorithm in Figure 3 shows the Bandwidth Allocation Algorithm employed to compute all-pair maximum bandwidth paths. We propose this Algorithm, using the following notation:

- fbw= Feasible bandwidth
- abw= available bandwidth
- Input: Feasible bandwidth
- Output: best available bandwidth

```

Initialize# Find and Use Available Bandwidth
Input: fbw;
Initialize: abw=0;
If fbw ≥ widest bw
    then abw = fbw
    Else abw = widest bw
Endif
For all n ∈ N do
    If bwn < abw then bwn = 0
Endfor

```

Figure 3. Pseudo code for available bandwidth

4.3. Segment Routing Emulation

We emulated the topology of the network using Mininet, a popular network emulator among Software Defined Network researchers. In Figure 4, we used the pseudo code to emulate the topology. Our implementations of the controller correctly assigned paths for each application. This section includes the setup and verification of a custom SR test application, utilizing Mininet and the ONOS controller. The SR tests contain a SR path selection process according to specific simulated application demands.

```

//Create LeafSpine Topology//
Input: Switch, Host;
Initialize: S=0, H=0, N=4;
For all i=1 to N do
    Si = addSwitch (Si);
    Hi = addSwitch (Hi, IPi);
    Li = createLink (Si);
Topo T = S, H, L;
Return T;

```

Figure 4. Pseudo Code for Creating SR topology

To create the traffic flows, ICMP requests and replies will be used. The system creates a network topology for segment routing tests. And the topologies are created in the ONOS virtual environment. The experiment is employed with the topology to evaluate bandwidth aware routing.

5. Experiment Environment and Results

This section describes the environment under which all the experiments are done. All the experiments are done on a virtual machine (VM) with the configuration using Leaf-Spine architecture. This Leaf-Spine architecture is designed to provide very scalable throughput in a uniform

and predictable manner across thousands to hundreds of thousands of ports. Every leaf switch connects to every spine switch in the fabric.

Our evaluation was performed on the topology depicted in Figure 2 that we created within the Mininet framework used for emulating virtual SDN networks. Mininet is a network simulator for simulating SDN scenario in the Linux environment. The network topology is comprised of 4 virtual switches interconnected with an SDN controller, 4 virtual hosts and virtual Ethernet links interconnecting the switches. In our simulation, we write a script based on Python to generate flows that we need. The SDN controller is an Open Network Operating System (ONOS) controller configured to compute the available bandwidth.

The ONOS SPRING-OPEN project VM image has a 64-bit Ubuntu 16.04 installed as the guest OS and the VM is preconfigured to run with two virtual CPUs and two GB of RAM. These are the minimum requirements to run the environment. The PC has Microsoft Windows 8.1 OS and Oracle's VirtualBox hypervisor installed. The system was run with Core(TM)1.6 GHz CPU and 4 GB of RAM.

The ping command can not only provide connectivity results in the network but can also be used for latency measures. Ping provides the capability to determine the size of the packet that you are sending on the network.

The ONOS controller at the network topology is used to load a configuration for the real time traffic in order to emulate routing capabilities on them. Each switch is assigned with a loopback IP address and Segment Routing nodes are identified with Segment Routing ID (SID).

5.1. Evaluation Results

This section shows results of the tests that we run to evaluate Segment Routing network performance. In this system, the controller includes a routing policy based on a maximum bandwidth threshold 10Mbps between the switches. If the used bandwidth is below the acceptable threshold, the controller provides a routing preference for best-effort flow, otherwise for higher bandwidth flow.

On the ONOS controller, it is time to setup the virtualized network in Mininet. The resulting output after command execution is shown in Figure 5.

```

root@SR-onos:~# ./sw4host4topo.py
Connecting to remote controller at 127.0.0.1:6653
*** Creating network
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s1) (h3, s2) (h4, s2) (s1, s3) (s1, s4) (s2, s3) (s2, s4)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4
*** Starting CLI:

```

Figure 5. The virtualized network with SR

For traffic engineering with SR, the standard best path selection behavior can be seen by conducting series of ping requests from across the network. In Figure 6, executing ping between hosts in Mininet is done with the following command:

```
mininet>pingall
```

```

mininet> pingall
*** Ping: testing ping reachability
h1 -> h2 h3 h4
h2 -> h1 h3 h4
h3 -> h1 h2 h4
h4 -> h1 h2 h3
*** Results: 0% dropped (12/12 received)

```

Figure 6. Ping between hosts-Screenshot

The available bandwidth performance is increased in real time traffic than best-effort traffic as shown in Figure 7. It represents utilized bandwidth for each port between switches and hosts with 5Mbps traffic and 10 sec. The results show that real time traffic is more than the best-effort traffic.

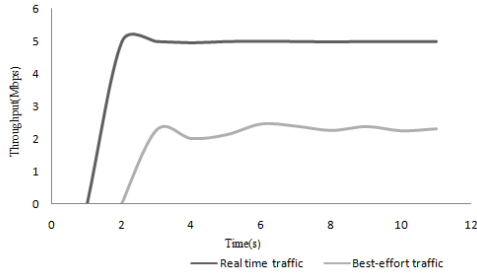


Figure 7. Available bandwidth-test with SR(5Mbps traffic)

Figure 8 also represents how the available bandwidth performance is increased in real time traffic than best-effort traffic. It represents utilized bandwidth for each port between switches and hosts with 10Mbps traffic and 10 sec. Results obtained show that real time traffic is more than the best-effort traffic.

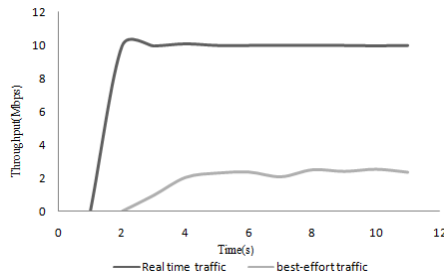


Figure 8. Available bandwidth-test with SR(10Mbps traffic)

The end-to-end delay for both switches is presented in Figure 9. The end-to-end delay is the sum of the delays experienced at each hop on the way to the destination. Test results in Figure 9 indicate that the average delay is 3.5ms for best-effort traffic and 0.94ms for real time traffic respectively.

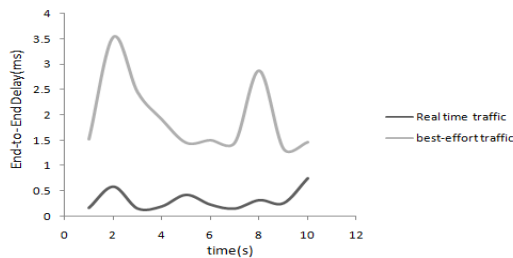


Figure 9. End to end delay-test with SR

The result of the bandwidth allocation algorithm is shown in Figure 10, which provides the distribution of SR path lengths. It is interesting to note that the mean number of hops in a SR path is 2 and the maximum number is 4.

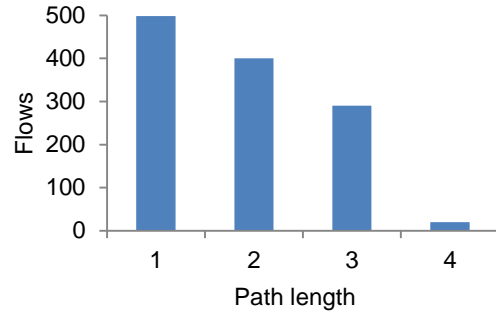


Figure 10. Distribution of SR path length (number of SIDs in the path)

6. Conclusion

In this paper we tackled the problem of Available Bandwidth calculation and monitoring in Software Defined Networks. We proposed a bandwidth allocation algorithm based segment routing for real-time and best-effort traffic. In this paper, we presented bandwidth allocation scheme based on SDN to provide Quality of Service (QoS) support for various services via experiments under our SDN testbed using Mininet for network emulation and ONOS for SDN controller with different kinds of traffic.

In the future work, we will further investigate to introduce other QoS parameters into QoS metrics, such as packet loss and reliability. We also used the Dijkstra algorithm to implement the bandwidth allocation. Then we will plan to extend the investigation of the local fast recovery with a large number of flows.

7. References

- [1] B.Pankaj. "ONOS Open Network Operating System An Open-Source Distributed SDN OS". 2013 Dec, 19, p.34.
- [2] S.Tomovic, I.Radusinovic, and N.Prasad, "Performance comparison of QoS routing algorithms applicable to large-scale SDN networks". In EUROCON 2015-International Conference on Computer as a Tool (EUROCON), IEEE 2015 Sep 8, (pp. 1-6).
- [3] H.Cho, J.Park, J.M.Gil, Y.S.Jeong, and J.H.Park, "An Optimal Path Computation Architecture for the Cloud-Network on Software-Defined Networking". Sustainability, 7(5), 2015 May 5 , pp.5413-5430.
- [4] U.Pongsakorn, I.Kohei, U.Putchong, D.Susumu, and A.Hirotake, " Designing of SDN-Assisted Bandwidth and Latency Aware Route Allocation". (HPC), 2014 Jul 21, pp.1-7.

[5] T.T.Nguyen, and D.S.Kim, “Accumulative-load aware routing in software-defined networks”. In Industrial Informatics (INDIN), IEEE 13th International Conference on 2015 Jul 22 (pp. 516-520).

[6] P.Megyesi, A.Botta, G.Aceto, A.Pescapè, and S.Molnár, “Available bandwidth measurement in software defined networks”. In Proceedings of the 31st Annual ACM Symposium on Applied Computing 2016 Apr 4 , (pp. 651-657).

[7] P.Megyesi, A.Botta, G.Aceto, A.Pescapé, and S.Molnár, “Challenges and solution for measuring available bandwidth in software defined networks”. Computer Communications, 99, 2017 Feb 1, pp.48-61.

[8] C.Cetinkaya, E,Karayer, M.Sayit, and C.Hellge, “SDN for segment based flow routing of DASH”. In Consumer Electronics–Berlin (ICCE-Berlin), 2014 IEEE Fourth International Conference on 2014 Sep 7 (pp. 74-77).

[9] D.J.Hamad, K.G.Yalda, and I.T.Okumus, “Getting traffic statistics from network devices in an SDN environment using OpenFlow”. ITaS, 2015 Sep, pp.951-956.

[10] R.Kumar, M.Hasan, S.Padhy, S., K.Evchenko, L.Piramanayagam, L., S.Mohan, and R.B,Bobba, “End-to-End Network Delay Guarantees for Real-Time Systems using SDN”, 2017.

Data Science

Uniformly Integrated Database Approach for Heterogenous Databases

Hlaing Phyu Phyu Mon, Thin Thin San, Zinmar Naing, Thandar Swe
University of Computer Studies (Meiktila), Meiktila, Myanmar
hlaingphyuphyumon16@gmail.com

Abstract

The demands of more storage, scalability, commodity of heterogenous data for storing, analyzing and retrieving data are rapidly increasing in today data-centric area such as cloud computing, big data analytics, etc. These demands cannot be solely handled by relational database system (RDBMS) due to its strict relational model for scalability and adaptability. Therefore, NoSQL (Not only SQL) database called non-relational database is recently introduced to extend RDBMS, and now it is widely used in some software developments. As a result, it becomes challenges regarding how to transform relational to non-relational database or how to integrate them to achieve business purposes regarding storage and adaptability. This paper therefore proposes an approach for uniformly integrated database to integrate data separately extracted from individual database schema from relational and NoSQL database systems. We firstly try to map the data elements in terms of their semantic meaning and structures with the help of ontological semantic mapping and metamodeling from the extracted data. We then cover structural, semantical and syntactical diversity of each database schema and produce integrated database results. To prove efficiency and usefulness of our proposed system, we test our developed system with popular datasets in BSON and traditional sql format using MongoDB and MySQL database. According to the results compared with other proficient contemporary approaches, we have achieved significant results in mapping similarity results although running time and retrieval time are competitive with the others.

Keywords- Relational database, NoSQL database, ontological semantic mapping, database schema

1. Introduction

In today's IT software development, every developing process needs to use database for storage, analyzing and retrieval of various kinds of data depending on their goals. As IT technologies advances with evolution of cloud computing, big data, etc, it needs to store tremendous amount of data and information in different kinds of development platforms. Consequently, the role of relational database for

management and storage purpose becomes insufficient due to lack of large capacity, scalability and heterogenous capability to work with advanced database products and needs. Therefore, new innovation called NoSQL database system evolves so that the needs of current technology demands can be supplied. Meanwhile, the usage of relational DBMS cannot be discarded because many software products are still using RDBMS due to its rich features and usefulness. Therefore, there is a need to build a bridge for those two types of databases so that they can be integrated for simultaneously logical needs of data from physically distributed databases over heterogenous data sources [1].

In integrating the data from separated relational systems into a new one, there exit a lot of solutions [1]. However, to the best of our knowledge, there is only few researches [1,2,3] to integrate distributed relational and nonrelational database. There are many challenges to combine different complex database structures and schemas so as to migrate all required information of each into a new database one.

While relational database management system depends on relational data model such as MySQL, Oracle, PostgreSQL, etc, non-relational NoSQL management systems are using various kinds of semi-structured data model such as key-value stores, column stores, graph stores, etc [4,5,6]. Therefore, many scholarly works are being demanded to address the issues of structural mapping upon different syntactic and semantic structure, understanding the semantic meaning of database elements and relationships. Our paper therefore takes these challenges as research opportunities to figure out how data elements, relations and structures are semantically mapped with the use of ontological semantic definitions and how to transform them to well organized new database.

The contribution of this paper is introducing how to semantically map database definitions among different database elements, relations and structures without consideration of schema mapping, database aggregation and joins among different data sources. For our purpose, we particularly use MySQL (sql¹ file extension) for relational database and MongoDB (bson² file types) for

¹ sql-structured query language

² bson-binary json

non-relational database. MongoDB which acts like database as service over cloud network using rich storage structures and query languages [7].

The remainder of the paper is organized as follows. A brief note for background theory of NoSQL and relational database, and data integration problems and solutions are described in Section 2. In Section 3, we explain the structure and solution of proposed system and the analysis are described in section 4. We finally conclude the paper in Section 5 by exploring our intended future works.

2. Preliminary study

2.1. Background theory

2.1.1. NoSQL Database vs Relational DBMS. The term NoSQL was first introduced in 1998 for relational database to skip the use of SQL [10]. The term was used again in 2009 at the conferences of advocates of non-relational databases NoSQL meetup in San Francisco [11]. It is designed for rapidly iterated changing environment especially in agile software development process so that a significantly higher data throughput is produced, horizontal scalability is supported for huge volume of data storage and commodity hardware for more cost-effective alternatives.

Relational RDBMS database systems were developed in 70's to store structured data in the form of table with their own query language model called structured query language (SQL) [12].

In contrast to RDBMS, NoSQL uses structural, semi-structure and unstructured documents to store the data, and enables to scale the storage volume well in the horizontal direction for very large amount of data which are desperately demanded in cloud computing and big data storage. Moreover, the design of NoSQL does not rely on highly available hardware, and it challenges the shortcomings of RDBMS such as rigid schema design, performance of single servers and limited storage data (eg. 50 GB for inbox search at Facebook or 2PB in total at eBay).

2.1.2. Data Integration Problems and Solutions. The advent of NoSQL gains a great attention of research scholars and have been evolving many achievements and proposals to enhance NoSQL techniques. Among them, data integration from different databases involves with specific problems and solutions.

A logical integration of data separately stored in different databases reduces time-consuming, cost and human made errors for the processes which are using manual integration. Furthermore, a semantic based logical integration can handle complex structures and meanings of data elements which are going to combine as new one. Although there are many popular database

drivers such as JDBC, OLE DB, etc which use generalized query languages. They are also able to handle different database management systems but they lack of capabilities to work on structural, semi-structural and semantic differences of data sources [1]. Therefore, we need to develop a systematic integrated approach that can understand semantical and syntactical meanings of data elements, relationships and structures of different data sources so as to integrate different structures and schema types of relational and non-relational databases.

2.2. Literature review

The popularity of NoSQL becomes heated since very recent years. As it is, many scholarly works studies and proposes some advanced features and methods to interoperate NoSQL. However, only a few studies empathize on integration of data stored in NoSQL systems [13,14,15]. The research works [1, 16] propose uniform interface and platform to integrate databases. Whereas the work [1] presents uniform access platform to collect data from different separated database management system, the work [16] proposes a uniform interface that allows to access the data stored in different NoSQL systems (HBase, Redis, and MongoDB). The paper work [17] presents a framework to seamlessly fill the gap of SQL deficits with the help of document stores structure of NoSQL.

As explained above, data integration among different databases plays a key role in migration process. Therefore, in our paper, we propose an approach to integrate data from different sources without a need of concept of both relational and non-relational databases for the user and programming skill. They just need to load the databases they want and our system will map the required process and deliver the merged database in non-relational format (JSON³) to the users.

3. Problem architecture and solutions

3.1. Problem architecture

The architecture of our proposed approach integrates the idea of HybridDB [1] and our novel idea in integration of heterogenous databases. The workflow of our architecture initiates when a user request is received. The user request will be importing the databases they want to merge (Mysql and MonoDB files in our paper).

The user inputted databases files are accepted by database controller and query the database views, table views and dataset results with the aid of particular native driver of each different database: MySQL and

³ json-javascript object notation

MongoDB. The resulted query results are relayed to database modular that extracts particular connection and specification parameters for data records contained in each database. In this case, each database file may contain more than one table. The user is allowed to use any number of table for each database. We regard that those databases are already normalized. After database modular separates each table of each database definitions, the database manipulator organizes them into similar semantic concepts and maps each element (name, value types, relationships, structures, etc) of different data sources with the help of DB ontology. The sample scenario can be seen in Figure 1.

The core part of this architecture is database controller that accepts inputs, executes database operations on the source system with the aid of database manager which can access and control native drivers of all database types allowed by this system.

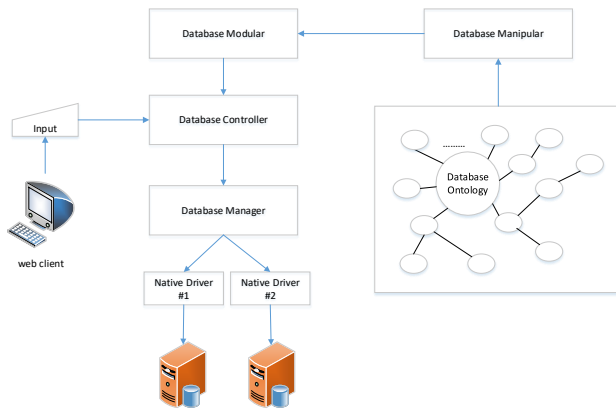


Figure 1. Architecture of proposed system

3.2. Problem Solution

3.2.1 Managing database operations: This process deals with inputted database files depending on their database types. The scenario we consider in this paper is staff information list of a university.

```

db.createCollection("staff_profile");
db.staff_profile.insert([
  {staff_ID:1,
  name:"Thein Tun",
  position: "Lecturer",
  address:{
    street:"PyiTawThar",
    city:"Yangon"
  },
  contact:[
    {name:"U Myo Myint",relationship:"Father"},
    {name:"Daw Sein",relationship:"Mother"}
  ]
}, {...}, {...}]);

```

Figure 2(a). CRUD database operation of MongoDB

The user may enter the staff information in two different files: .sql and .json file for MySQL and MongoDB integration. Our approach then uses different CRUD (create, insert, update and delete) operations for each particular database shown in Figure 2 (a) and (b) without human participation.

In the first place, after getting user inputs, the system will try to query data, value, data types, relationship and structure separately for each database type. The database controller and manger work together to get connection to native drivers and get all possible information on those inputted database script files.

```

create database staffs;
create table staff_profile {
  ID int (PK), staff_name varchar(30), rank varchar
(30), salary varchar(10), phone_number
varchar(15), town varchar(10)
};
create table staff_contact{
  ID int (FK),
  name varchar(20),
  relationship varchar(20)
};
insert staff_profile (1,'Thein Tun','Lecturer','200,000
MMK','00959-*****','Yangon');

```

Figure 2(b). CRUD database operation of MySQL

3.2.2 Structural, semantical and syntactical mapping:

The database manipulator understands the heterogeneity of data structure, relationships and semantic meaning of data objects of both database files with the help of database ontological structure. Here, we assume that there will be some relationships between two databases. The ontology extracts a real connection between data objects of both files and translates them, removes duplicate records, attributes and sometimes transforms some data into another types and structures. For those cases, we build ontology based on database terminologies and possible relationships of the dataset. In this paper, we train our ontology structure with 35 instances of datasets and test their usefulness with 800 datasets in evaluation stage. We use Protégé for structuring ontology and use OWL-API java platform for querying ontological meanings upon Tomcat web server.

3.2.3 Organizing integrated database file:

After understanding and manipulating the inputted two database files, the database controller merges them into new database one in NoSQL format, .json file format in this paper. The result file is then tested by opening the connection its native driver and perform essential CRUD operations before delivering to the users so that encountered errors can be solved in this stage. The final result for example scenario is shown in Figure 3.


```

db.staff_profile.insert([
  {
    staff_ID:1,
    name:"Thein Tun",
    position: "Lecturer",
    salary:"200,000MMK",
    address:
      {
        street:"PyiTawThar",
        city:"Yangon",
        phonenumber::"0095-9-***-***"
      },
    contact:
      [
        {name:"U Myo Myint",relationship:"Father"},
        {name:"Daw Sein",relationship:"Mother"}
      ]
  },
  {...},
  {...}
]);

```

Figure 3. Integrated database result

4. Experimental Results

4.1 Implementation Setting

The proposed system is developed with laravel 5.3 MVC framework and angularjs for front and back-end interfaces. Tomcat server is used for web server, and OWL-API is used to build semantic information of database elements and structures. For two different databases types, as mentioned earlier, sql file for MySQL and BSON (binary JSON) file for MongoDB are used. The datasets are download from the database [18,19] and tested with 35 instances of database with different 800 datasets. The result file is produced as json format to be compatible to run on any NoSQL database.

The system is implemented on a window 10 PC equipped with 3.10 GHz, Intel® Core TM of CPU and 4.0 GB of RAM.

4.2 Experimental Results

The system performance is evaluated with three main parameters: similarity rate, retrieval time and throughput time. These criteria are measured by varying database sizes and number of different datasets as illustrated below. To prove competitive results, we compare our evaluation results with other proficient works called HybridDB[1] and SOS platform [13].

a) impact of dataset size

We measure the retrieval and throughput time by varying the sizes of databases. As shown in Figure 4(a), both of retrieval times in MySQL and MongoDB become significantly low in all compared approaches when the database size increases as general theory. The retrieval and total throughput time for particular database size: small dataset (below 80 rows and below 10 columns), medium dataset (between 80 and 5000 rows, and between 10 columns and 30 columns) and large datasets (between 5000 and 10,000 rows and between 30 and 50 columns).

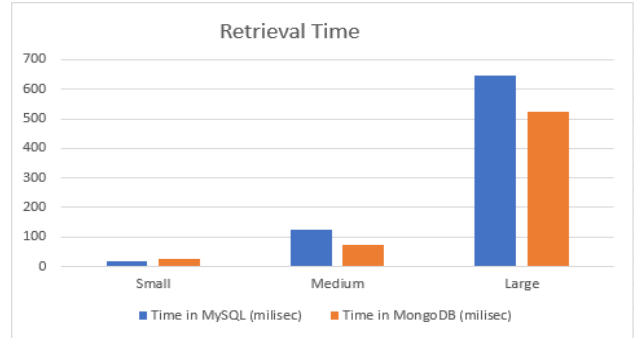


Figure 4(a). Measurement of retrieval time

The retrieval time starts when user request is sent from our system to native data source until getting the query results from them. For smaller database size, MySQL can work faster than MongoDB. For relatively increasing database size, MongoDB gets significant results in its speedy database operation.

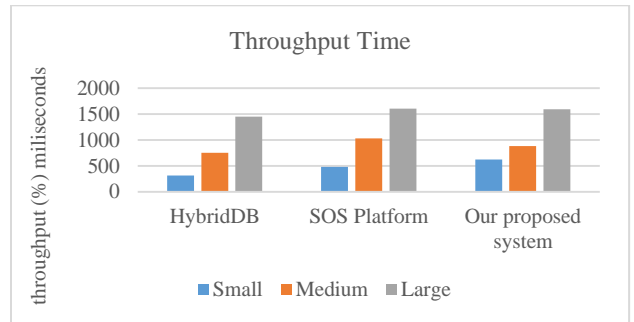


Figure 4(b). Measurement of throughput time

The throughput time means the total time since user inputs the files and until they receive the results. We got relatively similar result in these two retrieval and throughput time compared with HybridDB and SOS platform due to their significantly competitive methods.

b) impact of different dataset numbers

The similarity rate is measured how database ontology matches the elements, relationships and structures of two inputted database files. The higher value of similarity rate means exact similarity and the lower value describes higher dissimilarity. To test the similarity rate to show the efficiency and usefulness of ontological usage, we investigate our proposed system with compared works by testing different 800 datasets which are significantly different of major 35 instances of database files.

According to results described in Figure 4(c), our similarity rate is significantly higher in density of database sizes because the more classes we consider in mapping, the higher the similarity rate we can find due to semantic technology. For smaller data sets, the result

is not much different with popular approach HybridDB in this field while our result is better than SOS platform.

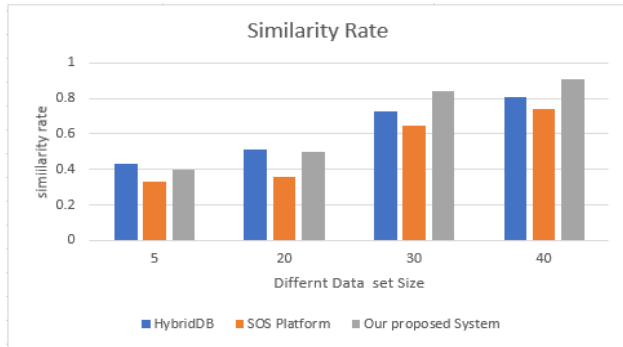


Figure 4(c). Measurement of similarity rate

5. Conclusion and future work

This paper has tried to fill the gaps of database integration problem by innovating uniformly integrated approach for different databases. We contributed an ontological mapping to understand semantic structure of data elements of relational and non-relational data sources. As the limitation of this paper, our system will be able to integrate two databases which are not zero relationships between them. The experimental proved that we have better results in similarity rate which is mostly needed in database integration area in order to minimize the complex and subtle meaning of data schema. We will extend this proposed work for further database operations such as join, aggregation, etc that are current limitations of this paper.

6. References

- [1] A. V. Fogarassy, T. Hulyak, "Uniform data access platform for SQL and NoSQL database systems", *Information Systems*, 69 (2017), pp.93-105.
- [2] W. Allen, "Unified data modeling for relational and nosql databases", 2016. <https://www.infoq.com/articles/unified-data-modeling-for-relational-and-nosql-databases>
- [3] R. Sellami, S. Bhiri, B. Defude, "Odbapi: a unified rest api for relational and nosql data stores", in: 2014 IEEE International Congress on Big Data, IEEE, 2014, pp. 653-660.
- [4] V. Abramova, J. Bernardino, P. Furtado, "Experimental Evaluation of NoSQL Databases", *International Journal of Database Management Systems (IJDMS)*, Vol. 6, No.3, June 2014.
- [5] B.G. Tudorica, C. Bucur, "A comparison between several nosql databases with comments and notes", in: 2011 RoEduNet International Conference 10th Edition: Networking in Education and Research, 2011, pp. 1-5
- [6] L. Dobos, B. Pinczel, A. Kiss, G. Racz, T. Eiler, "A comparative evaluation of nosql database systems", *Anales Universitatis Scientiarum Budapestinensis de Rolando Eotvos Nominatae Sectio Computatorica* 42 (2014), pp.173-198.
- [7] <https://www.mongodb.com/>
- [8] D. Kumawat, A. Pavate, Correlation of NOSQL & SQL Database, *Journal of Computer Engineering (IOSR-JCE)*, volume 18, issue 5, 2016, pp. 70-74.
- [9] A. B. M., Moniruzzaman and S A Hossain. "Nosql database: New era of databases for big data analytics-classification, characteristics and comparison." arXiv preprint arXiv:1307.0191 (2013).
- [10] Strozzi, Carlo: No-SQL-A relational database management system. 2007-2010. http://www.strozzi.it/cgi-bin/CSA/tw7/I/en_US/nosql/Home%20Page
- [11] Evans, Eric: NoSQL 2009, May 2009. Blog post of 2009-05-12. http://blog.sym-link.com/2009/05/12/nosql_2009.html
- [12] Fayeche, I. and Ounalli, H.: "Towards a Flexible Database Interrogation", *International Journal of Database Management Systems (IJDMS)* Vol.4, No.3, June 2012 .
- [13] P. Atzeni, F. Bugiotti, L. Rossi, "Uniform access to non-relational database systems: The sos platform" in: *Advanced Information Systems Engineering Springer*, 2012, pp. 160-174.
- [14] O. Cure, F. Kerdjoudj, D. Faye, C. Le Duc, M. Lamolle, "On the potential integration of an ontology-based data access approach in nosql stores", *Int. J. Distrib. Syst. Technol. (IJ DST)* 4(3) (2013) 17-30.
- [15] Y. Yuan, Y. Wu, X. Feng J. Li, G. Yang, W. Zheng, "Vdb-mr: mapreduce-based distributed data integration using virtual database", *Future Gener. Comput. Syst.* 26 (8) (2010) 1418-1425.
- [16] P. Atzeni, F. Bugiotti, L. Rossi, "Uniform access to nosql system", *Inf. Syst.* 43 (201) 117-133.
- [17] J.Roijackers, G. H. Fletcher, "On bridging relational and document-centric data stores", in: *Big Data*, Springer, 2013, pp. 135-148.
- [18] <http://jsonstudio.com/resources/>
- [19] <http://www.mysqltutorial.org/mysql-sample-database.aspx>

Analysis of Historical Census Household data with Similarity Threshold Method

Khin Su Mon Myint, Thet Thet Zin, Kyaw May Oo

University of Information Technology

Yangon, Myanmar

ksmonmyint@uit.edu.mm, thetthetzin@uit.edu.mm, kyawmayoo@uit.edu.mm

Abstract

Historical census data contains valuable information of families in a country. It captures information about ancestors. These data can be used to reconstruct important parts of a specific period in order to trace the households and families changes across time. Linking census data is a challenging task due to poor data quality, household changes over time. During the decades, a household may split multiple households due to marriage or moving to another household. This paper introduces an approach for data cleaning, standardization and linking of historical census data across time. The key fact of the proposed approach is firstly to detect households, clean and unified into standard format. After cleaning these records, approximate string similarity measures are used to link individual records and then define matched and unmatched records with similarity threshold method. The result of the experiment shows optimal threshold value which is efficient for household linkage.

Keywords- historical census data, data cleaning, data matching; record linkage, household linkage, and pairwise linkage

1. Introduction

Population census data provides valuable information of households in a region. They play an important role in analyzing the social, economic, and demographic aspects of a population [4, 5, 6].

Population census data collects every 10 years. These data allows us to reconstruct the aspects such as birth, death, education, occupation, etc. They help organization how our ancestors of the social and demographic changes in the country.

Linking records refer to the same households from several censuses which gives across the decades will greatly enhance in value. The linked results have been allowed to trace varies in the characteristics of individual households over time.

Linked information improves not only retrieval of information, but also provides new opportunities for improving the quality of the data. It can help social scientists with dynamic character of social, economic and demographic changes [9], which helps the reconstruction of the region.

Difficulties of historical census data linkage occur from several facts. These include poor data quality due to census data collection process. Importantly, the situation of individuals in a household may vary significantly between two censuses. For example, people are born and die, get married, change occupation, or moved home. As a result, linking individuals is not reliable, and many false matches are often generated.

Due to the benefits of historical census data linkage, there are large amount of data available, automatic or semi-automatic linking methods have been developed by data mining researchers and social scientists [3, 4, 5, 6]. These methods treat historical census data linkage as a special case of record linkage, and apply string comparison methods to match individuals. Some researchers use classification algorithms to classify matches or non-matches and use group linking approach to link households based on the matched records [2].

Most of researchers aim to find households with the majority of their members matched. However, during the ten years interval between two censuses, a household may split into multiple households due to marriage or moving to another household, or changing servants' jobs.

Most previous works in census household linking problem can only be matched each individual in one household to one individual in another household.

This paper proposes an approach for data cleaning, standardization and linking of historical census data using domain knowledge. This work considers each individual in one household to one individual in another household and also takes multiple household linking.

The main idea is to use household information in the cleaning and linkage steps. So, these records which contain errors and variations can be cleaned and standardized and the number of incorrect linked records can be reduced. The proposed approach starts by detecting Household Identifiers (HHIDs). These HHIDs, together with name, address, gender, and relationship to the household head attributes, are used to clean the data. Record linkage is performed on record pairs, and then the linking results are improved using similarities results.

The rest of the paper is organized as follows. Section 2 introduces related works in data cleaning and linking, as well as their application to historical census data. Section 3 introduces the historical census data collected from United Kingdom. In Section 4 gives an overview of the proposed approach. Section 5 describes cleaning

and standardization method and pairwise linking approach. The experimental results are reported in Section 6, and conclude this paper in Section 7 and point out future research directions.

2. Related Work

Difficulties of historical census data linkage came from several scenes. These include poor data quality and large amount of similar values in names, address and ages.

It has a more important fact that the condition of individuals in a household may vary significantly between two census periods. For example, people are birth and death, marriage, movement home or changing occupation.

As a result, linking individuals is not reliable and many false matched are generated. This is also a common problem in record linkage applications.

To improve the quality of historical census record linkage, it is very important to examine domain driven approaches. The understanding of the domain social sciences needs and combines this knowledge with the data cleaning and record linkage methods by the computer science community [3, 7, 8].

In few years, Christen et al. [2, 3] have proposed probabilistic data cleaning techniques for names and address that outperform traditional rules-based approaches. Christen has presented an overview of both pattern matching and phonetically encoding based name matching techniques.

In recent years, computer science researchers have been developed new record linkage techniques that can be used to meet the challenges presented by linking historical census data.

P. Christen [1] proposed a method by supervised learning and group linking methods to link historical census households across time. This approach first computes the similarity between record pairs and uses these similarities as input to Support Vector Machine (SVM) classifier, which classifies record pairs into a matched and non-matched class. They used group linking techniques to generate household linking similarities.

One problem in the above methods for historical census matching is that matching is performed on same household matching. However, a household may split multiple households between two censuses. So, previous proposed methods cannot get accurate household matching results.

This paper considers not only for the same household matching but also for the multiple household matching using Similarity Threshold Method.

3. Data Collection

This work uses two census datasets collected from the Ireland censuses within ten-year intervals (between 1901 and 1911)[10]. The data were collected on hand-

filled census forms which contain eleven attributes such as the address of the household, full names, ages, sexes, their relationship to the household, occupations and places of birth.

The quality of these digital forms varies a lot, due to the way the returns were completed and scanned. The next step of digitisation was a manual transcription of the digital form into tables and storing them in electronic spreadsheet tables. Table 1 shows a sample of census data in a spreadsheet.

Table 1. Sample census data

Surname	First name	Age	Sex	Relation to Head	Birthplace
Cairns	William	52	M	Head of Family	Co Antrim
Cairns	Letitia	51	F	Wife	Co Antrim
Cairns	Thomas John	24	M	Son	Co Antrim
Cairns	William Edwin	22	M	Son	Co Armagh
Cairns	Herbert Lavelet	20	M	Son	Co Antrim

The dataset contains records for each person in the district. There are 11 attributes for each record, which correspond to some important aspects of households. These attributes are shown in Table 2.

The purpose of this step is to improve data quality from the raw census data. It is applied for improving the quality of the data and formatting the data to a unified form. The census data return form was filled in by hand.

These include missing values, inconsistent values, and wrong values. Errors were introduced in these stages. An example is FIRST NAME attributes with digits, letters, punctuation, and other symbols which require cleaning and standardisation to be applied.

The other example is the type of AGE attribute, which is mixture of digits and letters. This implies that the values were entered in different formats. Therefore, data standardization is required.

It is aimed to improve the quality of the data and format the data to a reliable format in data cleaning and standardisation step.

Data cleaning step aims at removing the errors and missing values in the data. It applies look-up tables to eliminate records without meaningful values, and to replace incorrect attribute values with correct values.

Table 2. Census data attributes with descriptions

Attribute	Description
HHID	Id of the house
SURNAME	Surname of person in the house
FIRST NAME	First name of person in the house
RELATIONSHIP	The relationship to the head of the household

SEX	Gender of the person
AGE	Age of the person
BIRTHPLACE	Address of the person
RELIGION	The relation of person
OCCUPATION	The occupation of the person
MARITAL STATUS	The marital status of the person
LITERACY	The literacy status of the person

An example of data cleaning of gender values, for example, value “mm” is replaced with “m”. The standardization step formats the data into a unified form such as field names were standardized to uppercase letters and attributes values were converted to lowercase letters. It includes several operations, for example removing non-meaningful values such as “=”, “?” and non-standard words, such as “no entry” and “not identified” and unifying the age format into digits-only.

The purpose of household ID Detection is to assign a unique household ID (HID) to each household. In each census form, the relationship to the head of household attribute always starts with the head of household. A record in the household has a head of household role, the HID number is incremented by one, and this HID number is assigned to all following records until another record with a head of the household role is found.

4. Overview of Proposed Approach

The proposed approach constitutes four steps, as illustrated in Figure 1. The first step is data cleaning and standardization which solving the low quality data problem in historical data collection. The purpose is to find missing values, as well as to transform the data into a standardised form. This step also provides the data quality and increases the finding of true record matches between two datasets.

The second step is household detection, which assigns unique household ID (HHID) to each household. The HHIDs are used to define the household in the future.

The third step is blocking and indexing. In this step, datasets are subdivided into several blocks using a blocking keys (index keys), only records in the same block are compared with each other, that greatly reduces the number of record pairs which need to be compared and so speeds up the linkage process. Only record pairs which have an identical blocking key are compared with each other.

The fourth step is the record pair comparison which aims to find similarities between records. Several similarity methods have been used for this purpose. Finally, candidate record pairs are classified into matches and non-matches.

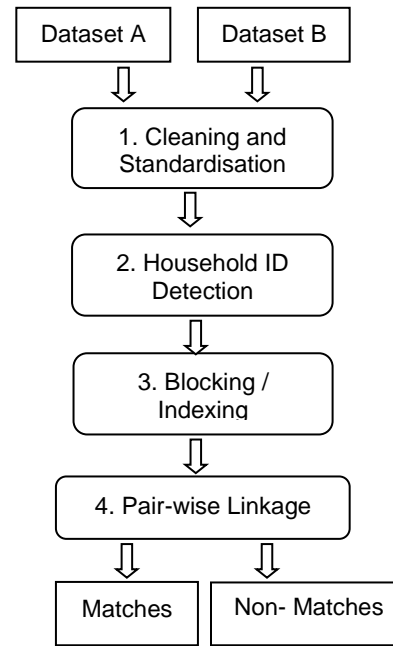


Figure 1. Historical census linkage process

5. Cleaning and Linking Historical Census Data

5.1. Data Cleaning and Standardisation

The data cleaning and standardisation tasks are applied for getting the better quality of the data and structuring the data to a unified form. Many non-meaningful values are in the data. These include symbols such as “=”, “?”, and non-standard words, such as “no entry” and “not identified”. All these values are not useful. These values have been removed to improve the data quality.

The standardization step includes several operations. They are:

- All values are converted into lowercase letters
- First and middle names are split into two attributes
- The age format into a digit-only format that represent an age as number of years

5.2. Automatic Household Detection

In the census table, the value for the Relationship attribute for each household should start by the head of the household. Based on the domain knowledge, possible values for the head of the household are “head”, “head of family”, “widow”, “widower” and “husband”. We have been developed a linear algorithm to scan through census data file. If the record has a head of household role, the household ID (HHID) number is incremented by one, and this HHID is assigned to all

rest records until other record with a head of household role is found. Algorithm 1 describes the construction of unique household ID.

Algorithm 1: HouseholdID Detection Algorithm

Input:
 - All households in the dataset

Output:
 - All households with unique household ID

1. household_ID = 0
2. for record \in House do
3. Get "Relation to Head" field value in record
4. If relationHead == "head of family" || relationHead == "head" || relationHead == "widow" || relationHead == "widower" then
5. household_ID = household_ID + 1
6. End If
7. End for

5.3. Blocking/ Indexing

Before the linking process, it is firstly applied a blocking technique to reduce the complexity of pairwise linking. This technique subdivides the datasets into several blocks, so only records in the same block are compared. When large datasets are used, the linking process is very time consuming. It is due to compare all pairs of records from both datasets.

The four key attributes (SURNAME, FIRST_NAME, SEX and ADDRESS) are selected and used Double Metaphone encoding algorithm to generate blocking keys. Double Metaphone phonetic algorithm allows multiple encodings for strings that have various possible pronunciations. This step really speeds up to the linking process.

The following three blocking keys are applied:

- first three letters of "SURNAME" attribute with "Double Metaphone" concatenated with the "SEX" attribute
- first three letters of the "FIRST_NAME" attribute with "Double Metaphone" concatenated with first four letters of the "ADDRESS" with "Double Metaphone"
- first three letters of the "FIRST_NAME" attribute with "Double Metaphone" concatenated with first four letters of the "SURNAME" with "Double Metaphone"

It was assumed that the true matches occur within the identical blocks. Only records which have the same block are compared with each other. Therefore, the time of record linkage process speed up.

5.4. Household linkage

When comparing records, appropriate approximate string comparison functions have been chosen for each

attribute. The list of attributes and functions used to compute the similarities between values is shown in Table 2. If the score of records are higher, the two attributes are more similar (scores of 1 indicate an exact match, 0 means no similarity).

Table 3. Similarity methods used for the five attributes

Attribute	Method
SURNAME	Q-gram
FIRST NAME	Q-gram
SEX	String extract match
AGE	Gaussian probability
ADDRESS	Longest common subsequence

Q-gram based approximate string comparison is applied on "SURNAME" and "FIRST NAME" attributes. Q-gram based approximate string comparison is to split the two input strings into short sub-strings of length q characters (called q-grams). This method is used to compare two strings based on q-grams value. In our experiment, we defined q-value is 2.

In "SEX" attribute, string extract match algorithm is applied to compare two sex values. Gaussian probability is used to compare the different "AGE" values.

Longest common subsequence is used to compare "ADDRESS" attribute. This algorithm repeatedly finds and removes the longest common sub-string in the two strings compared, up to a minimum length (sets to 2 or 3).

The attribute-wise linking generates a similarity score for each attribute. A vector $R_s(r_{t,i,j}, r_{r,i,j})$ can be got for record $r_{t,i,j}$ from one dataset and $r_{r,i,j}$ from another dataset. We denoted the similarity vector as $R_s(r, r')$. By summing over all attribute-wise similarity scores, a total similarity score $R_{sim}(a, b)$ can be calculated.

For $R_{sim}(a, b)$, the larger the similarity value, the more similar two records are. We find matched and non-matched category is comparing the similarity $R_{sim}(a, b)$ against a predefined threshold ρ . If $R_{sim}(a, b) \geq \rho$, the record pair is considered to be a match record pair. In the experimental section, we will discuss how the value for ρ is set based on the analysis of the linking results. After thresholding, multiple matches for a single record can be reduced.

6. Experimental Results

The aim of the experiments conducted was to evaluate the record matching using the different similarity threshold values. The goal was to get the optimal threshold value which achieves the best matching results for household linkages.

In this experiment, two census data from Ireland historical census datasets are used. These data collected

from the district of Aghagallon in Antrim in Ireland for the period of 1901 and 1911. There are 11 attributes for each record, full name, age, sex, relationship to the household head, occupation and place of birth et al. These data were standardised and cleaned before applying the household linkage step. In total, there are 96 and 97 records in the two datasets.

As mentioned previously, five attributes (SURNAME, FIRST NAME, SEX, AGE, and ADDRESS) were used in our study. After each of the attributes was cleaned, unique household ID (HHID) were identified.

Before pair-wise linking process, the datasets have been divided into many small blocks based on the three blocking keys as previously mentioned. This step tends to speed up our record comparison process.

Once the cleaned and identified household ID available, pair-wise linking is started with the records 1901 datasets compared with the records 1911 datasets. The record linkage step generated the similarity score of each selected attributes of the records, which the range of 0 and 1. If the scores are higher, the records are more similar. By combining all five score values, a total score $0 \leq S_{a,b} \leq 5$ can be calculated for each record pair $r_{a,b}$.

To define matched and un-matched record pairs, appropriate setting of the threshold value ρ is very important. The linking results with the respect value of the ρ are evaluated. The five threshold values (2.5, 3.0, 3.5, 4, and 4.5) are applied to evaluate the results as shown in Figure 2.

The number of records in the 1901 data set with exactly one matched record and with multiple matched records in the 1911 data set, when different threshold values ρ have been set.

The spread of single matched records and multiple matched records are different for different ρ value. The numbers of record with multiple matches have been reduced by increasing the ρ .

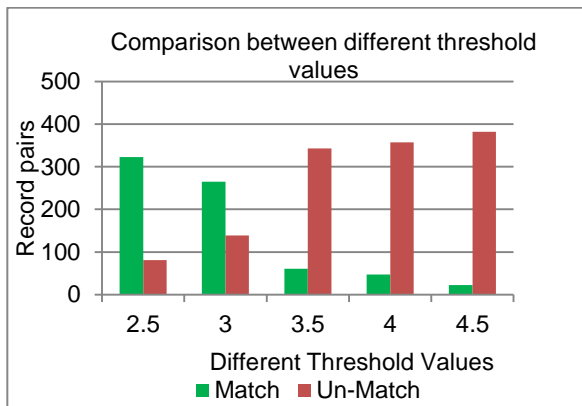


Figure 2. Matched and unmatched record pairs with different threshold ρ value

When ρ value is set to 4.5, which can be considered that only selected attributes are used, there are only 22

single match record pairs. However, many true multiple record pairs are still containing in the unmatched pairs.

When ρ value is set to 4, there are only 47 single (one-to-one) match record pairs but no multiple (one-to-many) matches. So, no multiple matches are found when $\rho > 4$. On the other hand, when ρ is too low, a lot of multiple false matches are generated.

The precision rate, recall rate and accuracy of the record pairs are evaluated on different threshold ρ values. Figure 3 shows the precision rate, recall rate and accuracy with different threshold values.

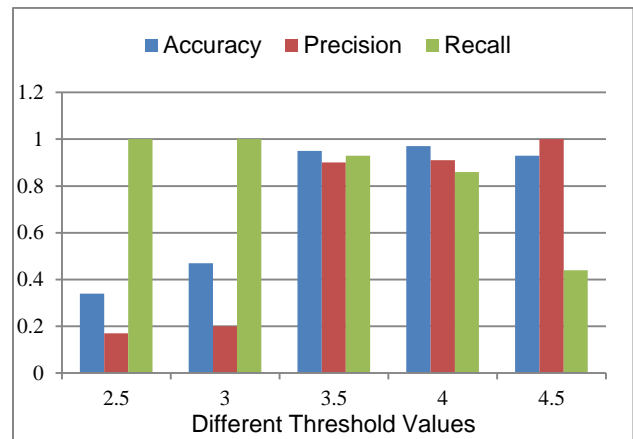


Figure 3. Comparison of performance of record linkage with different threshold ρ value

As the data shown in Figure 3, the precision rates for the five threshold values are from 17% to 100%, the recall rates are from 44% to 100% and the accuracy rates are from 34% to 97%.

It was found that threshold value 4 achieves the best accuracy rate of 97%, precision rate of 91% and recall rate of 86%. Although it had the best accuracy rate, it had missed many multiple (one-to-many) true links.

Threshold value 3.5 can provide 95% at accuracy, 90% at precision and 93% at recall rate. Threshold value 3 can provide 47% at accuracy, 20% at precision and 100% at recall rate. It generates many false matches.

By manually evaluating the results, threshold value 3 covers not only single match record but also multiple match records. It can provide one-to-one household linkage and one-to-many household linkage.

This suggests that 3.5 could be an appropriate threshold value for record linking in our system. Therefore, the experiment helps us to select the most appropriate threshold value for record matching.

7. Conclusion

In this paper, a data cleaning and linking approach with similarity threshold method for historical census data have been described. This approach uses household information to take the record cleaning and linking steps. The record linking is executed in two steps. The first step computes each record similarity scores using

approximate string matching algorithms. Then a pairwise record linkage is defined with the total similarity values by setting appropriate threshold values.

The experimental result shows that the matched and un-matched record pairs with different threshold values. The result also shows that ambiguous match results exist after the threshold step. This is due to the fact that structures of two households are very similar and family members can change substantially over time.

In the future, a classification algorithm will be explored which is used to improve more accurate one-to-one or one-to-many household matching performance. This includes household splitting into multiple households, children in that household getting married between the decades.

8. References

- [1] Z. Fu, P. Christen, Mac Boot, "A Supervised Learning and Group Linking Method for Historical Census Household Linkage", Proceedings of the 9-th Australasian Data Mining Conference (AusDM'11), Ballarat, Australia, 2011
- [2] Fu, Z., Christen, P., Boot, M.: Automatic cleaning and linking of historical census data using household information. In: IEEE ICDM Workshop. pp. 413–420 (2011).
- [3] P. Christen, "Development and user experiences of an open source data cleaning, deduplication and record linkage system," ACM SIGKDD Explorations Newsletter, vol. 11, no. 1, pp. 39–48, 2009.
- [4] Fure, E.: Interactive record linkage: The cumulative construction of life courses. Demographic Research 3, 11 (2000)
- [5] S. Ruggles, "Linking historical censuses: a new approach," History and Computing, vol. 14, no. 1+2, pp. 213–224, 2006.
- [6] Bloothoof, G.: Multi-source family reconstruction. History and Computing 7(2), 90–103 (1995)
- [7] D. V. Kalashnikov and S. Mehrotra, "Domain-independent data cleaning via analysis of entity-relationship graph", ACM Transactions on Database Systems, vol. 31, no. 2, 2006
- [8] B.- W. On, N. oudas, D. Lee, and D. Srivastava, "Group linkage" ,in Proceedings of the IEEE 23rd International Conference on Data Engineering, 2007
- [9] D. Quass and P. Starkey, "Record linkage for genealogical databases," in ACM KDD Workshop, Washington DC, 2003
- [10] <http://www.census.nationalarchives.ie>

Multidimensional Analysis for Census Data by Applying Star Schema Model

Myint Myint Thein, Myint Myint Lwin, Aye Chan Mon, May Thu Aung
*University of Information Technology,
University of Computer Studies (Maubin), Myanmar*
{myintmyintthein, myintmyintlwin, ayechanmon, maythuaung}@uit.edu.mm

Abstract

In recent years, the high value of multidimensional data has been recognized in both the academic and business communities. Star schemas are the primary storage mechanism for multidimensional data that is to be queried efficiently. It supports relationships between fact and dimension tables and creating combination dimensions with a key, resulted to improve query performance for large quantities of data. This paper is presented for multidimensional data model, that is called star schema to store large amount of census data. This star schema can be used for business related queries on Census data for visualization report. This paper aims to enhance the interactive visualization process with more relevant operations for manipulation of various attributes by using the Pentaho Business Analytics (BA) Suite.

Keywords—Agglomerative, census data, clustering, K-means

1. Introduction

In recent years, large multidimensional databases or data warehouses have become common in a variety of applications. Data warehouse (DW) is a collection of technologies that is enabling the decision maker to make better and faster decisions. The major challenge with these databases is to extract meaning from the data, they contain discover structure, find patterns and derive causal relationships. Collecting data for a Business Intelligence (BI) application is done by building a data warehouse where data from multiple heterogeneous data sources is stored. Transferring the data from the data sources to the data warehouse is often referred to as the Extract, Transform and Load (ETL) process. The data is

extracted from the source, transformed to fit and finally the data is loaded into the warehouse. The ETL process often brings issues with data consistency between data sources. In order to load it into the data warehouse the data has to be consistent, and the process to accomplish this is called data cleaning.

A star schema is a method of organizing information in a data warehouse that enables efficient retrieval of business information. The star schema represents to the end user a simple and query-centric view of the data by partitioning the data into two groups of tables: facts and dimensions. Facts are the data keys being organized around a large central table, and dimensions contain the metadata that is related to a set of typically smaller tables. Data stored in a star schema is defined as being “denormalized.” Denormalized means that the data has been efficiently structured for reporting purposes. The goal of the star schema is to maintain enough information in the fact table and related dimension tables so that no more than one join level is required to answer most business-related queries.

This paper represents star schema model for storing Census data. The data is retrieved from data sources by making ETL process in Pentaho Data Integration (PDI) open source tool and loaded into the star schema data warehouse on PostgreSQL database. It provides the useful information to the appropriate decision makers for business decision.

2. Related Work

Sudhir B. Jagtap and Kodge B.G. [4] have made an attempt to demonstrate how one can extract the local (district) level census, socio-economic and population related other data for knowledge discovery and their analysis using the powerful data mining tool Weka. Their primary available data such as census (2001), socio-economic data, and

few basic information of Latur district are collected from National Informatics Centre (NIC), Latur, which is mainly required to design and develop the database for Latur district of Maharashtra state of India. The database is designed in MS-Access 2003 database management system to store the collected data and analyzed data by using Weka tool.

M. Yost, J. Nealon presented a star schema model [1] that can be used for any census or survey to track the full history of the data series and to standardize the metadata. The dimensional model represents a relational database model that facilitates the gathering of a great deal of this information and knowledge about the data, stores it, organizes it and then relates it directly to the factual data being analyzed.

O.C. Okeke, B.C. Ekechukwu [3] proposed to develop mining model applicable to the analysis of Nigeria census data by harnessing the power of data-mining technique that could uncover some hidden patterns to get their geo-spatial distribution. This is represented decision tree learning for approximating discrete-valued target function and decision tree algorithm was used to predict some basic attributes of population in the census database.

N. Stolba, A.M. Tjoa [2] to explain the integration of data warehousing, OLAP and data mining techniques in the field of health care, and an easy to use decision support platform, which supports the decision making process is the process of building care providers and clinical directors. They offered three case studies, which show that the clinical data warehouse that facilitates evidence-based medicine is a platform for reliable, robust and easy to use to make strategic decisions, which are of great importance to the practice of medicine, and the acceptance of evidence-based.

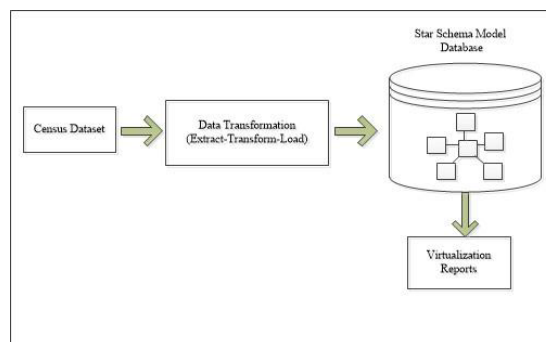


Fig. 1: Proposed System Flow

M.Khan and S.S.Khan [7] surveyed about various visualization methods. The authors point out different available visualization techniques are used for different situation and data mining techniques, mining results can present effectively by using visualization methods.

3. Proposed System Architecture

The 2014 Myanmar Population and Housing Census data was undertaken by the Ministry of Immigration and Population that is a fundamental source of information, it includes main categories such as education, fertility, mortality, migration, disability, population projections, gender, housing conditions and assets, youth, and elderly. In Fig. 1, this proposed system illustrates the transformation from dataset to product analysis reports by creating multi-dimensional model (star schema). The first stage started with data preparation for real census data which may involve cleaning data and selecting subsets of records related with education category. The second stage used data transformation (Extraction-Transformation-Loading ETL) tool is data extraction from raw data file (.excel) and then insertion into fact and dimension tables on the PostgreSQL database. The dimension tables are age, gender, literacy, attendance, grade and employment and the one fact table is education table. The final stage demonstrated the impact of age, literacy, attendance, grade and employment on the gender class to produce analysis reports by using Pentaho dashboard tool.

A. Star Schema Model

The relational model and the Structured Query Language (SQL) allows the user to efficiently and effectively manipulate a database. They record information in two dimensions table structure and automate repetitive tasks. Simple entity relationship diagram (ER) consists of many relationships between tables to retrieve query. For this retrieve query, the user needs to fulfill certain criteria. If the user could be required query result,

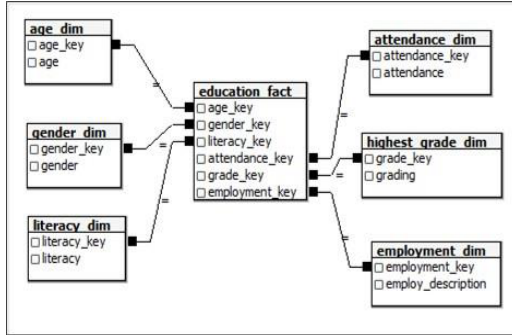


Fig. 2: Star Schema Design

Data warehousing provides an effective way to analyze the data and also uses query optimization by creating multi-dimensional data tables are called star schema. The star schema represents to the end user a simple and query-centric view of the data by partitioning the data into two groups of tables: facts and dimensions. Facts are the data keys being tracked, and dimensions contain the metadata describing the facts. A star schema consists of a collection of tables that are logically related to each other.

In Fig.2, shows a star schema organized our education information from census data, there are one education fact and six main dimensions: age, gender, literacy, attendance, grade and employment. The fact table relates data keys to each dimension. Each dimension tables contain detail information on persons responding to census data. For example, the age tables defines different age level on different persons.

the query operation needs at least one internal join across two tables. Thus, it could be acceded lower performance significantly because of the time required for joining many tables. [6] Another problem is that large amount of data in form of normalization requires a lot joins of many tables. If the user used this model for analysis, it responds very slowly to new analytical requirements.

B. Data Transformation Tool

The ETL tool has three tasks to build data warehouse: (1) data is extracted from different data sources, (2) propagated to the data staging area where it is transformed and cleansed, and then (3) loaded to the data warehouse [5]. An ETL system consists of three functional steps:

Extraction: This step is responsible for extracting data from

the source systems. Each data source has its distinct set of characteristics that need to be managed in order to effectively extract data for the ETL process.

Transformation: This step tends to make the extracted

data to gain accurate data which is correct, complete, consistent, and unambiguous. This process includes data cleaning, transformation, and integration. It defines the granularity of fact tables, the dimension tables, data warehouse (star or snowflake), derived facts, and slowly changing dimensions.

Loading: In this loading step, extracted and transformed

data is written into the dimensional structures actually accessed by the end users and application systems. Loading step includes both loading dimension tables and fact tables.

In this data transformation step, we used Pentaho Data

Integration tool that is free open source data warehousing tool for academic and research purpose. This tool provides access to PostgreSQL database directly. In Fig.3, we extracted data from census data file (excel), the related data inserted into each of age, literacy, attendance, grade, employment and gender dimension tables and education fact table.

The education fact table is 10143 instances. The number of records of dimension tables will be shown in Data Description Section. By

Table 1 (a) Education Fact Table

age	literacy	attendance	grade	employment	gender
1	1	1	1	1	1
2	2	2	2	2	2
3	2	2	2	2	2
4	1	1	1	3	1
5	1	1	3	3	3
...
...

Table 2 (b) Grading

grade-key	grading
1	NG
2	none
3	other
4	1
5	2
6	3
7	4
8	5
9	6
10	7
11	8
12	9
13	10
14	11
15	college
16	vocationaltraining
17	undergraduate
18	graduate
19	master
20	Ph.d
21	postdoc

Table 3 (c) Age

age-key	age-range
1	0-10
2	11-20
3	21-30
4	31-40
5	41-50
6	51-60
7	61-70
8	71-80
9	81-90
10	91-100

applying this tool, the total transformation time of data processing is 10.8 msec.

Table 4 (d) Attendance

attend-key	attendance
1	previously
2	currently
3	NG

Table 5 (e) Employ

employ-key	employ
1	NG
2	other
3	student
4	gov-employer
5	org-employer
6	own-worker
7	family-worker
8	household-
9	unexper-worker
10	withexper-
11	general-worker
12	retired
13	disabled

Table 6 (f) Literacy

literacy-key	literacy
1	yes
2	NG
3	no

Table 7 (g) Gender

gender-key	gender
1	male
2	NG
3	female

shows the gender dimension table contains male, female and NULL they are not given.

C. Dataset Description

TABLE I. shows the database is designed in PostgreSQL9.5 database management system to store the collected data. In database, there are one education fact and six dimension tables: age, gender, literacy, attendance, grading and employment. The fact table relates data keys to each dimension. Each dimension tables contain detail information on persons responding to real-world dataset. The education fact table has 6 attributes and 10143 records (instance) in the dataset. For example TABLE IV: (d) shows the attendance dimension table contains previously, currently and NG they are not given. TABLE VII: (g)

4. Data Visualization

A. Visualization

Data are presented in a variety of formats and it is difficult to analyze when it is needed to use for decision making processes. Visualization is the best solution to produce graphical representations of data or concepts for decision making. Data visualization can provide large or complex data with a well-designed visual or graphics and can add value for underlying data. Large data are difficult to identify without the help of visuals because most people cannot interpret a table of data without time consuming analysis and tables of data cannot make a decision efficiently. Visuals can turn a data table into a graphic that can be quickly interpreted so it is easier to use evaluation findings. Visualization methods can visualize the innumerate amount of the analytical results as diagrams, tables and images. Visualization is the best solution to produce graphical representations of large amount of data or concepts for decision making. Visualization for Big Data differs from all of the previously traditional visualization techniques. A visual

can be either static or interactive. Static visuals are the most common since they are the simplest to produce. Interactive data visualization is a technique of analyzing

data, where a user interacts with the system that results in visual patterns for a given set of data.

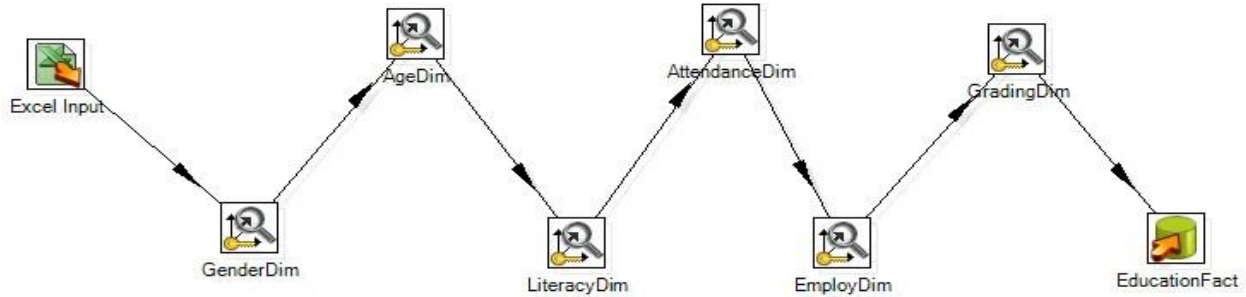


Fig. 3: Data Transformation Step

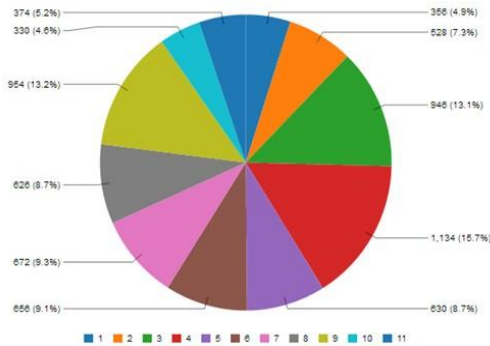


Fig. 4: Analysis for Grading with Male

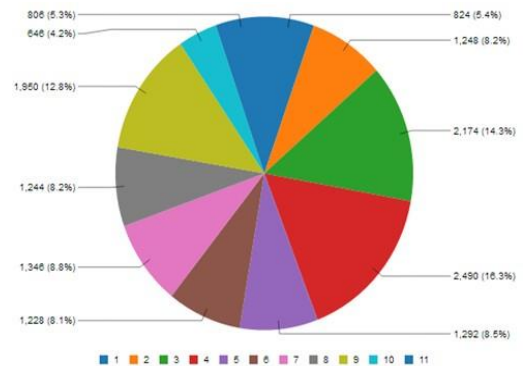


Fig. 5: Analysis for Grading with Male and Female

B. Visualizations for Census Data

Data visualization is using techniques that extract useful information. Visualization will make it easier to extract business intelligence. The ability to make timely decisions based on census data is crucial to decision making in national development. To display interactive visualization process, the visualization tools Pentaho Business Analytics (BA) Suite generated census data according to various attributes. This tool is a platform which can access, integrate, manipulate, visualize, and analyze your data. Whether data is stored in a flat file, relational database, Hadoop, NoSQL database, analytic database, social media streams, operational stores, or in the cloud, this

tool can discover, analyze, and visualize data. The Pentaho will create advanced visualizations of data and provide powerful insight of data.

C. Data Visualizations Results

Data visualization is the study of representing data in some systematic form, including attributes and variables for the unit of information. These data presentation should be beautiful, elegant, descriptive, and interpretable in order to convey message to the reader effectively. Data visualization represents data in the way that simplifies data interpretation and its relationship. The

following section is a summary of the data used in the data collection and creation of the visualizations for Myanmar census data.

In Fig.4, shows that the analysis of primary to high school students for Male. The children is age 4 year as kindergarten, they are attending at preschool, the percentage of preschool children are 5.4 and number of children is 824. The number of primary one children is 1248 and 8.2%. The number of primary two children is 2174 and 14.3%. The largest percentage is 16.3% and the number of primary three children is 2490. The number of primary four children is 1292 and 8.5%. The number of primary five children is 1228 and 8.1%. The number of primary six student in high school is 1346 and 8.8%. The number of primary seven student is 1244 and 8.2%. The number of primary eight student is 1950 and 12.8%. The number of primary nine student is 646 and 4.2%. The number of primary ten student is 806 and 5.3%, they are matriculation for attending university.

In Fig.5 describes that the analysis of male and female between primary to high school grading. The largest number of male and female is 2490 and the total percentage is 16.3 of primary three children. The smallest of male and female in primary nine students is 646 and 4.2%. In this paper, our visualization reports are not enough to leverage the benefits of business intelligence in such a dynamic industry.

5. Conclusion

This paper represented on a case study in which data warehouse tool has been applied to some data that is a portion of real census data. And then, it focused on the database model is star schema and converted data into the postgresQL database. The star join schema represents a relational database model

that facilitates the gathering of a great deal of this information and knowledge about the data, stores it, organizes it, and then relates it directly to the factual data being analyzed. Data-mining helps governments, individuals, companies to uncover hidden patterns in large database which is used for development and making decision from virtualization reports using census data. For future work, we have to present the needed and right information should also find a way to get to the right people in right time. We have to do these reports not only help to understand the past, but also work to find new opportunities and emerging trends in future.

6. References

- [1] M. Yost and J. Nealon, Using A Dimensional Data warehouse to standardize survey and Census Metadata.
- [2] N. Stolba and A.M. Tjoa, The relevance of data warehousing and data mining in the field of evidence based medicine to support healthcare decision making. European Social Fund (ESF), under grant 31.963/46- VII/9.
- [3] O.C. Okeke and B.C. Ekechukwu, Using Data-Mining Technique for Census Analysis to Give Geo-Spatial Distribution of Nigeria. IOSR Journal of Computer Engineering (IOSR-JCE), p-ISSN: 2278-872 Volume 14, Issue 2(Sep-Oct, 2013),PP01-05.
- [4] S.B. Jagtap and Kodge B.G, Census Data Mining and Data Analysis using WEKA. International Conference in "Emerging Trends in Science, Technology and Management-2013, Singapore.
- [5] S.H. Ali El-Sappagh, A.M. Ahmed Hendawi and A.H. El Bastawissy, A proposed model for data warehouse ETL processes, Journal of King Saud University- Computer and Information Sciences (2011) 23,91-104
- [6] W.Q. Qwaider, Apply On-Line Analytical Processing (OLAP) with Data Mining For Clinical Decision Support International Journal of Managing Information Technology (IJMIT), Vol.4, No.1, February 2012
- [7] M. Khan and S.S.Khanr, Data and Information Visualization Methods, and Interactive Mechanisms: A Survey, International Journal of Computer Applications (0975 - 8887), Volume 34- No.1, November 20.

Data Compression Strategy for Reference-Free Sequencing FASTQ Data

Hsu Mon Lei Aung, Swe Zin Hlaing
University of Information Technology, Yangon, Myanmar
hsumonleiaung@uit.edu.mm, swezin@uit.edu.mm

Abstract

Today, Next Generation Sequencing (NGS) technologies play a vital role for many research fields such as medicine, microbiology and agriculture, etc. The huge amount of these genomic sequencing data produced is growing exponentially. These data storages, processing and transmission becomes the most important challenges. Data compression seems to be a suitable solution to overcome these challenges. This paper proposes a lossless data compression strategy to process reference-free raw sequencing data in FASTQ format. The proposed system splits the input file into block files and creates a dynamic dictionary for reads. Afterwards, the transformed read sequences and dictionary are compressed by using appropriate lossless compression method. The performance of the proposed system was compared with existing state-of-art compression algorithms for three sample data sets. The proposed system provides up to 3% compression ratio of other compression algorithms.

Keywords- Genomic Sequencing data, lossless compression, reference-free sequence, reference-based sequence.

1. Introduction

The utilization of next generation sequencing data has greatly increased on genomics analysis, hereditary disease diagnosis, therapy and food security, etc. The rate of storage space capacity, processing and data transferring of large genomic sequencing data are rapidly grown up. It becomes a challenge on the storage of sequencing data. Without any efficient compression, the storage and transmit size of sequences will be large.

The compression of genomic sequences is divided into two main categories: reference-free compression methods and reference-based methods [6]. The main idea of reference-free sequence compression is exploiting the structural and statistical properties of data and storing the sequencing reads with specific compressive encoding schema.

Reference-based compression is exploiting the similarity between a target sequence and a reference sequence. The target is aligned (mapped) to the reference.

The reference-based methods do not encode the original read data but the mismatches between these sequences are encoded.

Lossless data compression is used when the original data source files are so important that cannot accept any loss in details. The popular general-purpose compression algorithms are bzip2 and gzip. The bzip2 is an open-source file compression program that uses the standard Burrows–Wheeler (BWT) algorithm. It performs block-based sorting for text transformation. The transformation is reversible, without needing to store any additional data [5]. The gzip is based on deflate algorithm, which is a combination of LZ77 and Huffman’s coding. However, the general-purpose algorithms cannot compress the biological sequencing data well because they are not taken into account the characteristics of DNA data. The dictionary-based method [3] is substituting the words by indices that relate to the dictionary of words [8].

Although several specific sequencing data compression methods have been proposed, they have many trade-off of the compression performance parameters and some methods have limitation about type of genome sequences. And the availability of assembled reference sequences for all the organisms are difficult.

There are many different file formats, such as FASTA, Multi-FASTA, FASTQ and SAM/BAM to store biological sequencing data. FASTQ, modified version of traditional FASTA file, is the defacto standard for storing data from next generation sequencing platform. It is a text-based format which represents biological sequences and their corresponding quality scores with ASCII characters. This information is stored in the form of read blocks. Example of a read block is shown in Figure 1.

```
@SRR1063349.15409 15409 length=202
TCGCGCTGCAACCTTATGACGTGGTGTATGTACCACC GCCCCGGTTCCCG
CTGGAACCGTCTGATCAATCAGTTGCTGCCAATTAGCGGTGTTGCACGGC
CCGACGCGTACTGGTAATTCGCGATCTCACTTCCAGCGTCATATTGGGGCC
CGCTACGTTGGGTCGTGGCGTAATTGATCAATCAGGCGCGGGG
+SRR1063349.15409 15409 length=202
CCCCFFFFFFHHHHJJJEHHIJGHECGHIIJJJJJJJJJJBBBEHGHFFDCEDDDD
DDDDDDCCCD>C4>ACDCDDDDDDCECEDD5>5<<>4@@@FFFFFFHHGH
HJJJJEDHIIJJJJJJGJJJJHCHHF?CBECF>ACDDDBDDDDDBD<8?BBD?AB@
BB>BD>BBDEDEDEDCCDDDD@BB
```

Figure 1. Example of a read block in FASTQ

A read block contains four parts. These are sequence identifier, raw read sequence, description field (optional) and quality scores. A description field is also called second header line that starts with "+" .The information is the same as the identifier, but it can also be blank. So, this field can be omitted.

The remainder of the paper is organized as follows: Section II reviews the methods proposed for the compression of biological sequences in FASTQ file format. Section III describes the proposed method in detail. Section IV presents the performance results for the proposed method. Finally, Section V describes conclusions and directions for further work.

2. Related Work

Fqzcomp and Fastqz methods [1] are proposed to compress the files in FASTQ format. The first method uses a byte-wise arithmetic coder and context models. The second method breaks the input file into three separate streams and uses libzpaq compression library for specifying the context models in ZPAQ format based on PAQ, which combines the bit-wise predictions of several context models. The performance trade-off of two methods depends on the use case.

The XM (eXpert Model) method [2] compresses each symbol by estimating the probability distribution based on previous symbols information. After the symbol's probability distribution is determined, it is encoded using arithmetic coding. The method maintains three types of expert models such as Markov expert, Context Markov expert and Repeat expert that can provide a probability distribution of symbols and considers the next symbols. The algorithm's compression speed is very slow when it is applied on large genome data sets.

In [4], the DNACompact algorithm is encoded by word-based tagged code. It is a two phases-compression method. The first phase is transformation that transformed four symbol space into a three symbol space. The second phrase is encoding scheme that encoded by WBTC.

The DNA-COMPACT [7] is a two-pass lossless DNA compression based on a pattern-aware contextual modeling technique.

In the first pass, there are two schemes for with and without reference sequences. In the second pass, non-sequential contextual models are used. For the non-aligned sequences, this method is not suitable to get good compression performance.

In [9], FASTQ compression method based on concept of prediction by partial matching (PPM) is proposed. This method develops context based models customized to each FASTQ field. The models are coupled with an adaptive arithmetic coder (AAC) to compress data. The proposed method gains in overall storage space on a sample dataset. The limitation of the proposed method is needed to get faster performance.

DMcompress method [10] firstly analyzes the raw sequence data by calculating first order entropy and then determines the Markov model orders according to the Laplace estimator. Finally, the data is compressed by arithmetic coding. It can get effective compression performance for the latest bacteria genome sequences with around 0.02 bps reduction.

Lossless light-weight reference-based compression algorithm, LW-FQZip [11], identifies and eliminates redundancy information independently from other three components of a FASTQ file. Then incremental method is applied for identifier and run-length limited encoding method is used to encode the quality scores. And then, a light-weight mapping model is used which maps them against external reference sequences to encode short reads. Finally, all the processed data streams are packed together by applying LZMA algorithm. LW-FQZip can get optimal compression ratio, but its processing time and memory utilization are very high.

In this paper, a new lossless compression strategy of biological sequences without reference sequences for FASTQ data is proposed. The proposed method can create dynamic dictionary and preprocessed sequences by finding matching symbols in sequence itself. For unmatched symbols, the appropriate compression algorithm is applied. The compressed files are decompressed and transformed into original files by using reversible transformation and compression.

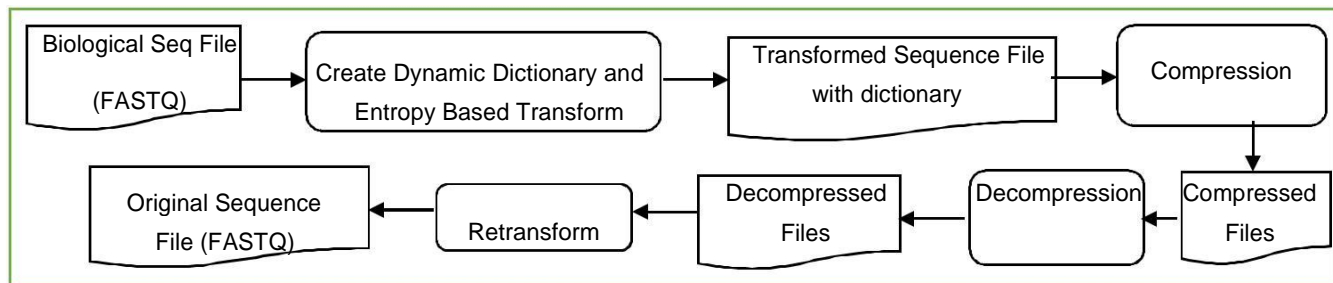


Figure 2. Overview of proposed system design

3. Proposed System

The overview design of the proposed system is shown in Figure 2. Firstly, input file is transformed and created dictionary based on entropy calculation. Secondly, the transformed sequence is compressed by BWCA that is a combination of BWT with compression techniques Move-To-Front transform (MTF), Run-Length Encoding (RLE) and the Huffman Coding.

The flow of a compression process is shown in Figure 3. Firstly, the proposed system splits the FASTQ file into two files. These files will process in parallel. Each read block contains four lines. The third line can be omitted for compression. The three lines (Sequence Identifier, Read Sequences, Quality Score) can be compressed.

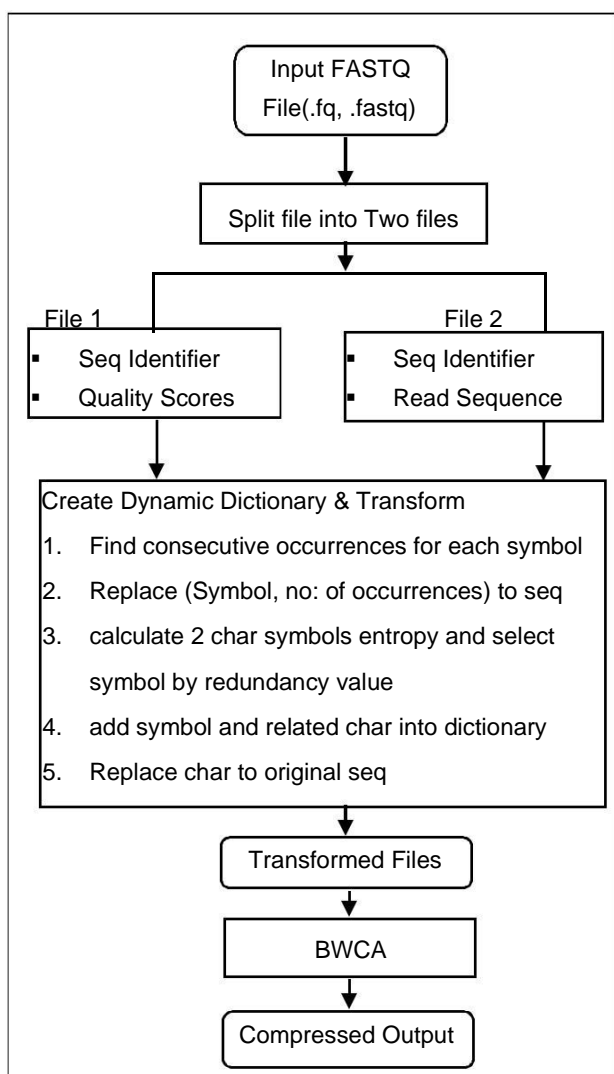


Figure 3. Flow diagram of compression process

Secondly, the dictionary and transformed sequences are created. In step (1) and (2), the system counts the

consecutive occurrences for each symbol that include one character and replaces the (symbols, occurrences) pair into original sequence. Step (3) calculates entropy of symbols including two characters. The entropy calculation equations are shown as follows:

$$H(P) = - \sum_{j=1}^n p_j \log_2 p_j \quad (1)$$

In equation 1, $H(P)$ is the smallest number of bits required to represent the symbols in file. (P_j) represents the probability of each symbol.

The proposed system calculates redundancy by using equation (2). $H_{max}(P)$ is the maximum entropy for same number of states.

$$R = \frac{F}{H_m F} = 1 - H(P) \quad (2)$$

According to the experiments, the numbers of permutations of four symbols taken 2 at a time always have highest redundancy. So, the proposed system, firstly creates two characters symbols dictionary. And then calculate other symbols redundancy. If the value is zero or less than 0.6, it can be added that symbol to dictionary. In step (5), the system replaces the symbols into sequences with char from dictionary. Next the transformed sequences are compressed by using BWCA.

The decompression is the reverse process of compression. In decompression part, the compressed file is firstly decompressed using BWCA and then retransformed it into original file using dictionary.

4. Experimental Result

In this section, two real-world FASTQ data sets and one sample data set are used to test the performance of the proposed system. All file sizes are shown by MB.

Table 1 shows the original file sizes and compressed file sizes of the proposed method that combines with lossless data compression algorithms and popular compression tools such as bzip2 and zip.

Table 1. Comparison of compressed files size

Data Sets	File Size (MB)	bzip2 (MB)	zip (MB)	Proposed +BWCA (MB)	Proposed +Arithmetic Coding (MB)
SRR445780_1.fastq	29.3	5.19	5.94	4.7	10.5
ERR016352_1.fastq	37.6	12.5	14.4	11.2	16.3
SRR1063349.fastq	30	7	8	6.2	11.4

In Figure 4, the average compression ratios of four methods are compared. The combination of proposed method and BWCA gets the higher ratio than other methods.

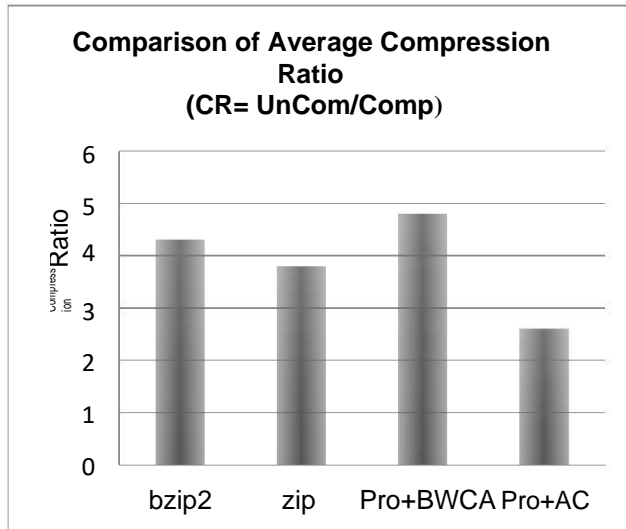


Figure 4. Comparison of average compression ratio

Figure 5 shows the average space saving percentage of four methods. The method (proposed plus BWCA) can save 78% of space, and bzip2 can get 75%. So, the combination of proposed system and BWCA can provide more file compression results.

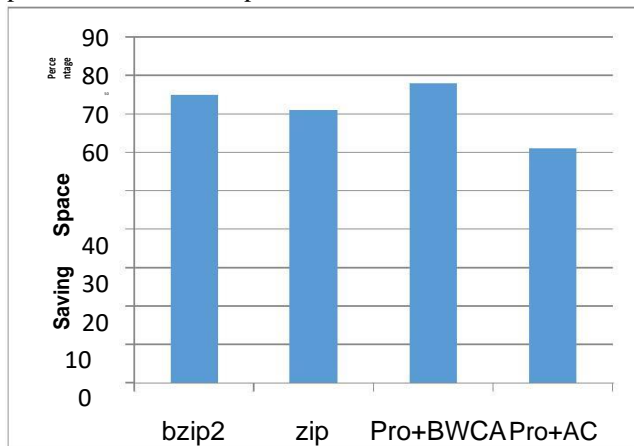


Figure 5. Average Space Saving Percentage of four methods

5. Conclusion

There are many challenges in biological sequence processing and compression. Many different data compression methods have advantages and disadvantages. Some are time-consuming to process but accurate; others are simple to compute but less

powerful. Transformation can get better data form before data compression. The proposed system will provide the efficient compression ratio for biological data sequences in FASTQ file format. In future, the proposed method will modify the performance in other sequence file formats. And it also will improve the processing time performance of the proposed system.

6. References

- [1] Bonfield, J.; Mahoney M, "Compression of FASTQ and SAM format sequencing data", PLoS ONE, 2013.
 - [2] Cao, M.; Dix, T.; Allison, L.; Mears, C, "A simple statistical algorithm for biological sequence compression", Data Compression Conference, Snowbird, UT, USA, 2007.
 - [3] D. Bhattacharya, S. Chakraborty, P. Roy, A. Kairi, "An Advanced Dictionary Based Lossless Compression Technique for English Text Data", CIIT, 2015.
 - [4] Gupta, A.; Agarwal, S, "A novel approach for compressing DNA sequences using semi-statistical compressor", International Journal of Computers and Applications, 2011.
 - [5] Hsu Mon Lei Aung, Aye Sandar Win, Swe Zin Hlaing, "Data Transformation for Textual Unstructured Data Compression", ICCA, 2017.
 - [6] M. Hosseini, D. Pratas and Armando J. Pinho, "A Survey on Data Compression Methods for Biological Sequences", Information, 2016.
 - [7] P. Li, S.Wang, J. Kim, H. Xiong, L. Ohno-Machado, and X. Jiang, "DNA-COMPACT:DNA Compression Based on a Pattern-Aware Contextual Modeling Technique," PlosOne, 2013.
 - [8] R.R.Baruah, V.Deka, M.P. Bhuyan, "Enhancing Dictionary Based Preprocessing for Better Text Compression", IJCTT, 2014.
- A. R. Srikanth Mallavarapu, P. Kumar Chinnamalliah., and Ajit S. Bopardikar, "Context based compression of FASTQ data", IEEE, ISCAS, 2016.
- B. R. Wang, M. Teng, Y. Bai, "DMcompress: dynamic Markov models for bacterial genome compression", Bioinformatics and Biomedicine BIBM, IEEE, 2016.
- C. Y. Zhang, L. Li, Y. Yang, X. Yang, S. He and Z. Zhu, "Light -weight reference-based compression of FASTQ data", BMC Bioinform. 2015.

Natural Language Processing

Domain-specific Sentiment Dictionary Construction for Sentiment Classification

Aye Aye Mar, Nyein Thwet Thwet Aung, Su Su Htay

Faculty of Information Science, University of Information Technology, Myanmar
ayeayemar, suhtay, nyeinthwet}@uit.edu.mm

Abstract

Sentiment dictionaries are commonly used to solve the problem of sentiment classification for customer reviews. The number of sentiment words in the generalized dictionaries such as SentiWordNet is limited and lack of many sentiment words especially domain-specific sentiment words. Different domains have different sentiment words and the sentiment of a word depends on the domain in which it is used. In this paper, an approach based on Point-wise Mutual Information (PMI) is proposed to construct a domain-specific sentiment dictionary effectively and automatically. The proposed system is evaluated on three diverse datasets from different domains by using 10-fold cross validation. Accordingly to the experimental results, the goodness of the extracted dictionary is relatively high and significantly improves the performance of sentiment classification. The experimental results show that the extracted domain-specific dictionary outperforms the generalized dictionary, SentiWordNet. The proposed method learns the domain-specific sentiment words efficiently and it is domain adaptable.

Keywords- Sentiment Analysis, Polarity Classification, Sentiment Dictionary, Domain-specific Sentiment Words, Point-wise Mutual Information

1. Introduction

The amount of user generated data on the web is increasing more and more during the last few years. As a consequence of this, sentiment analysis from these data has become a prominent research area. Sentiment analysis is a kind of text mining combined with the natural language processing and computational linguistics. It is the task of extracting valuable information from a collection of documents containing opinions, feelings and attitudes.

Sentiment analysis is applied in three levels of granularity, which are document level, sentence level and aspect level also called feature level. The key factor of all these levels is to identify the polarities of the sentiment words. The polarities of some sentiment words vary based on the domain in which it is used. “Unpredictable” may have a negative sentiment in a car review as in “unpredictable steering,” but it could have a positive sentiment in a movie review as in

“unpredictable plot” [1]. Most of the existing sentiment dictionaries specify the polarity of such words generally instead of considering the polarity of these words for each specific domain. Therefore, the methods that are able to construct domain-specific sentiment dictionaries are essential for an accurate sentiment classification system.

This paper proposes an approach for constructing a domain-specific dictionary by using labelled review datasets as the training. The approach considers the probability distribution of the sentiment words with the class labels which is computed by Point-wise Mutual Information (PMI). The proposed method solves the following three main problems of sentiment analysis: (1) the need of human effort to construct a domain-specific dictionary manually (2) missing the polarities of domain-specific sentiment words when the generalized sentiment dictionaries are lack of them (3) the problem to identify the right polarities for domain dependent sentiment words.

The rest of the paper is organized as follows: Section 2 summarizes the related work. The proposed system is presented in Section 3. In Section 4, the experimental evaluations are described. Section 5 concludes the paper and the future work of the proposed method is presented in Section 6.

2. Related Work

There are three common approaches for generating sentiments of words: manual, dictionary-based and corpus-based approaches. The manual approach simply uses human knowledge to decide the sentiment of a word. Meanwhile, dictionary based and corpus-based approaches automatically generate sentiments of words using dictionaries and corpuses respectively [2]. Dictionary-based approach is one of the main approaches to extract sentiment words in sentiment analysis [1]. Due to the importance of sentiment words within sentences, many approaches have been proposed to predefine the polarity of sentiment words.

A manually built lexicon has been used to classify the text as the positive or negative. In [3], Hatzivassiloglou and Wiebe assumed that adjectives are the clues to trace the sentiment orientation of a given text. Based on the manually created lexicon for adjectives and their semantic orientation values (SO),

for any given text, all adjectives are extracted and associated with their dictionary SO values. The overall sentiment score is obtained by summing up all adjective SO scores within the given text. The given text is then classified as bearing a positive or negative sentiment based on the overall score for the obtained adjectives within it.

The need of creating new hand-built lexicons for the new domains is very labor intensive. In order to create a sentiment dictionary efficiently and escape any manual effort, Turney [4] proposed a simple promising approach to create a sentiment dictionary in an automatic way. The dictionary was built by using the positive and negative seed words. In order to find the correlation between a seed word and the target word, the author used mutual information approach which is based on statistical data extracted from the web with the aid of AltaVista search engine. The target word is passed as a query to the search engine i.e., either with the word “excellent” or the word “poor”. The semantic orientation is then acquired based on the mutual information between the target word with the word “excellent” or with the word “poor”. If the attained mutual information score for the target word with the word “excellent” is greater than the one with the word “poor”, the target word will be classified as positive, otherwise it will be classified as negative.

In [5], Hu and Liu claimed that the created dictionary list can be further expanded by utilizing synonym and antonym sets in WordNet [6]. The polarities of groups of synonyms are assumed to be similar e.g., “beautiful” and “pretty” while the polarities of antonyms are supposed to be opposite e.g., “excited” and “bored”. However, Leung et al., argue that 580 semantic similarities do not necessarily employ sentimental similarity accordingly to statistical evidence obtained from movie review data [7].

Manually created dictionary would be convenient to detect a sentiment only for a given domain. Therefore, researchers created publicly available lexical resources such as SentiWordNet. SentiWordNet is a lexical resource of sentiment information for terms in English language designed to assist in opinion mining tasks. Each term in SentiWordNet is associated with numerical scores for positive and negative sentiment information [9] [10]. SentiWordNet can be used for sentiment analysis of all domains. However, the number of terms defined in SentiWordNet is limited [2].

Different domains have different kinds of sentiment words. Although sentiment words can be the common words among different domains, all of the sentiment words cannot be the same. Even though the same sentiment word is contained in the sentiment words lists of many domains, its polarity will be changed depending on the domain that is associated with. A positive or negative sentiment word may have opposite orientations

in different application domains. For example, “suck” usually indicates negative sentiment, e.g., “This camera sucks,” but it can also imply positive sentiment, e.g., “This vacuum cleaner really sucks.” [1].

An automatic approach for constructing domain-specific sentiment dictionary is necessary for improving the sentiment classification. The number of terms in SentiWordNet is limited and usually lack of many sentiment words, especially domain specific sentiment words [2]. This is the motivation of our research to propose an approach for building the domain-specific dictionary by using a labelled training dataset instead of using generalized SentiWordNet dictionary for sentiment classification.

The main goal of the proposed method is to seek the relevant sentiment words for a given domain effectively and automatically. The system is also aimed to develop robust classification approach of customer reviews based on domain-specific labelled training datasets by applying statistical approach. Moreover, the system analyses the performance of Point-wise Mutual Information (PMI) method by using different review datasets from different domains.

3. Proposed System

The proposed system is mainly composed of three components: preprocessing, constructing sentiment dictionary and classifying the reviews by utilizing the extracted dictionary. Firstly, all of the review datasets are preprocessed. Secondly, domain-specific dictionary is constructed by computing sentiment orientation of these subjective words based on Point-wise Mutual Information (PMI). Finally, review documents are classified by applying the extracted domain-specific sentiment dictionary. Figure 1 shows the system flow of the proposed system.

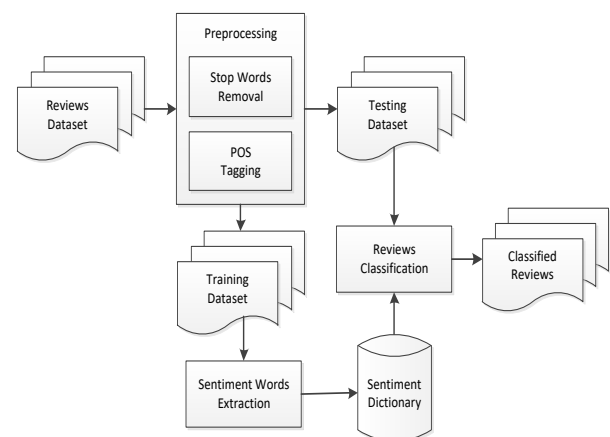


Figure 1. System flow of the proposed system

3.1 Preprocessing

Preprocessing is necessary before extracting the sentiment features. It includes two parts: POS tagging and stop words removal. First and foremost, POS tagging is done by applying the Stanford POS tagger tool¹. POS tagging means labelling each word in a sentence with its appropriate part of speech such as noun, adjective, adverb etc.

Stop words such as verb to be, pronouns, prepositions and conjunctions do not give meaningful information for sentiment analysis. So, the stop words² are removed to save the processing time.

3.2 Constructing Sentiment Dictionary

Most of the previous works consider only adjectives as the sentiment words. In similar to adjectives, adverbs and verbs also describe sentiments as the adjectives. The experimental results shows that that polarity classification is more accurate by considering the polarities of adjective, adverb and verb instead of adjective alone. Therefore, not only adjective but also adverb and verb are considered as the sentiment words in this system. The sentiment dictionary is constructed by using the Algorithm 1.

The positive and negative sentiment scores of each sentiment word are computed based on the Point-wise Mutual Information (PMI) [11]. If a word is occurred frequently and predominantly in one class (positive or negative), then that word would have high polarity. If the positive PMI score of a sentiment word is greater than its negative PMI score, it indicates that the word has occurred mostly in positive documents. Alternatively, it indicates that the word has occurred mostly in negative documents if the negative PMI score of a word is greater than its positive PMI score. Point-wise Mutual Information (PMI) is used to calculate the strength of association between a word and positive or negative documents in sentiment analysis. The positive PMI-Score and negative PMI-Score of each sentiment word w is computed by Eq. (1) and by Eq. (2) respectively. In this system, both the positive PMI-Score and negative PMI-Score of each sentiment word are taken into account for computing the polarity of the review document.

$$PMI(w, Positive) = \log_2 \frac{P(w, Positive)}{P(w)P(Positive)} \quad (1)$$

$$PMI(w, Negative) = \log_2 \frac{P(w, Negative)}{P(w)P(Negative)} \quad (2)$$

¹ <https://nlp.stanford.edu/software/tagger.shtml>

² <http://xpo6.com/list-of-english-stop-words/>

Where, $P(w, Positive)$ is the joint probability of co-occurrence of sentiment word w found together with the class Positive and $P(w)$ and $P(Positive)$ are the probability of occurrence of sentiment word w and class Positive independently.

After computing the positive PMI-Score and negative PMI-Score for each sentiment word, the system constructs the sentiment dictionary which includes the sentiment words together with their respective POS tag (Part of Speech tag), positive PMI-Score and negative PMI-Score.

Algorithm 1. Algorithm for constructing sentiment dictionary

Input : A given training reviews set S with the POS tagged words $\leftarrow \{ s_1(w_1, w_2, \dots, w_n: label), \dots, s_n(w_1, w_2, \dots, w_n: label) \}$

Output: sentiment dictionary which contains sentiment words together with their respective POS tag, positive score and negative score

counts the occurrence frequency of each word $w \in s \in S$

```

1: for each training review  $s \in S$  do
2:   for each word  $w \in s$  do
3:     if ( $w$  is adjective or adverb or verb) then
4:       if ( $w$  is not in sentiment word list  $L$ ) then
5:          $L \leftarrow w$ 
6:       end if
7:       if (label of  $s$  is positive) then
8:         increase one to  $pos\_count$  of  $s$ 
9:       else (label of  $s$  is negative)
10:        increase one to  $neg\_count$  of  $s$ 
11:       end if
12:     end if
13:   end for
14: end for
15: create the frequency table by using  $pos\_count$  and  $neg\_count$  of each  $w \in L$ 
16: compute the probability table from the frequency table
    # compute positive PMI-Score and negative PMI-Score for each  $w \in L$ 
17: for each  $w \in L$ 
18:    $pos\_score \leftarrow PMI(w, Positive)$  by Eq.1
19:    $neg\_score \leftarrow PMI(w, Negative)$  by Eq.2
20: save  $w$  into sentiment dictionary  $D$  together with  $POS$  tag,  $pos\_score$ ,  $neg\_score$ 
21: end for

```

Algorithm 2 classifies the different testing datasets by utilizing the domain-specific sentiment dictionary extracted from the respective training review datasets.

Algorithm 2. Algorithm for classifying reviews documents by using the extracted domain-specific dictionary

Input : testing reviews set T with the POS tagged words $\leftarrow \{t_1 (w_1, w_2, \dots, w_n), \dots, t_n (w_1, w_2, \dots, w_n)\}$, sentiment dictionary D

Output : testing reviews set T with assigned labels

classify the testing reviews set by using the extracted sentiment dictionary

```

1: for each testing review  $t \in T$ 
2:   for each  $w \in t$ 
3:      $total\_pos\_score \leftarrow pos\_score$  of  $w$  in  $D$ 
4:      $total\_neg\_score \leftarrow neg\_score$  of  $w$  in  $D$ 
5:   end for
6:   if ( $total\_pos\_score > total\_neg\_score$  ) then
7:      $t \leftarrow$  Positive
8:   end if
9:   if ( $total\_neg\_score > total\_pos\_score$  ) then
10:     $t \leftarrow$  Negative
11:   end if
12:   if ( $total\_neg\_score == total\_pos\_score$  ) then
13:     $t \leftarrow$  Neutral
14:   end if
15: end for

```

4. Experiment Evaluation

4.1 Dataset Description

The publicly available three diverse review datasets are used to evaluate the domain adaptability of the proposed method. These three domains are movie, product and hotel. We used publicly available Cornell movie review dataset³ of Peng and Lee for movie domain. Some of the hotel reviews from tripadvisor are taken for hotel review dataset⁴. For product domain, the beauty product reviews⁵ of amazon product review datasets are used. All of the three datasets contain 50% positive reviews and 50% negative reviews to maintain the class distribution. The description of these three diverse datasets is shown in Table 1.

³ <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

⁴ <http://sifaka.cs.uiuc.edu/~wang296/Data/index.html>

⁵ http://www.ilabsite.org/?page_id=1091

Table 1. Dataset description

Domain	Positive	Negative	Total
Movie	1000	1000	2000
Product	4500	4500	9000
Hotel	12500	12500	25000
Total	18000	18000	36000

4.2 Preparing Training and Testing Data

In our evaluation, splitting the datasets into training and testing involves the k-fold cross validation method [8]. In k-fold cross validation method, the data is split into k folds where k-1 folds is used for training the algorithm and the remaining one fold is used for testing the algorithm. The final measure of performance takes the average of the results of all folds. In this work, we used 10-fold cross validation to make robust evaluation.

4.3 Evaluation Metrics

Five evaluation metrics, which are precision, recall, F-measure, accuracy and failure-ratio, are used to evaluate the effectiveness of the system. These are calculated by using Eq. (3)-(7) respectively.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F - measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (5)$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

$$Failure - Ratio = \frac{NumberOfMisclassified\ Reviews}{TotalNumberOf\ Reviews} \quad (7)$$

Where:

TP refers to the number of true positive reviews.

TN refers to the number of true negative reviews.

FP refers to the number of false positive reviews.

FN refers to the number of false negative reviews.

Number of Misclassified Reviews refers to the reviews labelled to the class label which was not included in the actual class labels.

Total Number of Reviews refers to the number of all reviews.

4.4 Experimental Results

This section analyses the experimental results of the proposed method and the baseline SentiWordNet dictionary on three diverse datasets. Table 2 shows the evaluation results of baseline SentiWordNet and the proposed method by using the movie, hotel and product review datasets. Among the five performance measures, the proposed method has significant results than the baseline in precision, F-measure and accuracy.

A few explanations concerned with the failure-ratio should be made here. The failure-ratio in Table 2 means that the error that is occurred when a review document is labelled with the class labels that is not really present. As shown in Table 1, the datasets contain only two class labels, positive and negative. There were no neutral review documents in the datasets. However, both the baseline method SentiWordNet and the proposed method make classification to a few documents as the neutral documents (which is not present in the actual class labels). Labelling the documents as the neutral is happened when the total positive score is equal to total negative score of the review document.

Table 2. Experimental results in % of SentiWordNet (SN) and the proposed method (PM)

Dataset	Product Dataset		Movie Dataset		Hotel Dataset	
	SN	PM	SN	PM	SN	PM
Precision	58.05	85.46	57.75	78.16	76.75	81.69
Recall	88.11	84.55	88.15	76.64	96.86	92.54
F-measure	69.94	85.00	69.67	77.28	85.63	86.76
Accuracy	62.25	85.20	62.05	77.58	77.10	85.71
Failure-Ratio	0.18	0.6	0	0.12	0.33	0.54

For product review dataset, the experimental results show that proposed method (PM) has significant high results than the baseline SentiWordNet in precision, F-measure and accuracy except low result in recall. In the proposed model, precision is improved dramatically by 27.41%, the F-measure is increased significantly by 15.06 % and the accuracy is improved inevitably by 22.95% except the decline of 3.56% in recall. Both the baseline method and the proposed method have failure-ratio of 0.18% and 0.6% respectively.

As in the product domain, the performance of proposed method in movie domain has a visible improvement in precision, F-measure and accuracy with the increment of 20.41%, 7.61% and 15.53%

respectively. The recall of the proposed method is decreased by 11.51%. In the view point of failure-ratio, the baseline method is failure free in this domain although the proposed method has a slight failure-ratio of 0.12%.

In hotel domain, both the methods have a pretty good evaluation results and the proposed method is improved in terms of precision, F-measure and accuracy. The proposed method has higher precision by 4.94 %, increased F-measure by 1.13% and raised accuracy by 8.61% than the baseline except the low recall by 4.32%.

To sum up about the comparative results of the baseline method and the proposed method, the proposed method is improved dramatically although it has low recall and a slight failure-ratio than the baseline.

5. Conclusion

This paper proposed an approach to solve the problem of automatic construction of domain-specific sentiment dictionary. The extracted dictionary is evaluated by using 10-fold cross validation to be robust evaluation of the system. The experimental results demonstrate that the proposed method efficiently learns domain-specific sentiment words. The precision, F-measure and accuracy of the proposed system have a significant result than the baseline generalized dictionary, SentiWordNet.

6. Future Work

As the future work of the system, negation case will be considered to improve the performance of the system. Currently, the system is able to analyze sentiments in document level. To get the sentiments of customers in more details, aspect level sentiment analysis will have to be done. The sentiment targets of the sentiment words will be detected in order to make the right decision what the customers like and dislike.

7. References

- [1] B. Liu, "Sentiment analysis and opinion mining." Synthesis lectures on human language technologies 5.1, 2012, pp. 1-167.
- [2] N.H.Nguyen, T.V.Le, H.S.Le, T.V.Pharm, "Domain specific sentiment dictionary for opinion mining of vietnamese text." International Workshop on Multi-disciplinary Trends in Artificial Intelligence. Springer, Cham, 2014.
- [3] V.Hatzivassiloglou, J.M.Wiebe, "Effects of adjective orientation and gradability on sentence subjectivity", In Proceedings of the 18th conference on Computational linguistics-Volume 1, Association for Computational Linguistics, 2002,pp. 299-305.

- [4] P.D.Turney, "Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews." In Proceedings of the 40th annual meeting on association for computational linguistics,. Association for Computational Linguistics, 2002, pp. 417-424.
- [5] M.Hu, B.Liu. "Mining and summarizing customer reviews." In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 168-177.
- [6] G.A.Miller. "WordNet: a lexical database for English." Communications of the ACM 38.11 (1995): 39-41.
- [7] C.W.Leung, S.C.Chan, F.Chung. "Integrating collaborative filtering and sentiment analysis: A rating inference approach." Proceedings of the ECAI 2006 workshop on recommender systems. 2006, pp. 62-66.
- [8] R.Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection." In *Ijcai*, 1995,vol. 14, no. 2, pp. 1137-1145.
- [9] S.Baccianella, A.Esuli, and S].Fabrizio "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." In LREC, vol. 10, 2010, pp. 2200-2204.
- [10] A.Esuli , F.Sebastiani, "SENTIWORDNET:A publicly available lexical resource for opinion mining", In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06), 2006, pp. 417-422, Genova, IT.
- [11] K.W.Church, and H.Patrick "Word association norms, mutual information, and lexicography." Computational linguistics 16, no. 1, 1990, pp. 22-29.

Domain-Specific Sentiment Lexicon for Classification

Thet Thet Zin, Kay Thi Yar, Su Su Htay, Khine Khine Htwe,
Nyein Thwet Thwet Aung, Win Win Thant

University of Information Technology, Yangon, Myanmar

*thetthetzin@uit.edu.mm, kaythiyar@uit.edu.mm, suhtay@uit.edu.mm, khinekhine@uit.edu.mm
nyeinthwet@uit.edu.mm, winwinthant@uit.edu.mm*

Abstract

Nowadays people express their opinions about products, government policies, schemes and programs over social media sites using web or mobile. At the present time, in our country, government changes policies in every sector and people follow with the eyes or the mind on these policies and express their opinion by writing comments on social media especially using Facebook news media pages. Therefore, our research group intends to do sentiment analysis on new articles. Domain-specific sentiment lexicon has played an important role in opinion mining system. Due to the ubiquitous domain diversity and absence of domain-specific prior knowledge, construction of domain-specific lexicon has become a challenging research topic in recent year. In this paper, lexicon construction for sentiment analysis is described. In this work, there are two main steps: (1) pre-processing on raw data comments that are extracted from Facebook news media pages and (2) constructing lexicon for coming classification work. The word correlation and chi-square statistic are applied to construct lexicon as desired. Experimental results on comments datasets demonstrate that proposed approach is suitable for construction the domain-specific lexicon.

Keywords- social media, sentiment analysis, lexicon

1. Introduction

The World Wide Web and the online media provide a forum through which an individual's process of decision making may be influenced by the opinions of others. Online sites such as rottentomatoes.com, allow movie lovers to leave reviews for movies they have seen. Online sites, such as Facebook and blogs, allow users to leave opinions and comments. Other online sites, such as cnn.com and globeandmail.com, allow readers to leave comments. These kinds of online media have resulted in large quantities of textual data containing opinion and facts. Researchers have long measured people's opinion using carefully designed survey questions, which are given to a small number of volunteers. The maturation of social media offers alternative measurement approaches. Social media, now used regularly by more than 1 billion

of the world's 7 billion people, contains billions of such communications [1]. Researchers have begun providing these data for a wide range of applications including predicting the stock market and understanding sentiment about products or people. At the present time, Myanmar people express their opinion, feeling and daily activities using Facebook social network by updating status or writing comments on posts using Myanmar language. Therefore, our research group intended to mark people's opinion on important news articles. This is one of the supporting facts for the success of government policies. Sentiment analysis on news articles, especially 21st century Panglong Conference, is ongoing research. In this paper, lexicon construction for sentiment analysis using people comments on news articles is presented.

Feature extraction from data resources is an important step and essential process in sentiment analysis. There are several approaches to extract features from sentences: extraction based on frequency count, n-grams, extraction by exploiting opinion and target relations, extraction using supervised or unsupervised learning, and using topic modeling. According to class information availability in data, there are supervised feature selection approaches [2] as well as unsupervised feature selection approaches. [3] Extracted unigrams, bigrams and combination of unigrams and bigrams is this three different feature vectors applied to different classifiers. In most of the research work, unigram and bigram are highly recommended for tweeter sentiment analysis. In [4] Popescu and Etzioni introduced an unsupervised information extraction system which mines reviews in order to build a model of important product features, the evaluation by reviewers and the relative quality across products. In [5], researchers proposed within class popularity (WCP) feature selection mechanism and then the performance of WCP is then compared with the performance of the most commonly used measures-mutual information, information gain and chi-square. Most of the current approaches study the adaptation or sentiment transfer learning of a trained classifier supervised techniques or lexicon unsupervised techniques from one domain to another which involves having a general lexicon to start with, but very few works actually focus on techniques that build specific domain lexicons without requiring a priori knowledge. Whilst supervised

sentiment classifier performs very well for the domain in which they were trained for, they usually perform very poorly when adapted or transferred to another domain [6].

In this paper, the focus is on feature selection by construction lexicon using n-gram, word correlation and chi-square statistic methods. These feature selection methods are applied to Facebook users' comments on news media pages written in Myanmar language. Myanmar texts are sequence of characters without word boundaries. Therefore, user comments need to be parsed and tokenized into individual words first. In this work, Myanmar syllable segmenter for Unicode Myanmar is used for syllable segmentation. The paper is organized as follows. In the next section data collection process is briefly described. Section 3 contains pre-processing on collected unstructured data. Section 4 provides feature filtering and selecting by applying word correlation and chi-square method. Lexicon Coverage Analysis and Experiments and Results are described in section 5 and 6 respectively. The final section contains a discussion of the obtained results, some remarks and issues that remain to be addressed and that we intend to investigate in future work.

2. Extract Users' comments from Facebook

The internet is a resourceful place with respect to sentiment information. From a user's view, people are able to post their own content through various social media. From a researcher's view, many social media sites release their API, allowing data collection and analysis by researchers and developers. Facebook also releases API for data collection. But it has privacy issues for the researcher. However, news media pages focus on public and not intended to personally. According to my knowledge, news media pages post their articles with public setting. Thus, privacy is not issue for this work. In this paper, the comments are obtained from the Facebook news pages using the Facebook graph application programming interfaces (Graph API) tool on special news article, the Union Peace Conference-21st century Panglong Conference. This conference is very important for our country and people have various opinions on this. First time of conference began on 31 August 2016 and second time began on 24 May 2017. It is planned to the conference in every six month until the agreement is reached and negotiations and political dialogue will be continued. Occasionally, the government holds the panel discussion and revised previous conferences and prepares for the coming conference. People express their opinions on these conferences and events that are related to conferences by writing comments on Facebook news pages. We extract people comments related to peace conferences on news pages. Myanmar people use 10 popular news media pages and Information Committee page according to Myanmar Facebook page statistics by

socialbakers.com. Therefore, users' comments are collected from these news media pages using graph API. When data are collected, some of the articles are overlaps on news pages but the users are different. The following figure shows that number of posts related to this specific article and number of extracted comments from these media pages. There are 27337 comments are extracted from 11 news pages. Facebook posts can be as long as 5,000 characters and comments have a maximum of 8,000 characters. Some comments are too long but don't include any opinion words. Some people write just only opinion words. Average length of the comments in collected data is 21 words. The average number of words in collected data is 238532 words and the format of prepared lexicon is "word #polarity".

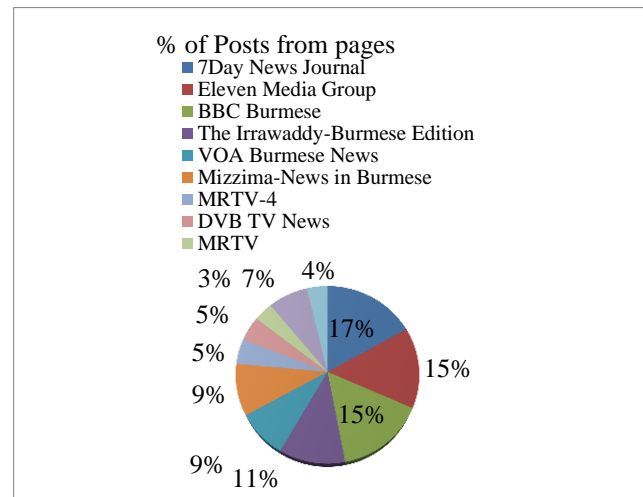


Figure1. Percentage of posts from Facebook pages

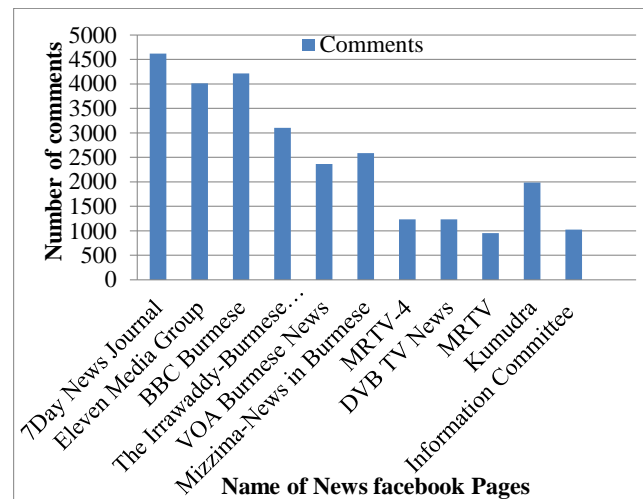


Figure2. Number of comments from news pages

3. Pre-processing

Online data have several flaws that potentially hinder the process of sentiment analysis. The general techniques for data collection from the web are loosely controlled. Therefore, the resultant datasets consist of irrelevant and redundant information. Several pre-processing steps are applied on the available dataset to optimize it for further experimentations. The proposed flow diagram for constructing sentiment lexicon is shown in figure 3.

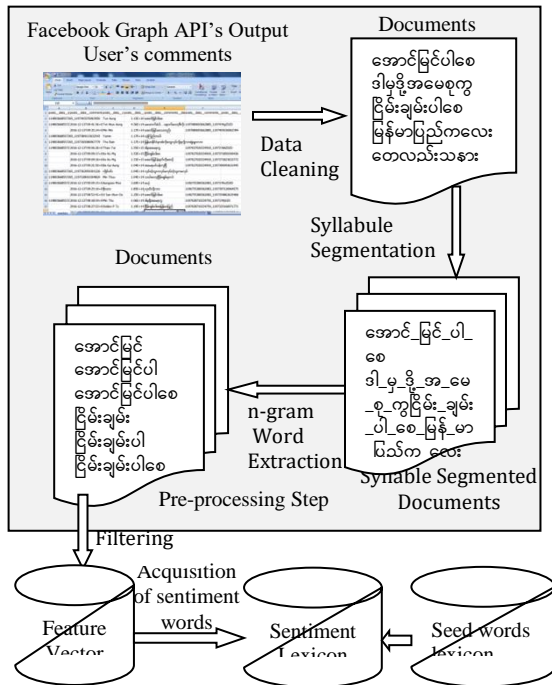


Figure3. Sentiment Lexicon Construction

3.1 Data cleaning process

Data must be preprocessed in order to perform any data mining functionality. Data preprocessing is done to eliminate the incomplete, noisy and inconsistent data. The biggest advantage of looking at words and phrases directly rather than using dictionaries is that one gets much more comprehensive coverage of the language that may be significant. Many of the “words” would not show up in any standard dictionary, either because they are emoticons or variant spellings, or because they are multiword expressions. Abbreviations are often used, orthographic mistakes are made on purpose and hashtags as well as emoticons are present in order to communicate the message of the author in a few words. Further, we find enthusiasm and extroversion are indicated by “အောင်မြင်ပါစေ (be success)” or “ထောက်ခံတယ်!!!! (Okay)” that is simply misspelling words indicates low

conscientiousness, or novel spellings signal multicultural backgrounds. New types of words such as emoticons (:) and <3), hashtags (“#Bieber”), URLs (“http://www.aapss.org”), photos or stickers comments are presented in social media data. The aforementioned flaws have been somewhat overcome in the following ways:

Step 1: In Myanmar language there are two types of fonts; Unicode based fonts and ad hoc font (Zawgyi). Facebook supports two types of these fonts. But most of the Myanmar people write comments using ad hoc fonts because firstly Facebook supports only this font. This font is not obeying Unicode encoding rules and different to process for other works such as searching, segmentation, etc. Unicode is the standard for character encoding in Myanmar. Thus, all comments extracted from Facebook are changed to Unicode fonts. There are many code conversion tools (Kanaung converter, Myanmar NLP Unicode converter, ThanLwinSoft converter, Burmese Font converter and etc.) for Myanmar language. In this work, reliable Burmese Font converter [15] is used for this purpose. In this font converter website, users give ratings about converter. This converter is 99% convertible now. Moreover, 1000 comments written in ad hoc font change into Unicode font using this converter and we check manually the output results. Converter can change 1000 comments correctly. But it fails in changing in some Myanmar Pali (Buddhism) words. These words are rarely found in people comments.

Step2: Eliminate all words except Myanmar Unicode characters. By doing like this, white spaces between characters, emoji comments, myanglish (written mixing with Myanmar and English words) words, number characters, blank lines, symbol characters and punctuation marks are already eliminated. Orthographic mistakes and multiword expressions still exist in comments dataset.

3.2 Syllable segmentation process

As mention above, Myanmar language does not have boundary word markers. Syllable segmentation is done on cleaning dataset by using syllable segmentation tool. This tool is developed by University of Information Technology (UIT) students. The accuracy of this tool is 100% for our test data and available for research purpose. Test data contains sentences from Myanmar middle school textbook. But if sentences do not match with syllable units; it may be fail in segmentation. However, at the present time, this tool can fully support for our work based on testing. In the later, we will test and compare with other syllable segmentation tools for this work. In this paper, word segmenter is not used. Myanmar language has resource spare problem. Thus, large-scale

general purpose lexicons are not publicly available at the present time. For this issue, this paper presented construction of specific domain-specific lexicon.

3.3 Words Extraction process

Syllable segmented dataset is the input for the entity extraction module. The sentence will have some valuable information about its sentiment and the rest of the words will not give any clue regarding the sentiment. Such words should be removed from sentiment lexicon. Syllable segmented dataset is tokenized applying the n-gram method by setting the minimum and maximum grams. According to our analysis result on 2000 comments, bigram is set as minimum and 6-gram as maximum gram. As the same time, words extraction process also calculated frequency count for each n-gram word.

4. Filtering process

After pre-processing on the dataset, domain dependent n-gram words are already extracted. Many duplicated n-grams and meaningless words also appeared. The more training data, the more duplicated and unnecessary words such as verb suffixes for sentiment analysis are came out. The performance of the lexicon can increase by filtering some unnecessary non-opinion words from n-gram lexicon features. There are two steps to perform this purpose.

Step1: From n-gram words, the spare words or features are removed based on frequency counts. In order to avoid data sparseness problem frequency counts of the word is less than 5 times; this word is removed from lexicon bag of words list.

Step2: The N-gram words list also contains Myanmar stop words, verb suffixes eg. (“ဝိဝေဝေ”), conjunction words, preposition words and numbering suffixes words eg. (“ထွေထွေ”). These words are also removed from the list. Myanmar stop words list, verb suffixes list and numbering suffixes list are created in my previous research work. Some stop words are manually corrected based on 1000 comments. There are totally 603 stop words for this work.

5. Generating sentiment lexicon

This step is for the acquisition of sentiment words in lexicon. The words in the lexicon have positive or negative polarity. Some of the words may be neutral. This paper only focuses on positive and negative polarity classes. Therefore, it is needed to decide polarity of every word in the lexicon and filtering neutral word from lexicon as much as we can. The main purpose is to increase the system accuracy, and to decrease the computational cost because of the overuse data. The

selection process is conducted by selecting every relevant feature that is for the input feature having a correlation to the output from the system. Firstly, seed words lexicon is created from labeled comments. Opinion words are extracted from these comments to create seed lexicons by manually. Positive seed words and negative seed words are manually extracted from positive and negative comment sentences. There are 305 positive words and 201 negative words. There are two steps in building sentiment lexicon.

Step1: words in lexicon are labeled the class (positive or negative) by using seed words lexicon and calculate the probability

$$w_i^c = \log \frac{p(w_i, w_c)}{p(w_i)p(w_c)}$$

And then calculate class probability of correction word labeled and unlabeled word. Selected maximum probability of class polarity and label this word with this class. If this combination can determine positive or negative, we label this combination words as maximum probability of polarity class. Repeat this steps again until no more combination words appear. After this process, some words remain as unlabeled words in lexicon.

Step2: Statistic approach is one effective way to do a feature selection process within the data. The word scores of the words are tested based on chi-square method. It also creates a list of all positive and negative words. There are two events, the observed count “O” and the expected count “E”. Chi-square measures how much the expected count and observed count deviate from each other. The two events are occurrence of the word/feature and occurrence of the class. When the two events are independent, the observed count is close to the expected count, thus a small chi square score. The higher value of the X^2 score, the more likelihood the feature is correlated with the class, thus it has to select for sentiment lexicon.

$$X^2 = \frac{N(AN - MP)^2}{PM(N - P)(N - M)}$$

A: the total number of positive words that contain feature X

M: the total number of words that contain feature X

N: the total number of words

P: the total number of positive words

After applying step1 and step2, unlabeled remaining words are assumed as neutral and removed from lexicon.

6. Lexicon Coverage Analysis

Sentiment lexicon represents how a word is distributed among the set of all opinion words. If the classification result is not optimally distributed across the space of unique words, it might be better to greedily increase the word coverage from the perspective of the sentiment

lexicon extraction. To approach the initial dataset problem from the lexicon coverage view, we test lexicon coverage analysis for finding maximum lexicon size for classification. Firstly, we put longest 1000 sentences in the system and extracted opinion words. We iteratively add the next 1000 sentence that has the minimum cosine similarity between the words that have been covered. According to analysis, the sentiment lexicon for news domain is stable (no new opinion words appeared) over 8000 training sentences.

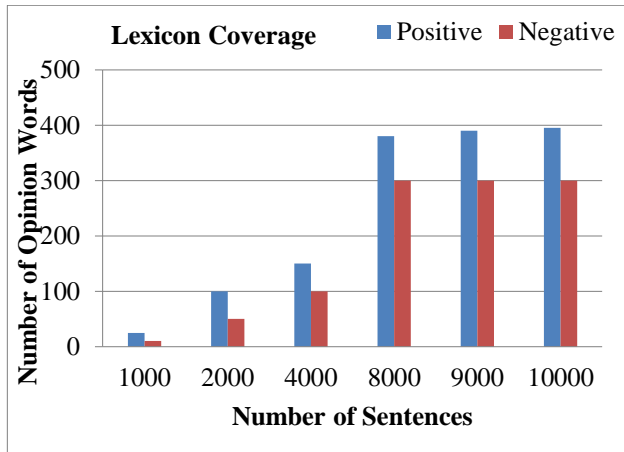


Figure4. Lexicon Coverage Analysis Results

7. Experimental Results

Above section we analyzed coverage of lexicon. We test how effect this sentiment lexicon on news domain in sentiment labeling task. To evaluate the accuracy, we followed a 5-fold cross validation process on 17337 sentences: each dataset was randomly split into five different non overlapping training and test sets. Each fold has 3467 sentences. There are 27337 comment sentences. In this work, 10000 sentences are already used in sentiment lexicon construction process. Therefore, remaining 17337 sentences are used to test created lexicon. Precision measures how many sentiment words are correctly identified in their classes. Recall expresses how many words, in the whole set, have been correctly recognized: a low recall means that many relevant comments are left unidentified. We assumed that a word not found in the lexicons has a neutral polarity. Five-fold cross validation results precision is 60.4%. Recall on testing dataset is 71%. Five folds cross validation results are shown in Figure 5.

Error Analysis is carrying on every fold results. People have complex ways of expressing opinions. We manually performed error analysis on dataset. Error analysis revealed that most of the errors are revealed to neutral words. Some of the positive polarity words come out as neutral. This fact leads to decrease performance of the lexicon. There are two type of strong errors are

recognized. First error is negative prefix words. Some of the positive words have negative polarity with negative prefix of Myanmar language. But the system cannot label such words correctly. Another is spelling missing, spelling mistake and dialect. The system cannot determine their classes. But they have respective polarity class.

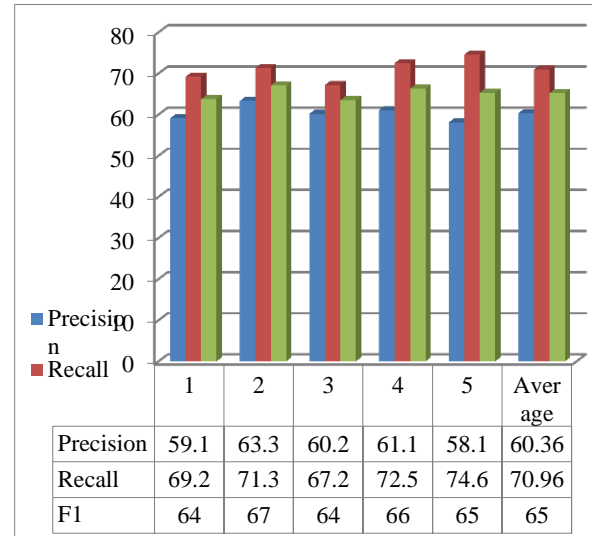


Figure5. Cross Validation Result

8. Conclusion and Future Work

In Myanmar, the intensive use of Internet, especially social media, to express opinion or view on certain matter, marks the opportunity to further develop the research in this field. In a sentiment analysis task choosing lexicons from the appropriate domain is important. There is a need for a method which can create domain-specific lexicons because there are no lexicons for every domain and creating them manually is expensive and requires an expert in that domain. The input of these methods is a small seed lexicon (semi-automatically) and unlabeled domain-specific texts. The best results were given by the manually assembled lexicon. But they are much more expensive and it is hard to create one for all domains, thus automatic methods are needed. The result shows that the proposed method is useful for increasing the performance of sentiment analysis systems in all domains.

As to future work, we intend to combine unsupervised feature selection method for lexicon expansion. Moreover, another work is needed on improving the accuracy of the sentiment classification on huge amount of information dataset.

9. References

- [1] H. Andrew Schwartz and Lyle H. Ungar, "Data-Driven Content Analysis of Social Media: A Systematic

Overview of Automated Methods”, ANNALS, AAPSS, 659, May 2015.

[2] J. Novakovic, P. Strbac, and D. Bulatovic, “Toward Optimal Feature Selection using Ranking Methods and Classification Algorithms”, Yugoslav Journal of Operations Research, DOI: 10.2298/YJOR1101119N, 21(2011), Number 1, pp.119-135.

[3] A. Go, R. Bhayani, and L. Huang, “Twitter sentiment classification using distant supervision,” CS224N Project Report, Stanford, vol. 1, 2009.

[4] Ana-Maria Popescu and Oren Etzioni, “Extracting Product Features and Opinion from Reviews”, Proceeding of Human Language Technology Conference and Conference on Empirical Methods in Natural Language, ACL, Vancouver, October 2005, pp. 336-339.

[5] S. R. Singh, H. A. Murthy and T. A. Gonsalves, “Feature Selection for Text Classification Based on Gini Coefficient of Inequality”, JMLR: Workshop and Conference Proceedings 10, the Fourth Workshop on Feature Selection in Data Mining, pp.76-85, 2010.

[6] Songbo Tan, Xueqi Cheng, Yuefen Wang, and HongboXu, “Adapting naive bayes to domain adaptation for sentiment analysis”, In European Conference on Information Retrieval. Springer, 2009, pp. 337–349.

[7]V. Jijkoun, M.de Rijke, and W. Weerkamp, “Generating Focused Topic-specific Sentiment Lexicons”, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Uppsala, Sweden, 11-16 July 2010, pp. 585-594.

[8] N. Kurian and S. Asokan, “Summarizing User Opinions: A Method for Labeled-Data Scarce Product

Domains”, International Conference on Information and Communication Technologies (ICICT 2014), ScienceDirect, 2015, pp. 93-100.

[9] N. Kaji and M. Kitsuregawa, “Building Lexicon for Sentiment Analysis from Massive Collection of HTML Documents”, Proceeding of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Prague, June 2007, pp. 1075-1083.

[10] L. Zhang, R. Ghosh, M. Dekhil, M. Hsu and B. Liu, “Combining Lexicon-based and Learning –based Methods for Twitter Sentiment Analysis”, HP Laboratories, 2011.

[11] F. Del Vigna, A. Cimino, F. Dell’Orletta, M. Petrocchi, and M. Tesconi, “Hate me, hate me not: Hate speech detection on Facebook”, 2015.

[12]V. Hangya, “Automatic Construction of Domain Specific Sentiment Lexicons for Hungarian”, 2013.

[13] A. Putra N and H. Sujaini, “Analysis of Extended Word Similarity Clustering based Algorithm on Cognate Language”, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol.4 Issue 11, November 2015.

[14] K. Labille, S. Gauch and S. Alfarhood, “Creating Domain-Specific Sentiment Lexicons via Text Mining”, Proceedings of workshop on issues of sentiment discovery and opinion mining, Halifax, Canada, August 2017.

[15]<http://burglish.mymm.org/latest/trunk/web/fontconv.htm>

Feature Selection for Categorization of Online News Articles in Myanmar Language

Myat Sapal Phyu, Win Win Thant, Thet Thet Zin

University of Information Technology, Yangon, Myanmar

myatsapalphyu@uit.edu.mm, winwinthant@uit.edu.mm, thetthetzin@uit.edu.mm

Abstract

In text mining, the feature selection plays an important role to reduce the high dimensionality of feature space. It can improve the accuracy of the document clustering process and help to avoid overfitting problem. Nowadays, the enormous amount of news article documents is widely available on the internet due to the rapid development of the web. Consequently, there is an urgent need to extract useful content from overloaded information. The categorization of online text documents is crucial to avoid information overload and it can help readers to find rapidly their interesting topic. The problem arises for text categorization is the large number of features space. This study has two phases, documents preprocessing and feature selection. Document preprocessing contains documents collection, syllable segmentation, word segmentation, removing stop words for extracting features from the collection of Myanmar online news documents including sport, health, crime etc. In this study, TF-IDF weighting method is adapted for feature selection. The experimental result shows the adapted TF-IDF method has higher performance than based TF-IDF method.

Keywords- Feature Selection, TF-IDF, Syllable Segmentation, Word Segmentation, Myanmar Online News

1. Introduction

In recent year, the rapid growth of using the internet leads to the information overload. People get a lot of information through the internet every day and waste much time to select their interesting information. Consequently, there is an urgent need to extract useful content from the enormous amount of information quickly and effectively. The categorization of news events is an important research area in text mining. It can enable the aggregation of news stories by topic and provide the basis for news recommender systems, a subclass of information filtering system. It can help readers to get the brief information about news documents before they read it. It is the way to automatically categorize news documents into predefined categories such as sports, education, and

technology etc. The main difficulty in text categorization is the high dimensionality of feature space. Ordinarily, the large number of features presents in the collection of documents and a few are informative. Accordingly, some important features are needed to select to reduce the dimensionality of feature space because it contributes directly to the accuracy of the document clustering. This study focuses on feature selection for categorization of Myanmar online news articles.

In this study, some preprocessing steps are needed to perform before extraction and categorization of news articles. The preprocessing steps contain listing stop words, syllable segmentation and word segmentation. Word segmentation is performed for extracting features (words) from collection of text documents.

Feature dimension reduction is an important part in text categorization. This study analyzes the TF-IDF method and makes adaption based on this method in order to get high accuracy for feature selection. In information retrieval, the TF-IDF is a well-known method to evaluate how important is a word in a document. It is a very interesting way to convert the textual representation of information into a term vector model. It is an algebraic model for representing text documents as vectors of identifiers.

The first step in converting the document into a vector space is to create a dictionary of terms (words) by selecting all terms from the document and transform it to a dimension in the vector space. As the main task is to select important features from documents, the stop words are ignored because they are not helpful to categorize the text documents. So, stop words list in Myanmar language is created by analyzing Myanmar online news documents to remove unnecessary features.

In order to extract features from collection of documents, it needs some preprocessing steps. In preprocessing step, the syllable segmentation and word segmentation are considered in order to specify each separate word as one feature. After extracting features, important features are selected by adapted TF-IDF method and compare the performance with existing TF-IDF method.

The rest of this paper is as follows, section 2 describes the related research that was published in the area of Myanmar word segmentation, syllable

segmentation and feature selection methods. Section 3 discusses the overview of feature selection process including collecting text document from Myanmar daily news websites [12] [15], preprocessing tasks, feature selection by TF-IDF method and its adaptation. In section 4, the nature of data set is discussed. The detail steps of preprocessing steps are explained in section 5. In section 6, feature selection by TF-IDF [10] method is presented. According to the testing result of baseline method, it is adapted in TF-IDF (Adaptive Method) in order to get better solution and discuss in section 7. Experimental results are discussed in section 8. According to the experiments, some problems are pointed in section 9. The last section concludes and discusses the future works.

2. Related Work

Document pre-processing and feature selection approaches are useful for text categorization process. Myanmar word segmentation and syllable segmentation play an important role in document pre-processing task. Many researchers did in Myanmar word segmentation [2] and syllable segmentation by using different methods [6] [7] [11] [2]. The feature selection approaches are most important research area in text mining and implemented by different methods [3] [4] [5].

Manually constructed context free grammar (CFG) is presented in [6] to describe the Myanmar Syllable Structure to identify Myanmar syllables. The syllable segmentation algorithm that can slice all of the input text string is developed in [7]. The input text strings are converted into equivalent sequence of category form and compare the converted character sequence with the syllable rule table to determine syllable boundaries. A syllable segmentation tool is developed for Myanmar text encoded with Unicode in [11].

The two steps method for syllable segmentation and syllable merging are proposed in [2]. Syllable boundaries are determined by the proposed six syllable segmentation rules and dictionary-based statistical approach is used to perform syllable merging.

In [3], terms are extracted from the documents by using term selection approaches tf-idf, tf-df and tf2 based on their minimum threshold value. This approach is intended to reduce the attributes and find the effective term selection method using WordNet.

In [4], Support Vector Machine is applied to classify Bangla document and TF-IDF is used for feature selection.

A new weighting method named TF-IDF-CF is proposed in [5] based on TF-IDF. It introduced a new parameter class frequency, which calculates the term frequency in documents within one class.

The main aim of this study is to adapt TF-IDF (Baseline Method) by analyzing, testing the baseline method with the input data set, Myanmar text documents collected from Myanmar online news websites [12] [15]. The experimental result shows the better performance of the TF-IDF (Adaptive Method) than baseline method. In the future, various feature selection methods will be tested and implemented with more data set in order to find better solution and intend to find high informative features for categorizing Myanmar online news documents in the future. It is intended to support the Myanmar online news categorization process to overcome the difficulties of Myanmar news readers to get their desired news rapidly.

3. Overview of Feature Selection Process

Figure 1. shows the overview of the feature selection process for categorizing Myanmar online news articles. Firstly, text documents are collected from social news websites. Then, the features are extracted from input text documents.

In order to extract features, input text documents are segmented into syllables and merged these syllables into meaningful word by matching Myanmar words dictionary [13] and then unnecessary words including city, date time words, number, non-Myanmar character and so on are removed. After that, features are selected by TF-IDF (Baseline Method) [10] and adapted the baseline method in order to get high accuracy. Then the selected features are collected into lexicon for the future categorization process.

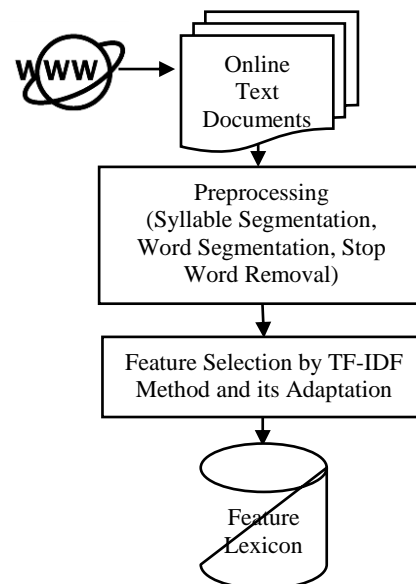


Figure 1. Feature Selection System Design

4. Data Set

Text data are collected from a Myanmar daily news website [12] [15] and extracted text by online text extractor [9]. Then, extracted text are converted into Unicode format by Zawgyi-One to Unicode converter [14]. Each news article is saved as text document (.txt) and used as input data set for feature selection process. Each news article generally contains about 10 sentences. In this study, 383 news articles for crime category, 320 news articles for sport category and 283 news articles for health category are collected as text documents and more articles for updated news will be collected in the future. Table 1. shows the data set on three different news categories. 16,381 features from 383 crime news, 12,273 features from 283 health news and 14,801 features are extracted from sport news.

Table 1. Data Set on Three Online News

No.	Category	Number of Documents	Number of Extracted Features
1.	Crime	383	16,381
2.	Sport	320	14,801
3.	Health	283	12,273

5. Preprocessing

Before the extraction of features from news documents, the following preprocessing steps are required to extract the features:

5.1. Syllable Segmentation

In order to extract the news features, it is needed to determine the word boundaries. The syllable boundary is determined as the preprocessing task for word segmentation. In this study, syllables are segmented by syllable segmentation method that is implemented by regular expression pattern [11]. The pattern is based on encoding order of Myanmar syllable.

According to the encoding order of Myanmar syllable [8], most of the Myanmar syllables start with consonant and the starting character of each syllable can also be determined as end of preceding syllable. So, the syllable boundary is determined by checking the consonant and marks with syllable boundary marker in front of the consonant except the consonant after a subscript character (“၂”), the consonant that is followed by a-that character (“၃”) or subscript character (“၂”) and standalone syllable such as (“က, ဤ, ဥ, ဦ, ဧ, ဩ, ဪ, ါ, ာ, ိ, ီ”). This approach can reduce time and space complexity and outperform than other syllable segmentation methods.

5.2. Word Segmentation

After segmenting each syllable, the segmented syllables are merged to form a word. The input text is segmented into each individual syllable. And then, segmented syllable are merged to become meaningful word by dictionary based maximum syllable longest matching approach using Myanmar words lists [13]. Myanmar words list contains 41482 words and some missing words in dictionary that are mostly used in sport, crime and health online news are added to this list by analyzing 383 crime documents, 320 sport documents and 283 health.

For instance, the popular football terms “ဦးဆောင်ဦး, ချေပဦး, စပေးဘောလုံး”, the name of famous professional footballer and the coach such as “စီရော်နယ်ဒို, အာစင်ဝင်းဂါး” and the most popular virus and diseases in medical field such as “ရာသီတုတ်ကွေး, လူတုတ်ကွေး, ဇီကာဦးရပ်စ်” are considered to be added to the Myanmar words lists. The words related with sport domain are mostly added to the dictionary because most of the sport news are international news and not contain in dictionary. Currently, 41767 words are presented in the Myanmar words list.

5.3. Removing Stop Words

Stop words are the set of commonly used words in any language. They are removed from feature space to reduce the noise and to enhance the computational efficiency of categorization.

In this study, stop words are collected by analyzing Myanmar online news. Most of the news contains the location and time information that are not important terms for categorizing news documents. After analyzing news documents, location, date, time words and the most commonly used prepositions, inflections; conjunctions are collected as stop words. Moreover, punctuation marks (eg., “။ .”), white spaces and other symbols (eg., “-/()[]{}”), non-Myanmar text (eg; A to Z), numerical text (eg., 0 to 9 ခု to ခု) are removed. In this study, 608 stop words are collected and more stop words will be added in the future. Table 2 shows the sample of stop words list.

Table 2. Sample of Stop Words List

ဧပြီ၊ ဇွန်၊ ဇူလိုင်၊ ဩဂုတ်၊ နာရီ၊ မိနစ်၊ စက္ကန့်၊ နေ့လည်၊ နတ်တော်၊ ပြာသို့ တပို့တွဲ
တိုင်းဒေသကြီး၊ မြို့နယ်၊ မြို့သစ်၊ ကျေးရွာအုပ်စု၊ ဘိုကလေး၊ ဓနုဖြူ၊ ဒေးဒရို၊ ဖင်လာဒို။
နောက်ထပ်၊ တခြား၊ နောက်တစ်ခု၊ နောက်တစ်ချက်၊ နောက်တစ်ယောက်။
သည့်အပြင်၊ ထို့ပြင်၊ ထို့အပြင်၊ ဒါ့အပြင်၊ နောက်ပြီး။

6. Feature Selection by TF-IDF (Baseline Method)

Features are selected and tested by adapted TF-IDF and original TF-IDF method, short for Term Frequency–Inverse Document Frequency that is intended to reflect how important a word is to a document in a collection or corpus. The TF-IDF value increases proportionally to the number of times a word appears in the document (one domain), but is offset by the frequency of the word in the corpus (all domains), which helps to avoid multi label problem.

$$TF - IDF = TF(t, d) * IDF(t, D) [10]$$

$$= \frac{Freq(t, d)}{|D|} * \log \frac{N(D)}{Freq(t)} \quad (1)$$

Freq (t, d) - the frequency of term t in one domain d (among sport or crime or health)

|D| - total number of term in the domain

IDF - Inverse Document Frequency

N (D)- number of domains

Freq (t) - number of domains that contain term t

TF-IDF method is tested with 16,381feature (terms) extracted from 383 crime news documents, 12,273 features (terms) extracted from 283 health news documents and 14,801 features (terms) extracted from 320 sport news documents from Myanmar news media sites [12] [15]. TF value is calculated within one domain. and IDF value is calculated across the domains. TF-IDF score of each feature is calculated and ranked by TF-IDF score and top 100 features will be selected as important feature for each domain. According to the some findings on TF-IDF (Baseline Method), it is adapted in TF-IDF (Adaptive Method) and the details are briefly explained in section 7.

7. Feature Selection by TF-IDF (Adaptive Method)

The concept of TF-IDF [10] is that TF-IDF score is highest when TF value is high (high frequency of term in one domain) and IDF value is high (does not contain frequently in all domains).

In this study, TF-IDF (Baseline Method) is adapted mainly for two kinds of features. Firstly, the features that contain in few domains (high IDF value) but it does not have much frequency in each domain (low TF value) have low TF-IDF scores even though they may be high informative than other features that have high term frequency and low IDF value. To solve this problem, if

the value of term frequency is higher, its value may raise than other features that contain in many documents. If we calculate the term frequency value with logarithm in both numerator and divisor as $TF = \log(Freq(t,d)) / \log(|D|)$, the quotient will be little higher than original TF method.

For instance, suppose that we have to consider four domains sport, education, health and sport ($N(D)=4$) and each domain has 500 extracted features ($|D|=500$). The feature "Champion" contains 10 times in sport domain ($Freq(t, d) = 10$) but not contains in other ($Freq(t)=1$). Then, the feature "Agreement" contains 50 times in sport domain ($Freq(t, d) = 50$) and it also contains in other two domains ($Freq(t)=3$). TF score of the feature "Champion" is 0.04, IDF score is 0.6021 and TF-IDF score of the feature "Champion" in sport domain is 0.012042.

TF score of the feature "Agreement" is 0.2, IDF score is 0.1249 and TF-IDF score of the feature "Agreement" in sport domain is 0.01249. According to the TF-IDF results, the feature "Agreement" that contains in many domains with high term frequency has higher TF-IDF score than the feature "Champion" that contains in only one domain that has low term frequency. The results are depicted in Table 3.

Table 3. Comparison TF-IDF Scores of Two Features by TF-IDF (Baseline Method)

Feature	TF	IDF	TF-IDF
Champion	0.04	0.6021	0.012042
Agreement	0.2	0.1249	0.012490

The purpose of adapting equation is to raise the value term frequency value for the features that only contain in one domain than the features that contain in many domains. By adjusting these values, the accuracy of selected features is higher than baseline method according to the experiments. Table 4. shows the TF-IDF score of the feature "Champion" and "Agreement" by adapted TF-IDF method. the feature "Champion" get higher TF-IDF score than the feature "Agreement". In these tables data are simplified for illustrative purpose.

Table 4. Comparison TF-IDF Scores of Two Features by TF-IDF (Adaptive Method)

Feature	TF	IDF	TF-IDF
Champion	0.3705	0.7021	0.260128 1
Agreement	0.6295	0.2249	0.141574 6

Secondly, in case of the features that have IDF value zero, IDF value is added 0.1 for smoothing as $IDF = \log(N(D)/Freq(t)) + 0.1$. Suppose that, we have to consider for two domains, sport and crime. For instance,

the term “Football” is frequently found in sport news. So, term frequency of “Football” in sport domain may be high. The term “Football” is also found in crime news as “Sixteen-year-old football player is killed instantly after the 10-foot log struck him on Thursday”. Although the term “Football” is found in crime news, it is a condition that is not occurring very often that term frequency of “Football” in crime domain may be low. If we use as $\log(N(D)/\text{Freq}(t))$, the score of TF-IDF value for both domain will be zero because $\log(2/2)=\log(1)=0$. If we add 0.01, TF-IDF score will be higher than that has higher term frequency value (sport domain).

In brief, the adapted TF-IDF method is to justify mainly for features that have very low term frequency with high IDF value and justification of TF-IDF value that has IDF value zero. Equation 2 shows the adapted TF-IDF method.

$$TF - IDF = \frac{\log(\text{Freq}(t, d))}{\log(|D|)} * \log\left(\frac{N(D)}{\text{Freq}(t)}\right) + 0.1 \quad (2)$$

8. Experimental Results and Discussion

In this study, 16,381 crime features, 14,801 sport features and 12,273 health features are extracted from 320 sport documents, 383 crime documents, and 283 health documents. Then features are selected by TF-IDF model and its adaptation model and then the performance is experimentally evaluated. Table 5. (a), (b), (c) show the top 8 terms for each category including sport, health and crime, ranked by TF-IDF score. In these tables, data are simplified for illustrative purposes. In reality, top 100 features are selected as important keywords for each category.

Table 5. (a) Sport Terms Table 5. (b) Health Terms

Sport Category		Health Category	
Term	TF-IDF	Term	TF-IDF
မြိုင်ပဲ	0.61569	ကင်ဆာ	0.49203
ယုဉ်မြိုင်	0.59391	သုတေသီ	0.48936
ကစားသမား	0.58658	ရဲဘိုင်နှုန်း	0.46564
ပရိသတ်	0.52659	သုတေသန	0.44441
ချွန်ပီယံ	0.51387	နှလုံး	0.4405
ကလပ်	0.50700	ကိုယ်ဝန်ဆောင်	0.4189
ဆီမီးပိုင်နယ်	0.49973	လက္ခဏာ	0.39852
	0.46233	မှတ်ဉာဏ်	0.38696

Table 5. (c) Crime Terms

Crime Category	
Term	TF-IDF
	0.66555
အမှု	0.59743
ပုဒ်မ	0.53251
ယူဆောင်	0.5286
ထုတ်ပြန်	0.47805
ခရီးသည်	0.47805
စီးသတ်	0.47506
အခင်းဖြစ်	0.47200
ပြုလုပ်	

In this study, to evaluate the performance of feature selection process, top 100 features are selected as positive tuples (important features) and 300 features that have lowest TF-IDF scores are selected as negative tuples (not important features to be removed). To check the accuracy of selected features, domain specific lexicons that contains the most widely used words about 350 words for each topic are manually constructed. Table 6. and 7. show the evaluation measure by TF-IDF (Baseline Method) and TF-IDF (Adaptive Method). As TF-IDF scores of the features that only contain in one domain increase and TF-IDF scores of the features that have much term frequency but contain in many domains (that can cause multi-label problem) decrease in TF-IDF (Adaptive Method), the performance of adaptive method is better than baseline method. According to the experiment, adapted TF-IDF model shows an improvement in performance than existing method.

Table 6. Evaluation Measure by TF-IDF (Baseline Method)

	Precision (%)	Recall (%)	F-Measure (%)	Error Rate (%)	Accuracy (%)
Crime	91	51	65	14	87
Sport	97	68	79	9	91
Health	91	73	83	11	91

Table 7. Evaluation Measure by TF-IDF (Adaptive Method)

	Precision (%)	Recall (%)	F-Measure (%)	Error Rate (%)	Accuracy (%)
Crime	96	59	73	11	89
Sport	100	77	87	5	94
Health	97	82	83	8	95

Table 8. describes the formulas of each measure and terms used in these formulas are briefly described [1].

Table 8. Evaluation Measure

Measure	Formula	Description
Accuracy	$\frac{TP + TN}{P + N}$	TP - True Positives refer to positive tuples correctly labeled
Precision	$\frac{TP}{TP + FP}$	TN - True Negatives refer to negative tuples correctly labeled
Recall	$\frac{TP}{TP + FN} = \frac{TP}{P}$	FP -False Positive refer to negatives tuples that were incorrectly labeled as positive
F-Measure	$\frac{2 * Precision * Recall}{Precision + Recall}$	FN -False Negative refer to positive tuples that were mislabeled as negative
Error Rate	$\frac{FP + FN}{P + N}$	P –the number of positive tuples N -the number of negative tuples

9. Error Analysis

The problem with TF-IDF method is that the ranges of TF-IDF scores for each domain are not on the same scale. The domains with large number of extracted features have higher TFIDF values than the other domains with the smaller number of features. So, it is needed to justify the number of input features for each domain in order to get reasonable scores. Then, the problem of out of dictionary words, for instance, the word "ဝယ်နယ်" has high tf-idf score in sport domain because the word "ဝယ်နယ်တီ" is often used in sport news and the word "ဝယ်နယ်တီ" does not contain dictionary. In this case, such kinds of words are added to Myanmar words list but some of the words cannot be noticed.

10. Conclusion and Future Works

This study especially focuses on feature selection and the aim of this study is to select high informative features from collection of online news documents for future online news categorization process. Myanmar news features are selected by TF-IDF (Baseline Method) and TF-IDF (Adaptive Method). The experimental results show the higher performance of adaptive method than baseline method. Further experimental work will be performed with cosine similarity on latent semantic analysis (LSA) vectors, the Latent Dirichlet Allocation (LDA) model and other feature selections methods then more categories of online news articles will be considered in the future.

9. References

- [1] Han, Jiawei and Kamber, Micheline and , and Pei, Jian "Data Mining: Concepts and Techniques (Third Edition)", Morgan Kaufmann, Third Edition, The Morgan Kaufmann Series in Data Management Systems, Boston, 2012, pp. 365-366.
- [2] Tun Thura Thet, Jin-Cheon Na, Wunna Ko Ko, "Word Segmentation for the Myanmar language", *Journal of Information Science* 34 (5), 2008, pp. 688-704.
- [3] Dadgar, Seyyed Mohammad Hossein, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. "A Novel text Mining Approach Based on TF-IDF and Support Vector Machine for News Classification." *In Engineering and Technology (ICETECH), 2016 IEEE International Conference on*, IEEE, 2016, pp. 112-116.
- [4] Islam, M.S., Jubayer, F.E.M. and Ahmed, S.I., "A Support Vector Machine Mixed with TF-IDF Algorithm to Categorize Bengali Document". *In Electrical, Computer and Communication Engineering (ECCE), International Conference on*, IEEE, February 2017, pp. 191-196.
- [5] Liu, M. and Yang, J., "An Improvement of TF-IDF Weighting in Text Categorization". *International Proceedings of Computer Science and Information Technology*, 2012, pp.44-47.
- [6] Tin Htay Hlaing, "Manually Constructed Context-Free Grammar for Myanmar Syllable Structure". *In Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL '12)*, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 32-37.
- [7] Z. M. Maung, Mikami Yoshiki, "Rule-based Syllable Segmentation of Myanmar Texts". *In Proceedings of the 6th Workshop on Asian Language Resources*, January 2008, Hyderabad, India, pp. 11-12.
- [8] K.Stribley, "Collation of Myanmar in Unicode", *technical report*, June 17, 2007.

[9] <https://boilerpipe-web.appspot.com>

[10] <https://en.wikipedia.org/wiki/Tf-idf>

[11] <https://github.com/ye-kyaw-thu/sylbreak>

[12] <http://news-eleven.com/>

[13] <https://raw.githubusercontent.com/lwinmoe/segment/master/burmese-word-list.txt>

[14] <https://thanlwinsoft.github.io/www.thanlwinsoft.org/ThanwinSoft/MyanmarUnicode/Conversion/myanmarConverter.html>

[15] <http://thithtoolwin.mmbloggers.com>

Software Engineering and Web Mining

Defining a Software Engineering Process with Cost-effective Security Requirements Implementation

Swe Zin Hlaing, Koichiro Ochimizu
University of Information Technology, Myanmar
swezin@uit.edu.mm , ochimizu@jaist.ac.jp

Abstract

Today, security problems involving computers and software are frequent, widespread, and serious. The number and variety of attacks by persons and malicious software from outside organizations, particularly via the Internet, are increasing rapidly, and the amount and consequences of insider attacks remains serious. Security is not just a question of security functionality; the properties desired must be shown to hold wherever required throughout the secure system. Because security properties are systems properties, security is an omnipresent issue throughout the software lifecycle. This paper describes the existing software development lifecycle with the integration of security engineering process. After that, we will adopt the Information System Environment in the case of the University of Information Technology (UIT). Moreover, this paper describes an Information Security Software Engineering (ISSE) process for discovering and addressing users' information protection needs based on the case study of UIT's information System Environment. Finally, the proposed system performs the quantitative risk analysis on the study of UIT Information System Environment.

Keywords- Information Security Software Engineering, Security Engineering Process, Quantitative risk analysis

1. Introduction

Security is often an afterthought during software development. A more effective approach for security requirement engineering is needed to provide a more systematic way for eliciting adequate security requirements. Information Systems Security Engineering (ISSE) is the art and science of discovering users' information protection needs and then designing and making information systems, with economy and elegance, so they can safely resist the forces to which they may be subjected. The main goal of this paper is to define the security software engineering (SSE) process with the existing software development lifecycle. In this process, the quantitative risk analysis is applied to SSE by implementing the cost-effective ways of security requirements. This paper is not intended to cover

security through the entire SDLC. This paper is organized as follows. Section "Security Software Engineering Process" discusses the traditional Software Development Lifecycle (SDLC) integrates with security process and the process of Information System Security Engineering. Section "Quantitative Risk Analysis" presents the analysis of information assets quantitatively. Section "The case study of UIT's Information Environment" that explain the information system of UIT.

Section "Evaluation of the process" that shows some value after performing risk analysis on UIT's information assets. Finally, the last section describes the conclusion and future works of this paper.

2. Security Software Engineering Process

The overview of the integration of security engineering process with an ordinary System Development Lifecycle (SDLC) as shown in Figure.1.

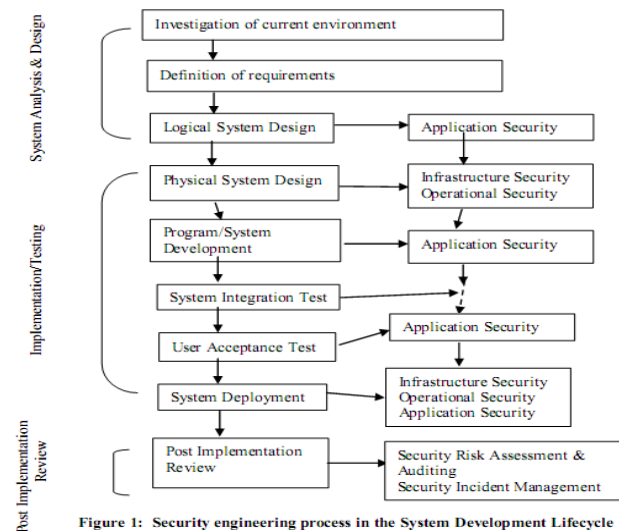


Figure 1: Security engineering process in the System Development Lifecycle

In this figure, each phase of the SDLC considers three types of security level: Application security, Infrastructure security and Operational security. Realizing security early, especially in the requirement phase, is important so that security problems can be tackled early enough before going further in the process and avoid rework. Requirement errors can be expensive

if they are not detected and fixed early in the development process.

The security engineering process proposed by Ian Sommerville is shown in Figure 2. Therefore, this paper proposed the integration of security engineering into the early phase of SDLC such as requirement definition phase and design phase.

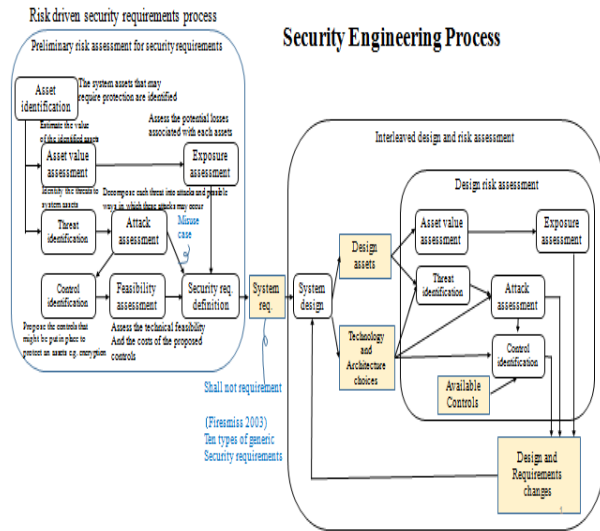


Figure 2: The security engineering process proposed by Ian Sommerville [1].

3. Quantitative Risk Analysis Process

The paper [2, 3] shows the step by step procedure of performing quantitative risk analysis. The main processes of quantitative risk analysis are as follow:

Perform risk assessment and vulnerability study: If the software development needs to consider the security, it is ensure to perform risk assessment and vulnerability study to produce risk factor matrix.

Estimate the cost of tangible/ intangible assets: During the study, we identify two types of assets such as tangible and intangible assets together with their estimated value. After that we need to determine the asset value.

Estimate the potential exposure factor (EF): We can estimate the exposure factor's value based on the identification of threats and/or survey and analyze to ask multiple questions to conduct the analysis.

Calculate Single Loss Expectancy (SLE): This process takes assets value and EF value as input parameters and produce the value of SLE.

Estimate Annualized rate of occurrence (ARO): Estimated frequency a threat will occur within a year and is characterized on an annual basis. A threat occurring once in 10 years has an ARO of 0.1; a threat occurring 10 times in a year has an ARO of 10.

Calculate Annualized Loss of Expectancy (ALE): It takes SLE and ARO as input parameters and produce ALE value.

Calculate Cost/Benefit Analysis: This process performs calculating the difference between the ALE prior to implementing the countermeasure to the ALE after implementing the countermeasures.

In this paper, we intend to integrate the security engineering process with quantitative risk analysis that emphasize on early stage of SDLC. Therefore, the proposed integration of the entire process that integrates security engineering process with quantitative cost analysis as shown in Figure 3.

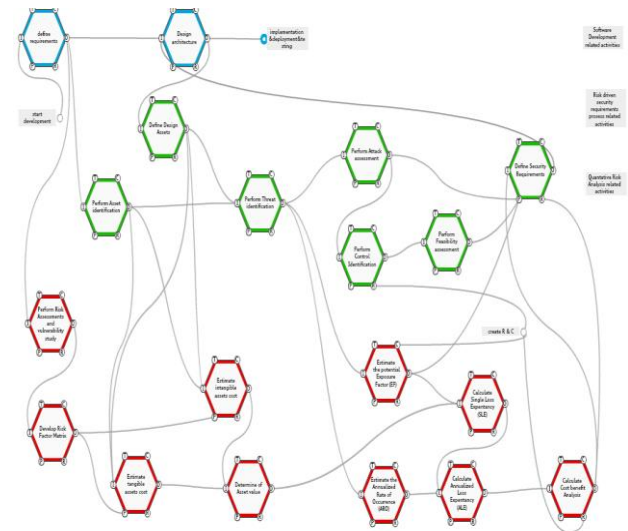


Figure 3: The entire integration process of security engineering process and quantitative risk analysis process

4. A case study of UIT's Information System Environment

By analyzing in the case of UIT Information Environment and list the following assets as shown in Table 1. To determine the cost of assets, we need to analyse the list of threats and vulnerabilities based on the case study.

In the section 4.1 describes the list of some identified threats based on our environment. And then vulnerability study on some of survey results describe in section 4.2.

Assigning values to tangible assets, the following are some typical methods for obtaining estimates for tangible assets.

Table 1. Categorization of Assets

Tangible Assets	Intangible Assets
Desktop PCs	Application Software
Laptop PCs	Technical Software
Servers	Electronic Data
Printers	Emails
Photocopiers	
Telephone	
Fax Machines	
Network Hubs and Routers	
Backup Media	
General Office Equipment	
Training Materials	
Personnel Files	

(a) Ask the IT manager for cost information regarding existing equipment, software and hardware.

(b) Conduct research on the Internet. Determine the age of current tangible assets, and calculate value by including depreciation.

There are two typical approaches for determining the valuation of intangible asset.

(a) Cost Approach – seeks to measure an asset’s fair market value, with depreciation also taken into account. The cost approach does not directly consider either the amount of economic benefits that can be achieved or the time period over which they might continue. A cost approach is typically used for valuing trade secrets and know-how.

(b) Income Approach – Focuses on the income producing capability of the intellectual property. The value is measured by the present value of the net economic benefit over the life of the assets. When the economic conditions are not favorable, the income approach leads to a relative low valuation of assets. This approach is best suited for the valuations of patents, trademarks, computer software, and copyrights.

4.1 Identification of Threats

We survey and analyze the UIT’s Information Environment [4] . We found out some identifiable threats and how often they occur and their impact as shown in Table 2 and Table 3.

Table 2. List of identified Threats

Ref.	Threat
1.1	Air condition failure
1.2	Damage to communication lines/ cables
1.3	Deterioration of storage media
1.4	Failure of communication services
1.5	Failure of network components
1.6	Failure of Database
1.7	Failure of power supply
1.8	Hardware failure
1.9	Illegal use of software
1.10	Maintenance error
1.11	Malicious software (eg. Viruses, worms. Trojan horses)
1.12	Software failure
1.13	Staff shortage
1.14	Theft

After identifying the possible threats of UIT’s environment, assign the risk values based on occurrences of threat (likelihood) and impact (severity). Table 3 shows the estimated risk assessment value on identified threats.

Table 3. Risk assessment value of threat

Ref.	Likelihood	Severity
1.1	M	VH
1.2	L	H
1.3	L	VH
1.4	M	H
1.5	M	H
1.6	M	H
1.7	L	VH
1.8	M	H
1.9	H	H
1.10	M	H
1.11	H	H
1.12	M	VH
1.13	M	H
1.14	M	H

4.2 Identification of Vulnerabilities

From the study of vulnerability, the list of vulnerabilities, both technological and organization-related, that can affect the organization's assets as shown in Table 4.

Table 4. List of identified vulnerabilities

Ref.	Vulnerability	Ease of exploitation
2.1	Absence of personnel	M
2.2	Insufficient security training	M
2.3	Lack of monitoring mechanisms	M
2.4	Inadequate recruitment procedures	M
2.5	Inadequate or careless use of physical access control to buildings, room and offices	M
2.6	Lack of physical protection for the building doors and windows	M
2.7	Location in an area susceptible to flood	H
2.8	Insufficient maintenance	M
2.9	Lack of periodic equipment replacement schemes	M
2.10	Unstable power grid	M
2.11	Lack of identification and authentication mechanisms	H
2.12	Inadequate network management	H
2.13	No "logout" when leaving the LAN	H
2.14	Uncontrolled downloading and using software	VH

5. Evaluation

After a vulnerability assessment and threat analysis, I have proceed to quantify the risk element. After conducting the survey of the organization, it would be much simpler if it can estimate ALE directly from using the risk analysis data referenced in paper [3]. In addition, I will need to add a ranking number from 1 to 10 for quantifying severity (with 10 being the most severe, and 1 of least severity) as a correction factor for the risk estimate obtained from the data table. For UIT's Information System Environment, I may study the

internet threats and issues such as uncontrolled downloading using software.

The estimated value of 78% detected students in UIT abuse of Internet access privileges (for example, downloading the video file in their classes or playing online game). So, this kind of vulnerability is very important for our environment. Conducting the risk analysis on uncontrolled downloading and using software, it has a severity ranking of 8 and we can use the corresponding adjustment factor used will be 1.1 as shown below.

Severity Ranking

10	9	8	7	6	5	4	3	2	1
1.2	1.2	1.1	1.1	10	10	0.9	0.9	0.8	0.8

Adjustment Factor

According to the data table in [5,6]

Annual revenue = \$ 0.01 Million

Number of students = 1000

Size Correction (using data from CSI) = $1000 / 4700 = 0.2$

ALEtable = \$ 536,000

ALEcorrected = $\$ 536,000 \times 1.1 \times 0.2 = \$ 117920$

I will study some survey data of ASIS report and estimate the ALE value of uncontrolled downloading using software in the case of UIT environment.

6. Conclusion and future works

This paper concerns the first step of integrating security engineering process and quantifying risk assessment. Then, we intend to analyze the critical security breaches concerning about our UIT environment and later on do the cost/benefit analysis on these data. After that the integration of all SDLC processes into the security engineering process should be performed. Finally, we need to evaluate our engineering approach is beneficial for financial and technological issue.

7. References

- [1] Sommerville,I " Software Engineering, tenth Edition, Pearson, 2015
- [2] ISO/IEC 27002:2005 Information technology- Security techniques- Code of practice for information security management , 2013
- [3] Tan.D, Quantitative Risk Analysis step-by-step, 2002

[4] Exemplar_ISMS Risk Assessment Manual Version1.4.rtf , <https://www.noexperiencenecessarybook.com/m7V6/isms-risk-assessment-manual-version-1-4.html>

[5] ASIS International. "Trends In Proprietary Information Loss Survey Report." Septem2002. URL: <http://www.asisonline.org/pdf/spi2.pdf>

[6] Computer Security Institute (CSI) . "2002 CSI/ FBI Computer Crime and Security Survey." Computer Security Issues & Trends.Vol.8, No.1 Spring 2002.

A Lightweight Size Estimation Approach for Embedded System using COSMIC Functional Size Measurement

Thandar Zaw, Swe Zin Hlaing, Myint Myint Lwin, Koichiro Ochimizu
University of Information Technology, Yangon, Myanmar
thandarzaw@uit.edu.mm, swezin@uit.edu.mm
myintmyintlwin@uit.edu.mm, ochimizu@jaist.ac.jp

Abstract

Functional Size Measurement (FSM) is an important component of a software project that provides information for estimating the effort required to develop the measured software. Although the embedded software is time-consuming to develop, COSMIC FSM can be estimated to get more accurate function size. The traditional Function Point methods are designed to measure only business application domain and are problematic in the real-time domain. As a result, COSMIC Functional Size Measurement (FSM) method is designed to measure both application domains. The design diagrams such as UML, SysML and the well-defined FSM procedure must use to accurately measure the functional size of embedded system. We have already developed the generation model based on SysML metamodel with an example of elevator control system. In this paper, we applied the generation model that is the classification of the instance level of object based on UML metamodel. After that, this paper also showed the mapping rules which mapped between the generation model and COSMIC FSM to estimate the functional size of embedded software with the case study of cooker system. This paper also proposed the light weight generation method of COSMIC FSM by using the generation model.

Keywords- Function Size Measurement (FSM); COSMIC (Common Software measurement Consortium); UML; MetaModel

1. Introduction

Software sizing is used to estimate the size of a software application. As the software development costs are increasing, the early prediction of the size of the embedded software should be achieved in the developmental process.

The functional size of software has become an important task in most of the industrial software development as it offers the valuable input to estimate the development effort model and tools. There are five measurement methods which have been recognized as standards IFPUG FPA [4], MK II FPA [5], FISMA [7],

NESMA FPA [6] and COSMIC FFP [8,9] to measure the functional size of software applications. COSMIC FSM has developed the most advanced method of measuring a functional size of software that overcomes several limitations of the traditional FSM methods. The traditional FSM methods are applied in business application domain and they are difficult to apply in real-time application domain. As a result, several methods have been proposed for FSM, one of which is the COSMIC FSM method [8,9]. COSMIC was designed to apply in various functional domains such as business application domain, real-time application domain. Some researchers proposed the embedded system with some modeling languages. But these aren't applied in the generation model by using UML metamodel of embedded system to estimate the functional size of software. To address this limitation, this paper proposes the generation model that is based on UML metamodel with COSMIC FSM concept. After that the mapping rules must be produced between the generation model and COSMIC. Finally, COSMIC calculated with the case study of "Cooker System" to estimate the size of software. The rest of the paper is organized as 5 sections. Section II presents the related work. Section III presents the generation method. Section IV provides the case study. Section V presents the conclusion.

2. Related Work

In [1], SYMONS, C. described the example that COSMIC concepts can be applied in any real-time software requirements to measure the functional size of real-time software to understand clearly any software engineer with alarm example. In [2], Soubra, H., et al. proposed the design of the FSM procedure based on the documentation of the mapping of the Simulink concepts to COSMIC concepts for the embedded real-time software system. The generation model by using UML metamodel can be defined with COSMIC concept. Then, the mapping rules which map the generation model to the COSMIC model define.

This paper proposes the mapping rules that can be used in different types of embedded software system and also proposes a light weight generation method of COSMIC FSM. The resulted software sizing

measurement can be used in many software industries to increase effort and productivity.

3. Generation Method

In Figure 1, the UML sequence notation is used to obtain the functional size of software as an input. After that, the generation model based on UML metamodel has to be translated into COSMIC concept by using the mapping rules. Finally, the result of function size of software is calculated by using COSMIC method.

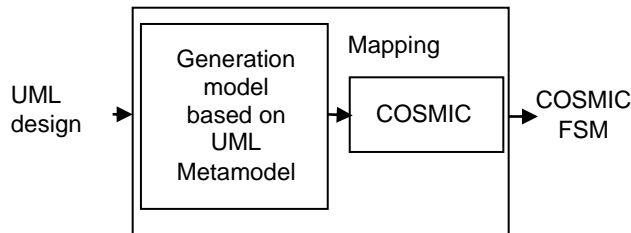


Figure 1. Flow of calculation (Processing)

Figure 2 shows the light generation method of COSMIC FSM based on generation model with three phases. In the generation phase, this paper defines generation model depending on UML meamodel that is based on the COSMIC concepts. In the mapping phase, we analyse the COSMIC concepts with the generation model to define the mapping rules. In the measurement phase, the results of the actual size measurement of the embedded system are calculated by using COSMIC method.

4. Case Study

4.1. Functional User Requirements

We adopt the specification of simple version of the Cooker system [10] to express the counting of COSMIC. After the specification of Cooker Software has defined, the UML representations can be developed. The basic functional requirements of this system are as follows:

1. When the power is switched on, the cooker software receives the input from the door and from a start button, and sends signals to switch an internal light, and the heater on or off. The software also sends signals to a timer to set the cooking time and can receive a signal from the timer when cooking is complete.
2. Cooking starts with pressing the start button provided the door is closed. If the door is open pressing the start button has no effect.
3. Opening the door during cooking turns the heater off.
4. The timer stops when the door is opened whilst cooking is in progress, or the timer signals that

cooking is completed and the timer resets itself to zero.

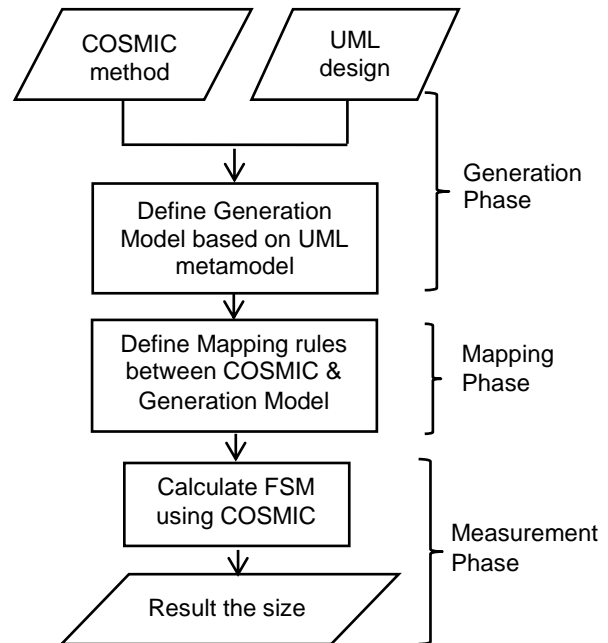


Figure 2. Calculation based on generation model

4.2. UML Model of a Case Study

The use case diagram of cooker system is shown in Figure 3. There are three actors in this system: timer, door sensor and start button that have relationship with four use cases of the system: DoorClosed, ButtonPushed, CookingEnded and DoorOpened by using the functional requirements.

4.3. UML Sequence model for COSMIC FSM

In this paper, the generation model define the classification of instance level of object depending on the partial UML sequence metamodel approach that allows reasoning about meta-elements and their relationships as shown in Figure 4. This generation model uses the UML metamodel through the profiling mechanism for COSMIC concepts. The sequence diagrams of cooker are used to apply the generation model as shown in Figure 5 to 12. The sequence diagram is appropriate to identify functional processes and data movements. Then, the number of data movements for each sequence diagram is calculated. Finally, the total size of system is calculated by aggregating all these data movements.

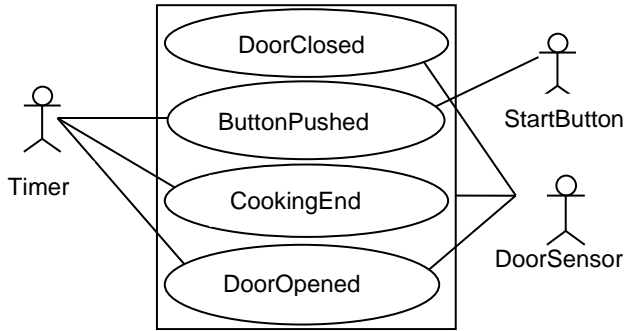


Figure 3. Use case diagram for cooker system

4.4. Mapping Rules

In this section, we define mapping rules that correspond to the COSMIC element and some instance level object of sequence diagram by using the UML metamodel concept through profiling mechanism. Table 1 summarizes the mapping rules from UML notations to COSMIC. Based on the correspondence, we define the mapping rules as follows:

Rule 1: Identification of the application boundary. The application border in Cooker system corresponds to UML use case diagram.

Rule 2: Identification of the functional users. COSMIC defines a (type of) user that is a sender and/or an intended recipient of data in Functional User Requirements of a piece of software. The concepts of COSMIC corresponds an objects in the sequence diagram.

Rule 3: Identification of the functional process. It requires the data from functional user that corresponds to interaction between objects that operate with one another in sequence diagram.

Rule 4: Identifying the data groups. A COSMIC data group corresponds to the data group that may be represented in sequence diagram by means of the flows of information between objects.

Rule 5: Identifying the four data movements. Sequence diagram represents these data movements. Each data movements correspond to an interaction messages in sequence diagram.

Rule 5.1: Identifying the Entry data movement. It moves the message from functional user to boundary.

Rule 5.2: Identifying the Exit data movement. It moves the message from boundary to functional user.

Rule 5.3: Identifying the Read data movement. It moves the single data group from persistent storage to functional process.

Rule 5.4: Identifying the Write data movement. It moves the single data group from functional process to persistent storage.

Rule 6: Applying the COSMIC measurement function. Each of the data movement (Entry, Exit, Read and Write) that identified in each functional process is added to obtain the functional size of that process.

Rule 7: Aggregation the functional size measurements.

Table 1. Mapping rules of COSMIC element and UML

COSMIC element	UML diagram
Boundary	Use Case
Functional User	Objects in Sequence Diagram
Functional Process	Interaction between objects
Data Group	Flows of information between objects
Entry Data Movement	Sequence message from Functional User to Functional Process
Exit Data Movement	Sequence message from Functional Process to Functional User
Read & Write Data Movement	Sequence message move single data group from persistent storage to a functional process

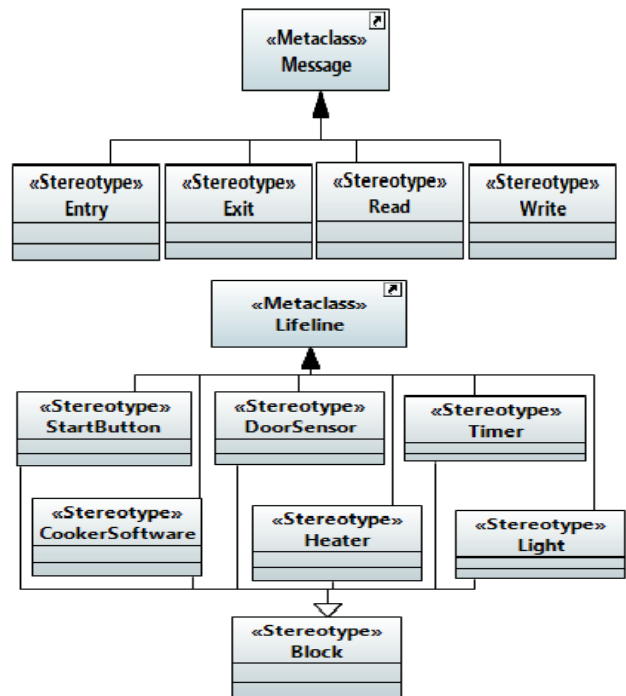


Figure 4. Generation model of classification of instance level by using UML metamodel

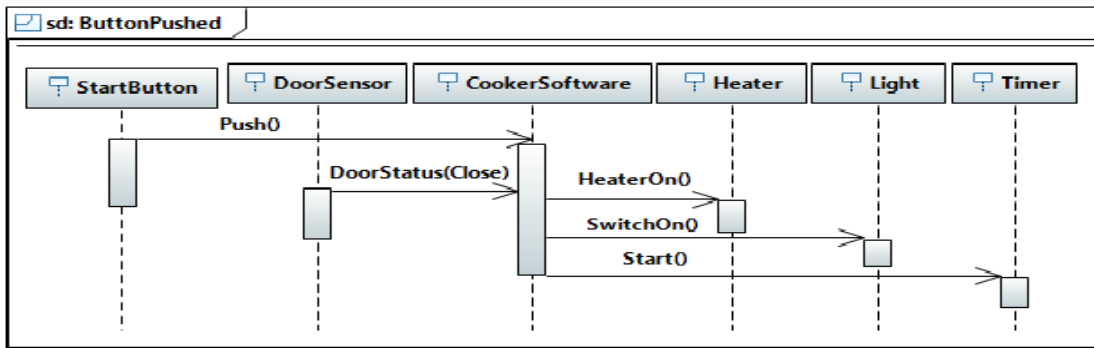


Figure 5. Sequence diagram for ButtonPushed

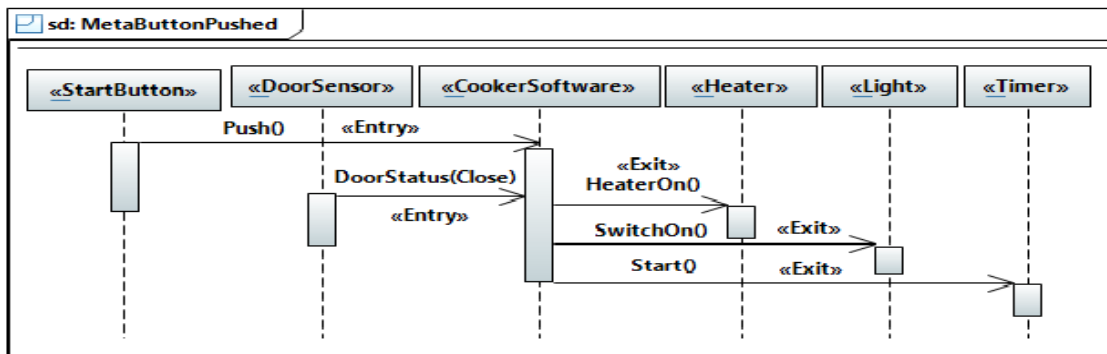


Figure 6. Sequence diagram for ButtonPushed using the generation model

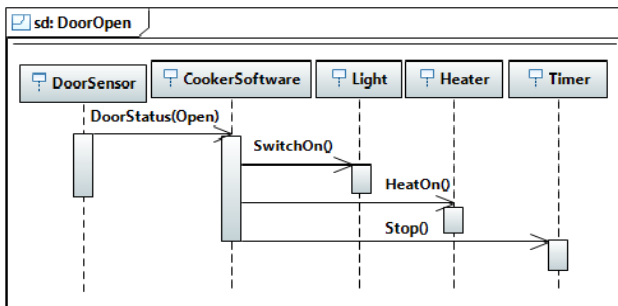


Figure 7. Sequence diagram for DoorOpen

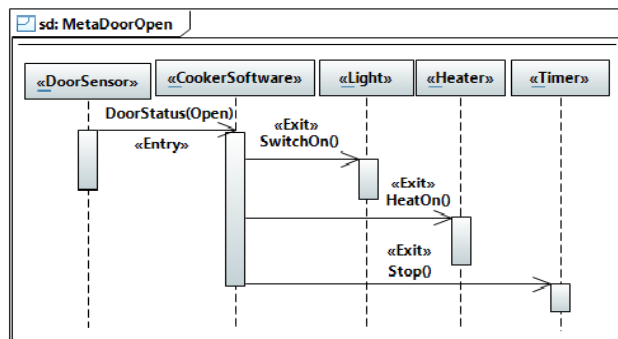


Figure 8. Sequence diagram for DoorOpen using the generation model

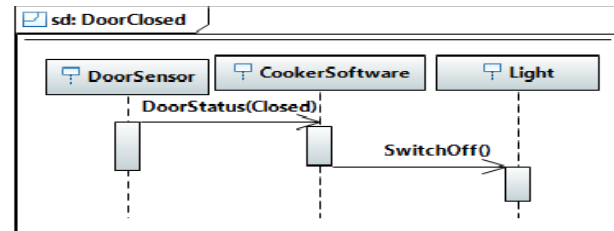


Figure 9. Sequence diagram for DoorClosed

All of the identified data movements of the functional processes of the whole system must be aggregated into a single functional size value by adding them together to obtain the functional size of the system.

4.5. COSMIC Data Movements

After defining the mapping phases, the data movements for each function can be calculated to estimate the size of software. These data movements are illustrated as shown in Figure 6,8,10 and 12. For the case study, the functional size identified 4 functional processes for DoorClosed, ButtonPushed, CookingEnd and DoorOpened.

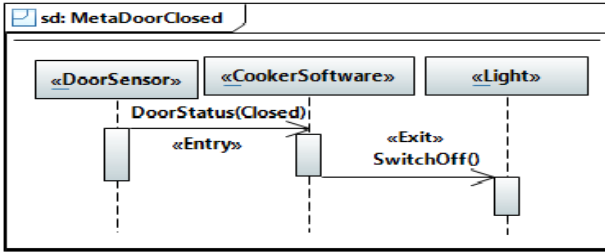


Figure 10. Sequence diagram for DoorClosed using the generation model

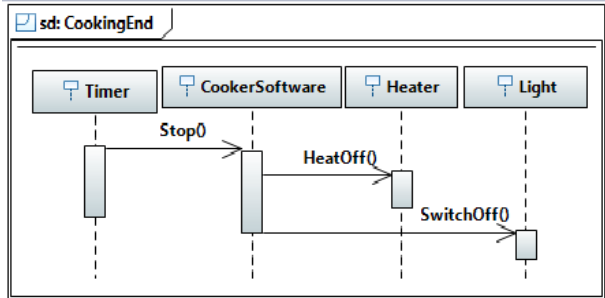


Figure 11. Sequence diagram for CookingEnd

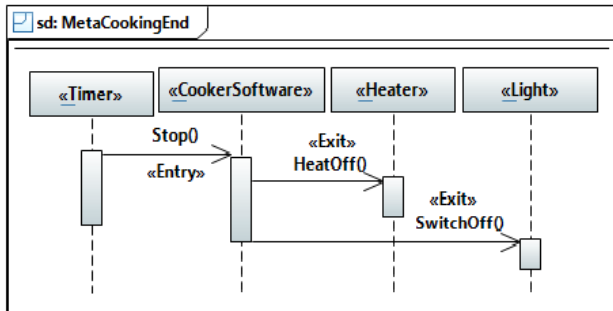


Figure 12. Sequence diagram for CookingEnd using the generation model

The COSMIC measurement standard, 1CFP is defined as the size of one data movement. In the functional process of ButtonPushed, it identified 2 Entry data movements. The Entry data movements counted the PushSignal attribute from start button and the Getdoorstatus from DoorSensor. It also identified 3 Exit data movements. The Exit data movements also counted the HeaterOn attribute to Heater, the LightOn attribute to Light and the Start attribute to Timer respectively. The subtotal of functional size for that functional process is 5CFP. By calculating the sizes from each function, the cooker system is estimated at 14CFP by adding all number of data. The data movement of each sequence diagram in this system is as shown in Table 2.

Table 2. Measurement of data movements for cooker system

Process	Message sending		Data Move_ments	CFP
	Message	Component of object involved		
Door Closed	DoorClsoe()	From Doorsensor	Entry	2 CFP
	Switchoff()	To Light		
Button Pushed	PushSignal()	From Start button	Entry	5 CFP
	Getdoorstatus (Close)	FromDoor Sensor	Entry	
	HeatOn()	To Heater	Exit	
	LightOn()	To Light	Exit	
	Start()	To Timer	Exit	
Cooking Ended	Stop()	From Timer	Entry	3 CFP
	Heatoff()	To Heater	Exit	
	Lightoff()	To Light	Exit	
Door Opened	Doorstatus (open)	From Doorsensor	Entry	4 CFP
	Lighton()	To Light	Exit	
	Heatoff()	To Heat	Exit	
	Stop()	To Timer	Exit	
Total				14 CFP

4.6. Automated Measurement

An automation tool will help in applying and using the COSMIC method in many industries. We develop the prototype tool for the automatic measurement of the functions as an example in Java. This paper uses the latest version of COSMIC measurement, version 4.0.1[9].

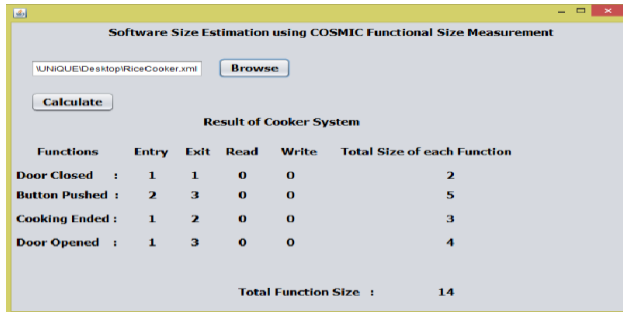
4.7. Result of Cooker System

We also develop the automated function size measurement from the requirements with cooker system. It is from the generation model based on UML metamodel. The measurement result of the cooker system is as shown in Figure 13. It gives the total function point number identified by adding the total number of Entries, Exits, Reads and Writes of the system. We have already applied our method by describing to another example. So, we apply our light weight generation method to both case studies successfully in generation model by using the SysML and UML metamodels.

4.8. Result of Elevator System

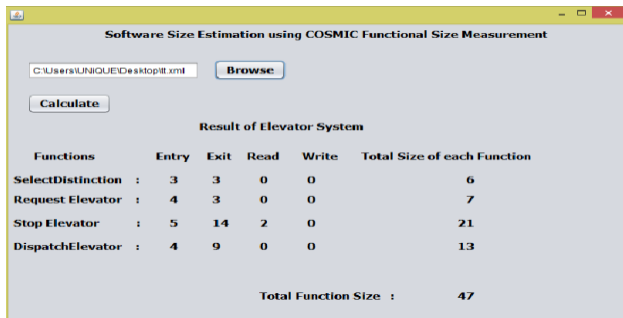
We have succeeded in automated calculation of the COSMIC measure with elevator system that is from the SysML generation model by using SysML metamodel [3].

Figure 14 shows our prototypes tool's interfaced and describes the measurement results of the elevator system.



Functions	Entry	Exit	Read	Write	Total Size of each Function
Door Closed :	1	1	0	0	2
Button Pushed :	2	3	0	0	5
Cooking Ended :	1	2	0	0	3
Door Opened :	1	3	0	0	4
Total Function Size :					14

Figure 13. Automated measurement results of Cooker System



Functions	Entry	Exit	Read	Write	Total Size of each Function
SelectDistinction :	3	3	0	0	6
Request Elevator :	4	3	0	0	7
Stop Elevator :	5	14	2	0	21
DispatchElevator :	4	9	0	0	13
Total Function Size :					47

Figure 14. Automated measurement results of Elevator System

5. Conclusion

Sizing with COSMIC is an excellent way of controlling the quality of the requirements. It improves estimating accuracy, especially for larger software projects. COSMIC FSM is a great importance in industrial software development, since it provides the necessary input to effort estimating models. In this paper, we showed the automated generation method of COSMIC FSM by using generation model based on UML metamodel. This generation method uses profiling mechanism and is also called the light weight generation method. We perform the case study of the cooker system which shows the usefulness of our light weighted approach.

I intend to develop the generation model depending on metamodel for basic modeling languages such as UML, SysML etc. which is useful for estimation the size of embedded system. After that, the general rules which allow mapping these generation model to COSMIC concepts must be defined. I hope to develop the automatic measurement of COSMIC size for different models designed in embedded system.

6. References

- [1] Symons, C.: "Sizing and Estimating for Real-time Software – the COSMIC-FFP method", In: DOD Software Tech News, Rome NY, vol. 9(3), 2006, pp. 5–11.
- [2] Soubra, H., Abran, A. , Stern, S. , Ramdan-Cherif, A., "Design of a Functional Size Measurement Procedure for Real-Time Embedded Software Requirements Expressed using the Simulink Model", Joint Conference of the 21st Int'l Workshop on and 6th Int'l Conference on Software Process and Product Measurement (IWSM-MENSURA), 2011.
- [3] Thandar Zaw, Myint Myint Lwin, Koichiro Ochimizu, Swe Zin Hlaing, "Software Size Estimation for Embedded Software using COSMIC FSM", Second Workshop on Advanced Technology, Myanmar, December 2016.
- [4] IFPUG (The International Function Point Users Group), Software Engineering- IFPUG 4.1 Unadjusted Functional Size Measurement Method- Counting Practices Manual, Switzerland,2009.
- [5]UKSMA(United Kingdom Software Metrics Association), Software Engineering-Mk II Function Point Analysis- Counting Practices Manual, Switzerland, 2002.
- [6]NESMA(Netherlands Software Metrics users Association), Function Size Measurement Method version 2.1 – Definitions and Counting Guidelines for the application of Function Point Analysis , Switzerland , 2005.
- [7] FiSMA (Finnish Software Measurement Association), Information technology - Software and systems engineering - FiSMA 1.1 functional size measurement method, Switzerland, 2008.
- [8] COSMIC (Common Software Measurement International Consortium), The COSMIC Functional Size Measurement Method Version 3.0.1: Measurement Manual, 2009.

[9] COSMIC, The COSMIC Functional Size Measurement Method version 4.0.1: Measurement Manual, 2015.

Sizing Real-time Real-Time Embedded Software, 2016.

[10] COSMIC, The COSMIC Functional Size Measurement Method, version 4.0.1: Guideline for

Mining Web Content Outliers by using Term Weighting Technique and Rank Correlation Coefficient Approach

Thinzar Tun, Khin Mo Mo Tun

University of Information Technology, Yangon, Myanmar

thinzartun@uit.edu.mm, khinmomotun@uit.edu.mm

Abstract

In the Internet area, World Wide Web (www) involves with voluminous amount of information with more redundant and irrelevant web pages. Outliers are the data that differ significantly from the rest of data. Web content mining is a subarea under web mining that mines required and useful knowledge or information from web page content. Web content outlier mining concentrates on finding outliers such as irrelevant and redundant pages from the web pages. Webs contain unstructured and semi-structured documents, so algorithms for web content mining are needed to handle both unstructured and semi structured documents. The proposed system based on big web data. The objective of proposed system is to obtain higher accurate result. In this proposal, Term Frequency Inverse Document Frequency (TF.IDF) technique based on full word matching with domain dictionary is used to remove the irrelevant documents from the unstructured web documents based on user's input query. Removal of outliers (irrelevant and redundant contents) from webs not only leads to reduction in indexing space and time complexity, but also improves the accuracy of search results. The documents that have very little similarity words from the user's input query are assumed as the web outliers. And then a mathematical approach called Spearman's rank correlation coefficient is used to remove the redundant web documents and to retrieve ranked relevant web documents.

Keywords- outliers, web content mining, term frequency, correlation coefficient

1. Introduction

With the exponential growth of information available on the internet, updating incoming data and retrieving relevant information from the web quickly and efficiently is a growing concern. Most of the web search engines typically employ conventional information retrieval and data mining techniques to discover automatically useful and previously unknown information from web. With the enormous growth on the web, users get easily lost in the rich hyper structure.

In addition, as most of the data in the web is unstructured, and contains a mix of text, video, audio etc. There is a need to mine information to cater to the specific needs of the users. Web mining is an emerging research area focused on resolving these problems [1].

Web mining is the application of data mining techniques to automatically discover useful and previously unknown information from the web documents. Web Mining has adapted techniques from the field of data mining, databases and information retrieval. In general, web mining tasks can be classified into three major categories: web structure mining, web usage mining and web content mining. Web structure mining is the discovery of interesting patterns from the hyperlink structure of the web. Web usage mining mines secondary information extracted from user interactions with the web while surfing. Web content mining aims to extract useful information from the web pages based on their contents. So similar pages can be grouped together to enhance performance. web content mining aim at summarizing information on web pages to facilitate efficient and effective information retrieval. [5].

Outliers are observations that deviate so much from other observations to arouse suspicions that they might have been generated using a different mechanism. Outliers may also reflect the true properties of data from rare and interesting events which may contain more valuable information than normal data. Outlier mining is dedicated to finding data objects which differ significantly from the rest of data. Traditional outlier mining techniques can easily detect outliers that present in numeric datasets, but it becomes extremely difficult to detect outliers which are in web dataset. Web outliers are data that present in web which has different characteristics from the web data taken from the same category. Different contents of the web pages from the category in which they were taken constitute web content outliers. Web content outliers mining concentrates on discovering outliers from the web contents of a web page [3].

2. Theoretical Background

The n-gram based and word based techniques are useable in the preprocessing part of mining web content outlier. Word based systems applies different techniques

than the n-gram based systems. Besides applying full word matching, the domain dictionary was indexed based on the length of word in order to enhance term searching quality. The word based technique just maintains the size of the words. Although the words are in variable length, the efficiency of word based web content outlier mining can be increased by indexing the words in two dimensional format (i, j) and indexing the domain dictionary based on length of the word. The organized domain dictionary ensured that the memory space, search time and run time for checking the relevancy of the web documents gets reduced [8].

The Term Frequency. Inverse Document Frequency (TF.IDF) is a weighting method often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word for a document in a collection or corpus. The TF.IDF value increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general. Variations of the TF.IDF weighting method are often used by search engines as a central tool in scoring and ranking a document's relevance given a user's input query [9].

In statistics, Spearman's rank correlation is a nonparametric measure of rank correlation. It assesses how well the relationship between two variables can be described using a monotonic function. The Spearman correlation between two variables will be high when observation has a similar. Spearman's coefficient is appropriate for both continuous and discrete variables including ordinal variables [10].

3. Related Works

The same authors use an n-gram method with domain dictionary and without domain dictionary in [4] and [5] to determine the similarity of strings and expand it to include pages containing similar strings. The experimental results show that finding outliers with high order n-grams (5-grams) perform better than lower order n-grams. The existing approach using WCOND Mine algorithm based on n-grams that works only for structured documents. The n-gram based systems become slowly for very large datasets because of the huge number of n-gram vectors generated during mining web content outliers. The word-based techniques just maintain the size of the words. Although the words are in variable length, the efficiency of word based web content outlier mining can be increased by indexing the words in two-dimensional format (i,j) and indexing the domain dictionary based on length of the word. The organized domain dictionary ensured that the memory space search time and run time for checking the

relevancy of the web documents gets reduced. The n-gram based system takes a longer time to complete a task than the word-based systems even though the size of data is not too large. A traditional weighting technique TF.IDF (Term Frequency * Inverse Document Frequency) from information retrieval is only compatible to use in detection web outliers; it even returns better results than previous works. But it cannot remove redundant web pages if they exist [8].

The author S.Poonkuzhali proposed a signed with weight technique based on full word matching for structured and unstructured documents to retrieve relevant document and linear correlation is used to remove duplicates [2]. A mathematical approach called Spearman's correlation coefficient is used to calculate the correlation between the document pairs to remove redundant web pages. This method depends on the term frequency of common words between document pairs that is ranked based on the frequency value. This method gives better performance than linear correlation and ranking correlation [6]. In the proposed system, Term Frequency Inverse Document Frequency (TF.IDF) technique based on full word matching with domain dictionary is used to mine and remove irrelevant web pages and Spearman's correlation coefficient is applied to eliminate redundant web pages.

4. Architecture Design

4.1. Extracted web pages

The document extraction is the process of retrieving the desired pages belonging to the category of interest. The documents are retrieved by search Engine based on the user's input query. Most of retrieved documents may or may not relevant to the user query [7].

4.2. Preprocessing

The extracted documents undergo the preprocessing step which consists of stop words removal, stemming and tokenization. Preprocessing is necessary to make the entire document in the same format. Stop words list typically consists of those word classes known to convey little substantive meaning such as articles (a, the), conjunctions (and, but), interjections (oh, but), prepositions (in, over), pronouns (he, it) and forms of the "to be" verb (is, are).

Stemming removes word suffixes which reduce the number of unique words in the index by reducing the storage space required for the index and speeds up the search process.

Tokenization is the process of breaking a stream of text into words, phrases, symbols, or other meaningful elements called tokens. A token is a string of characters,

categorized according to the rules as a symbol. The list of tokens becomes input for further processing. [7].

4.3. Generate Full Word Profile

The filtered datasets from preprocessing stage are then used to generate the full word profile. At this time, the domain dictionary has been indexed- based on the length of the word. The full word profile for the document is generated in matrix form (i.e., $W_{1,4}$ represents 4th word in 1st page). Then the j^{th} word from i^{th} page is taken and its length is calculated (i.e., $|W_{ji}|$) and depending on the number of characters, the respective index on domain dictionary is searched. If the words exist in both sides, it will be flagged as 1, otherwise 0 will be returned. Then the word frequency will be counted. The full word profile generated by indexing all word with two-dimensional format (i, j) where 'i' represent web pages, 'j' represent words and every word attached with word frequency, word length and the binary number which mentioned either it exists in domain dictionary or not [1].

4.4. Compute Relevancy with TF.IDF

In the weighting computation, a classic term weighting technique, TF.IDF from Information Retrieval (IR) was adopted to evaluate the representativeness of terms in the web content. The dissimilarity measure computed to determine the difference among pages within the same category. The Maximum Frequency Normalization applied to Term Frequency (TF) weighting because when the document length varies, the relative frequency is preferred. Since term frequency alone may not have the discriminating power to pick up all relevant documents from other irrelevant documents, an IDF (Inverse Document Frequency) factor which takes the collection distribution into account has been proposed to help to improve the performance of IR.

The dissimilarity measure will only compute the words that exist in the dictionary because the formula returns only a binary value. Then the words that did not exist in the domain dictionary will not be computed. The reason is the word that exists in the dictionary is more relevant to the domain category and it represents the power of the document. The outliers come out with the lowest frequency of word that exists in the dictionary and there will be only a few words that exist in the domain dictionary. Therefore, the dissimilarity measures will return a higher dissimilarity value than other web pages [8].

The dissimilarity equation is below:

$$DM_i = \frac{\sum_{i,j} [(0.5 + \frac{0.5 * f(t_j, e_i)}{MaxFreq(d_i)}) (\log_{10} \frac{N}{k})]}{e_i}$$

where $f(t_j, e_i)$ denotes the frequency of term t_j present in the document d_i in the domain dictionary, while $MaxFreq(d_i)$ determine maximum frequency of a word in a document, N is the total number of documents and k is the number of documents with term t_j appears.

4.5. Determination irrelevant documents

The output from the dissimilarity measure was ranked to determine outliers or irrelevant documents. The documents at the top will have high dissimilarity measure deviates more from the category of user interest. Also, the documents at the bottom will have less dissimilarity measure which is more relevant to the category of interest. So, the top 'n' documents that have high dissimilarity measure are declared as outliers based on threshold value.

4.6. Compute redundancy with Spearman's correlation coefficient

In Spearman's correlation coefficient method, frequency of all the terms in the document is calculated. Then the scoring or ranking should be made for each term based on the number of times it occurred in the document. The term having highest frequency should be ranked 1, similarity for other terms. If the term W_k is present in document D_i and not in D_j the rank of the term W_k for the document D_j will be zero. Next step is to compare all the document pairs to check for redundancy. The mathematical concept called correlation coefficient has been applied in this work to find out the redundant documents. Spearman's rank correlation coefficient equation is below:

$$\rho = |1 - \frac{6 \sum d^2}{n(n^2 - 1)}|$$

Where ρ is correlation value, d is given by $(x_i - y_i)$ where x_i and y_i are frequency of the term i in document D_p and D_q respectively has been used. n is total number of words in document D_p and D_q . Always the ρ value lies between 0 and 1. If the ρ value is 1 for document D_p and D_q then D_q is the redundant of D_p . If there is no common word between the two documents D_p and D_q then the ρ value will be 0 [6]. And then remove all redundant web documents.

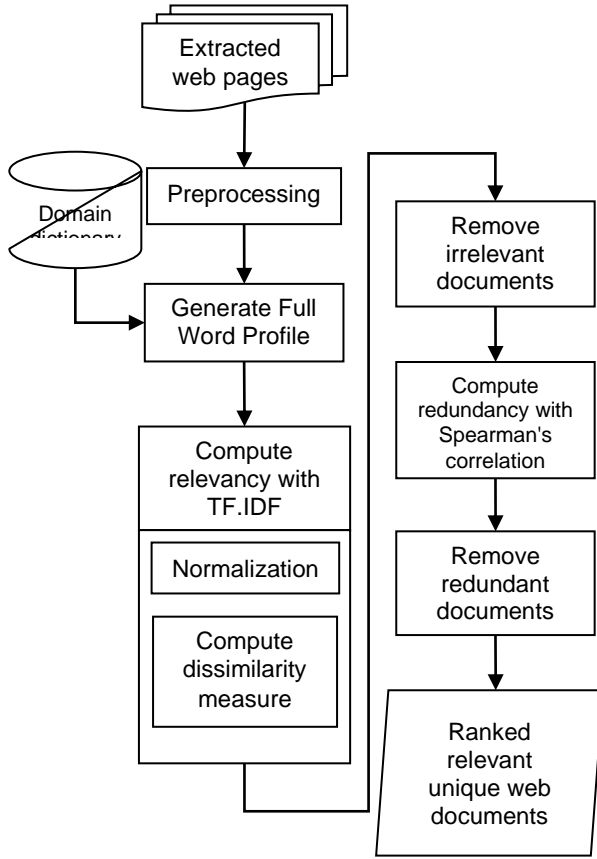


Figure 1. Architecture design of proposed system

5. Proposed Algorithm and Experimental Results

Input: Domain Dictionary and Web Documents d_i

Output: Ranked relevant unique web documents

1. Extract the set of documents
2. Preprocess the entire extracted documents by removing stop words, stemming and tokenization
3. Generate full word profile
4. Generate organized domain dictionary

//relevancy computation with TF.IDF

5. For (int i=0; i<NoOfDoc; i++) {
6. For (int j=1; j<=NoOfWords; j++) {
7. If (j exists in the domain dictionary) {
8. Compute dissimilarity measure (DM_i)

$$DM_i = \frac{\sum_{i,j} \left[0.5 + \frac{0.5 * f(t_i, e_i)}{MaxFreq(d_i)} \right] (\log_{10} \frac{N}{k})}{e_i}$$

- 9.}}

10. $DM_i = DM_i /$ number of words in the document that exist in the domain dictionary.
11. Rank the result of DM_i
12. Determine irrelevant documents and remove it // redundant computation with Spearman correlation coefficient method
13. Find the term frequency TF (W_{ik}) for all the words W_k in the given query for each document d_i where $1 \leq k \leq m$, m is the number of words in document d_i .
14. Form a $n \times m$ matrix where n is the number of words in given query and m is the number of retrieved documents.
15. Assign the term frequency ranking TFR (W_{ik}) to each words W_k in document d_i where $1 \leq k \leq m$. m is the number of words in document d_j .
16. Assign the term frequency ranking TFR (W_{ik}) to each words W_k in document d_i where $1 \leq k \leq m$. m is the number of words in document d_j .
17. For each document pair, perform the Spearman's rank correlation coefficient

$$\rho = \left| 1 - \frac{6 \sum d^2}{n(n^2 - 1)} \right|$$

18. If the ρ value is 1 then d_j is redundant document, else d_j is not a redundant.
19. Remove redundant documents

An analysis has been made with the proposed system and the existing methods. A case study has been tested with the dataset that consists 200 web pages from Science medical folder provided by the 20 Newsgroup datasets. There is no benchmark data for testing web content outliers, so embedded motive is the only way to know if the outliers returned are actually real outliers (irrelevant and redundant contents). The outliers usually constitute less than 10% of the entire dataset [8]. So, 20 web pages from the Course folder of University Cornell, provided by World Wide Knowledge Base (WEBKB) to detect outliers or irrelevant documents. Then documents are retrieved and processed with TF.IDF method to remove irrelevant documents. The results are ranked and top 20 web documents are defined as outliers or irrelevant documents. And another web documents are declared as relevant documents.

Next, Spearman correlation coefficient method is calculated for each of document pairs from relevant retrieved documents. Finally, the document having coefficient value 1 is defined as redundant document and removed it. It shows that the proposed method generates high F-measure and accuracy compared with the existing methods. Precision is the fraction of retrieved documents that are relevant to the query. Recall is the fraction of the relevant documents that are

successfully retrieved. F1-Measure is the harmonic mean of precision and recall. F1-Measure reaches its best value at 1 and worst value at 0. Accuracy is the measure which matches the actual value of the quantity being measured. The F1-Measure and accuracy results are shown in below:

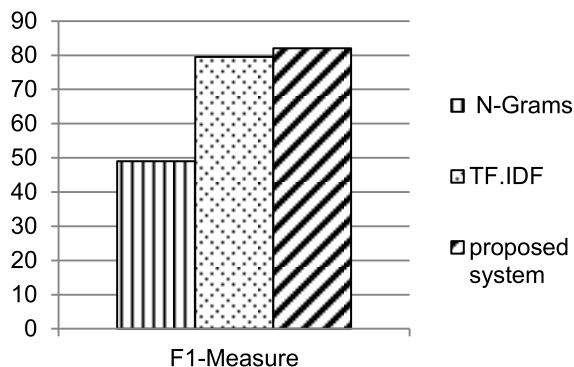


Figure 2. Results on F1-Measure

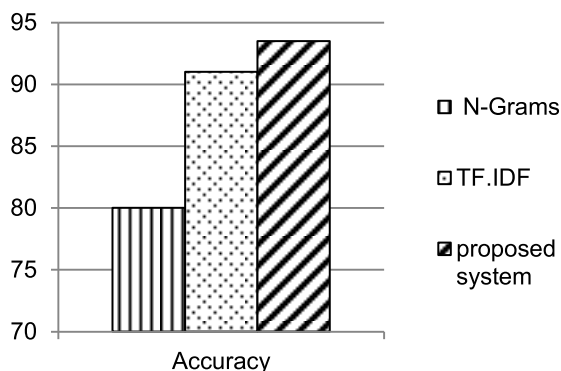


Figure 3. Results on accuracy

6. Conclusion

The massive growth of internet and World Wide Web encourages developing the automated tools to retrieve relevant sources quickly without duplicates. The key feature of the proposed system is to improve accuracy result. In the proposed system, the traditional term weighting technique TF.IDF based on full word matching with domain dictionary is used to remove irrelevant web documents. And Spearman's correlation coefficient method is used to calculate the correlation

between the document pairs to eliminate redundant web documents.

7. References

- [1] G. Poonkuzhali, K. Thiagarajan, K. Sarukesi, and G.V. Uma, "Signed approach for mining web content outliers," Proceedings of World Academy of Science, Engineering and Technology, Vol. 56, pp -820-824, 2009.
- [2] G. Poonkuzhali, K. Sarukesi, and G.V. Uma, "Web content outlier mining through mathematical approach and trust rating," 10th WSEAS International Conference on Applied Computer and Applied Computational Science (ACACOS '11), 2011.
- [3] K.Sarukesi, P.Sudhakar, S. Poonkuzhali, "Signed-With-Weight Technique for Mining Web Content Outliers", Special Issue of International Journal of Computer Applications (0975 – 8887) the International Conference on Communication, Computing and Information Technology (ICCCMIT) 2012.
- [4] M. Agyemang, K. Barker, and R.S. Alhaji, "Mining web content outliers using structure oriented weighting techniques and n-grams," Proceedings of ACM SAC, New Mexico, 2005.
- [5] M. Agyemang, K. Barker, and R.S. Alhaji, "WCOND-Mine: Algorithm for Detecting Web Content Outliers from Web Documents," Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC), 2005.
- [6] S. SATHYA BAMA, M.S. IRFAN AHMED, A. SARAVANAN, "A Mathematical Approach for Mining Web Content Outliers using Term Frequency Ranking", Indian Journal of Science and Technology, Vol 8(14).
- [7] S. SATHYA BAMA, M.S. IRFAN AHMED, A. SARAVANAN, "A Mathematical Approach for Improving the Performance of The Search Engine Through Web Content Mining", Journal Theoretical and Applied Information Technology, 20th February 2014, Vol.60, No2.
- [8] W.R.W. Zulkifeli, N. Mustapha, A. Mustapha, "Classic Term Weighting Technique for Mining Web Content Outliers", International Conference on Computational Techniques and Artificial Intelligence (ICCTAI'2012). Penang, Malaysia, 2012.
- [9] <https://en.wikipedia.org/wiki/Tf-idf>
- [10] https://en.wikipedia.org/wiki/Spearman's_rank_correlation_coefficient

Image and Signal Processing

Lane Detection System based on Hough Transform with Retinex Algorithm

Shwe Yee Win, Htar Htar Lwin
University of Information Technology
shweyeewin@uit.edu.mm, htarhtarlwin@uit.edu.mm

Abstract

Nowadays, automotive system becomes a great innovation in the world and lane detection system is important to control automobile vehicles. This paper has developed an efficient lane detection system to deal with different types of lighting conditions. Six types of edge detection techniques: canny, sobel, prewitt, Roberts, Laplacian of Gaussian (LOG) and zero-cross methods are analyzed. Line detection based on canny operator is developed. Moreover, Retinex algorithm is employed to normalize input images for all types of illumination. And Hough Transform with Retinex algorithm is developed to solve lighting problem. The proposed method is compared to Hough Transform with Otsu's threshold method. The experimental results show that the proposed method can reduce computation time and improve accuracy for lane detection system.

Keywords- Automotive System, Lane Detection, Hough Transform, Retinex

1. Introduction

Lane detection is the process to locate lane markers on the road and then describe these locations to an intelligent system. In intelligent transportation systems [4], intelligent vehicles cooperate with smart infrastructure to achieve a safer environment and better traffic conditions. The applications of a lane detecting system could be as simple as pointing out lane locations to the driver on an external display, to more complex tasks such as predicting a lane change in the instant future in order to avoid collisions with other vehicles. Some of the interfaces used to detect lanes include cameras, laser range images, and GPS devices [7].

In safety driving assistance system, the correct recognition of lane detection is the most important issue for automobile vehicles to achieve autonomous navigation. The level of autonomy ranges from fully autonomous (unmanned) vehicles to vehicles where computer vision based systems support a driver or a pilot in various situations. Fully autonomous vehicles typically use computer vision for navigation, i.e., for producing a map of its environment and for detecting obstacles.

Researchers have developed various lane detection methods based on computer vision. These detection methods can be divided into two types: model-based and

feature-based methods [6, 8, and 10]. In model-based methods, lane boundaries are presented by mathematical models while feature-based methods use segmentation methods to locate road areas. Moreover, model-based methods usually require a very complex modeling process involving much prior knowledge and background. Among model-based and feature-based methods, feature-based algorithms are efficient and popular.

For autonomous vehicle, detection of lane suffers from high computational complexities and poor performance under different lighting conditions. In this paper, we propose Hough Transform with Retinex Algorithm for lane detection to solve lighting problem and compare computational complexity to Hough Transform with Otsu's method.

The remainder of this paper is organized as follows: Section 2 provides review of researches in the literature related to lane detection in automotive systems. In Section 3, our proposed methodology for lane detection is described to solve different lighting conditions. Section 4 presents analysis of results of proposed algorithms. Finally, section 5 describes conclusion and future work.

2. Research Background

With the development of researches on autonomous vehicle, lane detection is becoming a more and more hot topic. Among lane marking detection algorithms, feature-based algorithms are efficient and popular, where many researchers have been done.

Jie Guo et al. [3] proposed an improved random sample consensus (RANSAC) algorithm combined with the least squares technique to estimate lane model parameters based on feature extraction. They achieved comparable results to other algorithms that only worked on detecting the current lane boundaries. However, from experimental results, there are still some difficult lane scenarios to be solved. Moreover, they presented future work that their framework will be integrated with tracking algorithms for improvement.

Hao Yu et al. [1] described a constraint between Sobel operator and Shen Jun edge operator to detect the lane marking points. In addition to, they presented a new vehicle detect algorithm which uses the shadow under the vehicle and the vertical edge to detect the candidate vehicles. Then they used Support Vector Machine (SVM)

and Histogram of Oriented Gradients (HOG) to verify their system.

Dong-Uk Kim et al. [5] proposed an efficient approach to lane and pedestrian detection by processing sequential images from a camera attached to a moving vehicle. They predicted the left and right lines of the current lane by finding high intensity pixels along multiple horizontal scan lines and connecting the detected pixel points.

Z. Teng et al. [9] proposed an algorithm which integrated multiple cues, including bar filter which has been efficient to detect bar-shape objects like road lane, color cue, and Hough Transform. To guarantee the robust and real-time lane detection, particle filtering technique has been utilized. This algorithm improved the accuracy of the lane detection in both straight and curved roads. It has been effective on a wide variety of challenging road environments. This method fails for the lane tracking when it is to be applied to particle filter in the dashed lane situation.

F. Mariut et.al [2] proposed an algorithm that automatically emphasizes the lane marks and recognizes them from digital images, by the use of Hough transform. This method also detects lane mark's characteristics and has the ability to determine the travelling direction. A technique that extracts the inner margin of the lane is used to ensure the right detection of the lane mark. The algorithm works very efficiently for straight roads but fails in some cases of curved roads.

3. Methodology

Our proposed methodology for lane detection to solve lighting problem is shown in Figure.1. This system includes four stages: pre-processing, edge detection, illumination reduction and line detection. In pre-processing step, three sub-stages are described: selection of ROI (Region of Interest), conversion RGB image into gray scale image and noise removal. The major purpose of our system is to detect lanes under different illumination conditions. And another purpose is to improve performance and processing time of lane detection system.

3.1. Preprocessing

This is the most important step in our system. First, we resize input image (as shown in Figure 2) and set ROI (Region of Interest) to reduce memory storage space. Therefore, only region of image which contains relevant information (i.e. lanes' boundaries) is obtained. Next, RGB color image is converted into grayscale image in order to improve processing speed as shown in Figure 3. The last step of preprocessing stage is de-noising. In this case, we apply Gaussian filter to remove noise.

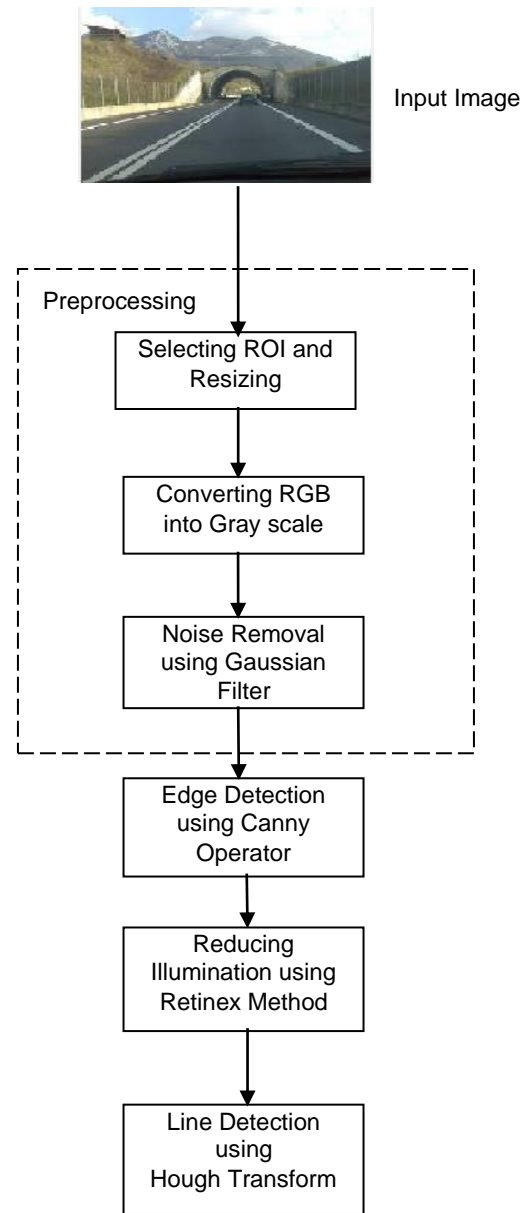


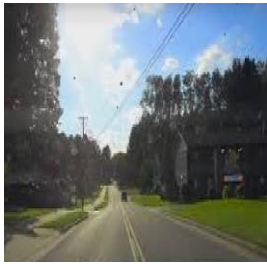
Figure 1. Proposed lane detection system

3.2. Gaussian filter

It is a type of filter which is applied to convolve with the image where the selection of the size of Gaussian kernel will affect the performance of the detector. Typically, it is only needed to calculate a matrix with dimensions $\lceil 6\sigma \rceil \times \lceil 6\sigma \rceil$ (where $\lceil . \rceil$ is the ceiling function) to ensure a result sufficiently close to that obtained by the entire Gaussian distribution.



Figure 2. Original image



(a) Resized image



(b) Grayscale image

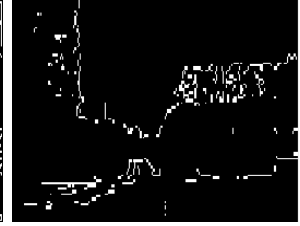


(c) Noise removal image using gaussian filter

Figure 3. Preprocessing steps of lane detection

3.3. Edge detection

Next, edge detection is performed. Edge information is the most commonly used features in lane detection system. In real-time situations, lane edge features may not be strong and may be affected by shadow. Therefore, the selection of edge detection operator is needed. In this case, sobel operator, canny operator, prewitt operator, Roberts, Laplacian of Gaussian and Zero-cross are individually experimented as edge detection algorithms as shown in Figure 4. From our experiments, canny operator is the most suitable one for next steps of lane detection system. Thus, this operator is accepted as edge detection algorithm for our proposed system.



(b) Sobel operator



(c) Prewitt operator



(d) Roberts operator



(e) Laplacian of gaussian



(f) Zero-cross method

Figure 4. Edge detection methods

3.4. Canny operator

The process of canny edge detection algorithm can be broken down to 5 different steps:

- (1) Apply Gaussian filter to smooth the image in order to remove the noise
- (2) Find the intensity gradients of the image
- (3) Apply non-maximum suppression to get rid of spurious response to edge detection
- (4) Apply double threshold to determine potential edges
- (5) Track edge hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

3.5. Reducing illumination using retinex algorithm

Lighting problem is crucial in Lane Detection System (LDS). In this case, Retinex Algorithm is applied for all

types of illumination. The Retinex theory motivated by Land [11] is based on the physical imaging model, in which an image $I(x, y)$, it could achieve sharpening with compensation for the blurring introduced by image formation process. Moreover, it could improve consistency of output as illumination changes.

$I(x, y)$ is regarded as the product $I(x, y) = R(x, y) \cdot L(x, y)$ where $R(x, y)$ is the reflectance and $L(x, y)$ is the illumination at each pixel (x, y) . Here, the nature of $L(x, y)$ is determined by the illumination source, whereas $R(x, y)$ is determined by the characteristics of the imaged objects. Therefore, the illumination normalization can be achieved by estimating the illumination L and then dividing the image I by it. In most Retinex methods, the reflectance R is estimated as the ratio of the image I and its smooth version which serves as the estimate of the illumination L .

$$R_i(x, y) = \log I_i(x, y) - \log [F(x, y) * I_i(x, y)] \quad (1)$$

$$R_i(x, y) = \log \frac{I_i(x, y)}{F(x, y) * I_i(x, y)} = \log \frac{I_i(x, y)}{\bar{I}_i(x, y)} \quad (2)$$

where $I_i(x, y)$ is the image distribution in the i^{th} spectral band and $R_i(x, y)$ is retinex output.

Gaussian function $F(x, y) = K e^{-(x^2+y^2)/c^2}$ where K is determined by

$$\iint F(x, y) dx dy = 1 \quad (3)$$

3.6. Line detection using generalized hough transform

In the final stage of our proposed system, Generalized Hough Transform is applied for lane detection.

To generalize the Hough algorithm to non-analytic curves, Ballard defines the following parameters for a generalized shape $a = \{y, s, \theta\}$ where y is a reference origin for the shape, θ is its orientation, and $s = (s_x, s_y)$ describes two orthogonal scale factors. As in the case of initial Hough Transforms, there is an algorithm for computing the best set of parameters for a given shape from edge pixel data. These parameters no longer have equal status. The reference origin location, y , is described in terms of a template table called the R table (as shown in table 1) of possible edge pixel orientations.

Table 1. Building R-table

i	ϕ_i	$R \phi_i$
1	0	$(Y_{11}, \alpha_{11})(Y_{12}, \alpha_{12}) \dots (Y_{1n}, \alpha_{1n})$
2	$\Delta \phi$	$(Y_{21}, \alpha_{21})(Y_{22}, \alpha_{22}) \dots (Y_{2m}, \alpha_{2m})$
3	$2\Delta \phi$	$(Y_{31}, \alpha_{31})(Y_{32}, \alpha_{32}) \dots (Y_{3k}, \alpha_{3k})$
...

The computation of the additional parameters s and θ is then accomplished by straightforward transformations to this table. The key to generalizing the Hough algorithm to arbitrary shapes is the use of directional information. Given any shape and a fixed reference point on it, instead of a parametric curve, the information provided by the boundary pixels is stored in the form of the R-table in the transform stage.

For every edge point on the test image, the properties of the point are looked up on the R-table and reference point is retrieved and the appropriate cell in a matrix called the Accumulator matrix is incremented. The cell with maximum 'votes' in the accumulator matrix can be a possible point of existence of fixed reference of the object in the test image.

Choose a reference point y for the shape (typically chosen inside the shape). For each boundary point x , compute $\phi(x)$, the gradient direction and $r = y - x$ as shown in the image. Store r as a function of ϕ . Notice that each index of ϕ may have many values of r .

One can either store the co-ordinate differences between the fixed reference and the edge point $((x_c - x_{ij}), (y_c - y_{ij}))$ or as the radial distance and the angle between them (r_{ij}, α_{ij}) . Having done this for each point, the R-table will fully represent the template object. Also, since the generation phase is invertible, we may use it to localise object occurrences at other places in the image.

4. Experiments and results analysis

We have evaluated the proposed algorithms with a laptop, in Matlab environment which has Intel(R) Core (TM) i3 CPU @ 2.53 GHz and 2.00GB RAM. Since Caltech's 2008 are available as public database, we have implemented our experiments on it. Four stages are presented in our proposed system: pre-processing, edge detection, illumination reduction and lane detection as shown in Figure 1.

Table 2. Computation time of edge detection algorithms

No.	Edge Detection Methods	Time (seconds)
1	Canny	0.623103
2	Sobel	0.608996
3	Prewitt	0.551883
4	Roberts	0.520003
5	Laplacian of Gaussian	0.660884
6	Zero-Cross	0.543144

The goal of our lane detection system is to solve lighting problem for all types of illumination. In this paper, we proposed an efficient lane detection system based on retinex algorithm which solves different lighting

conditions. Region of Interest (ROI) selection, grayscale conversion and removal of noise are applied in preprocessing stage.

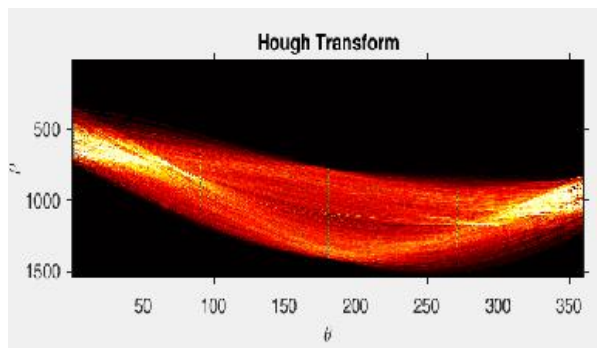


Figure 5. Hough transform using otsu's method

In edge detection, canny operator is the most suitable edge detector for our proposed system. Next, we have tested Otsu's threshold method and retinex method for lighting problem as shown in Figure 5 and 6. In this case, our proposed method gives better computation performance in short time as shown in Table 3.

Table 3. Computation time of illumination methods

Illumination Methods	1	2	3	4	5	6	7	8	9
Otsu's Method	2.8 (s)	0.5 (s)	1.2 (s)	0.7 (s)	0.8 (s)	0.9 (s)	0.9 (s)	1.4 (s)	0.8 (s)
Proposed Method	1.4 (s)	0.3 (s)	0.6 (s)	0.3 (s)	0.8 (s)	0.6 (s)	0.8 (s)	0.7 (s)	1.2 (s)

Moreover, lane detection using Hough transform and edge detection methods (canny, sobel, prewitt, Roberts, LoG, zero-cross and Otsu's threshold) have been tested individually and results are shown in Figure 7.

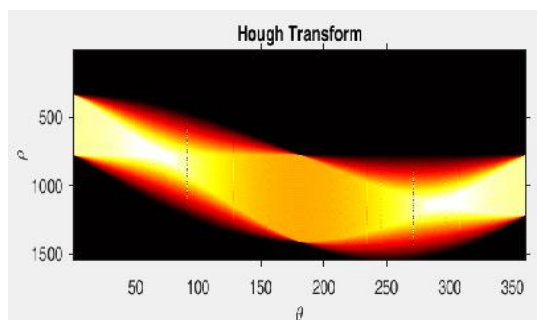


Figure 6. Hough transform using retinex method

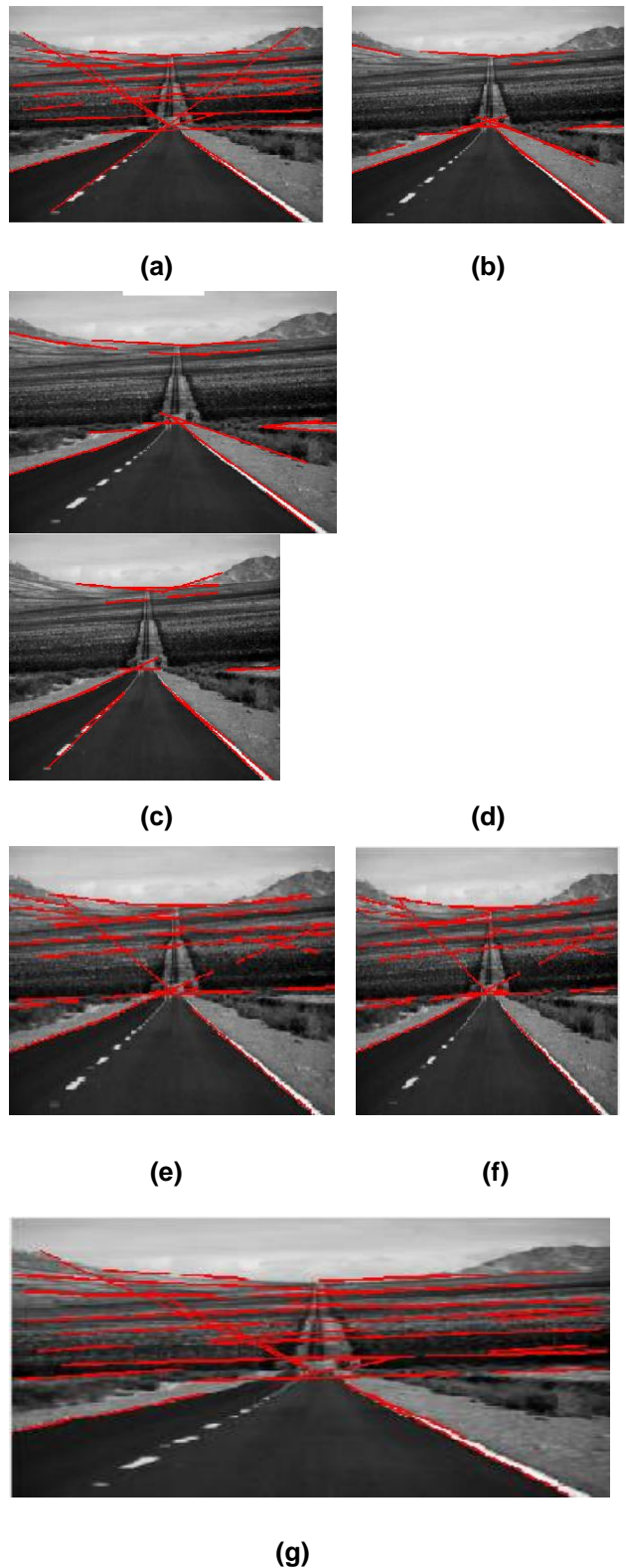


Figure 7. Lane detection using (a) canny, (b) sobel, (c) prewitt, (d) roberts, (e) laplacian of gaussian, (f) zero-cross and (g) otsu's threshold methods

5. Conclusion and future work

In real-time situations, lane detection system faces lighting problem for various types of illumination. Therefore, in this paper, we have especially presented Hough Transform with retinex algorithm for line detection. Moreover, six types of edge detection methods are individually experimented and canny operator is selected as the most suitable edge detection algorithm.

In future work, an efficient method based on Hough Transform and Retinex Method, which is to solve both straight lane detection and curve line detection, will be presented.

6. References

- [1] Hao Yu, Yule Yuan, Yueting Guo, Yong Zhao, "Vision-based Lane Marking Detection and Moving Vehicle Detection", *8th International Conference on Intelligent Human-Machine Systems and Cybernetics*, IEEE, 2016.
- [2] F. Mariut, C. Fosala and D. Petrisor, "Lane Mark Detection Using Hough Transform", *International Conference and Exposition on Electrical and Power Engineering*, IEEE, pp. 871 - 875, 2012.
- [3] Jie Guo, Zhihua Wei, Duoqian Miao, "Lane Detection Method Based on Improved RANSAC Algorithm", *Twelfth International Symposium on Autonomous Decentralized Systems*, IEEE, 2015.
- [4] S. Srivastava, R. Singal and M. Lumb, "Efficient Lane Detection Algorithm using Different Filtering Techniques", *International Journal of Computer Applications*, pp. 975-8887, 2014.
- [5] Dong-Uk Kim, Sung-Ho Park, Jong-Hee Ban, Taek-Min Lee, Yongtae Do, "Vision-based Autonomous Detection of Lane and Pedestrians", *International Conference on Signal and Image Processing*, IEEE, 2016.
- [6] Y. Wang, L. Bai, F. Michael, "Robust road modeling and tracking using condensation", *IEEE Transactions on Intelligent Transportation Systems* 9 (2008) 570-579.
- [7] A. Borkar, M. Hayes, M.T. Smith and S. Pankanti, "A Layered Approach to Robust Lane Detection at Night", *IEEE International Conference and Exposition on Electrical and Power Engineering*, pp. 735 - 739, 2011.
- [8] A. Guiducci, Parametric model of the perspective projection of a road with applications to lane keeping and 3d road reconstruction, *Computer Vision and Image Understanding* 73 (1999), 414-427.
- [9] Z. Teng, J.H. Kin and D.J. Kang, "Real-time Lane detection by using multiple cues", *IEEE International Conference on Control Automation and Systems*, pp. 2334 - 2337, 2010.
- [10] M. Aly, "Real time detection of lane markers in urban streets", *Intelligent Vehicles Symposium*, IEEE, 2008, pp.7-12, 4-6.
- [11] LAND E H, MCCANN J. "Lightness and retinex theory [J]". *Journal of Optical Society of America*, 1971, 61 (1): 2032 -2040.

Sparse Representation for Paddy Plants Nutrient Deficiency Tracking System

Zar Zar Tun, Khin Htar Nwe

University of Information Technology (UIT), Myanmar

zarzarhtun@uit.edu.mm, khinhhtarnwe@uit.edu.mm

Abstract

Moving object detection and tracking from consecutive frames of sensing devices (Unmanned Aerial Vehicles-UAV) needs efficient sampling from mass data with sufficient memory saving. Objects with super pixels are tracked by Compressive Sensing (CS) and the generative structural part model is designed to be adaptive to variation of deformable objects. CS can precisely reconstruct sparse signal with a small amount of sampling data. This system creates the sparse representation (SR) dictionary representing the nutrient deficiency tracking system for paddy plants to support the healthily growth of the whole field. This system uses compressed domain features that can be exploited to map the semantic features of consecutive frames. As the CS is a developing signal processing technique, a sparse signal is reconstructed with efficient sampling rate and creates the sparse dictionary. The SR for paddy plant health system can build rich information about paddy plants from signaling devices and can alert the deficiency conditions accurately in real time.

Keywords- Compressed Domain Features, Compressive Sensing (CS), Sparse Representation (SR), Dictionary Learning (DL), Paddy Plants

1. Introduction

Rice is the main food and the production of rice has become a major part of the economy in most Asian countries. Rice production is beneficial not only to farmers but also to the country. Rice fields are changing different brightness patterns at different plant growth stages while they are growing. The healthy state of the plants can be known from the pattern changes of the plants. This paper creates the nutrient deficiency tracking for paddy plant healthy growing system by using the motion values from a streaming video of the paddy fields. This system can alert the states of the plants in real time.

CS theory includes the three basic components as sparse representation, encoding measuring, and reconstructing algorithm [8]. SR is a novel signal sampling method for the sparse or compressible signal and has been successfully applied to signal processing. Sparse coding allows to represent a signal as a linear combination of a few atoms of a dictionary. Dictionary

learning methods determine the proper representation of data via decreased dimensionality subspaces.

Selection of extracted features plays an important role in this system. Other sparse representation systems use the sparse parameters from random frames that can cause computational complexity and needs more storage space in feature extraction. In this system, compressed domain features as Motion Vectors and residuals coefficients are extracted from video frames by partial decoding. Extracted information is representing the motion, spatial frequency, edge and color contents that can force to get accurate arguments. That can reflect the most matching edges as outliers and motion values as inliers for a specific condition of plants in creating the sparse dictionary.

The old classification systems extract HOG features and create Bag of Words model to detect objects and recognize actions. There are many drawbacks as memory complexity for storage space and decreasing the accuracy for real time system. The most important one is less of descriptors mapping from raw features to higher level feature labels. Sparse and redundant signal representations have recently used for solving existing problems as high transmission bandwidth and large storage memory allocation. CS can efficiently acquire and reconstruct a signal. The sparsity of a signal can be exploited to recover a signal from fewer samples than required in the Shannon-Nyquist sampling theorem [25]. This system creates the sparse dictionary using sparsity coefficients getting from compressed domain features for efficient tracking the healthy states of paddy plants in real time.

The rest of this paper is organized as follows: Session 2 reviewed related works concerning with this system. Session 3 will be discussed about the proposed system. Session 4 will be explained the experimental results which will be followed by conclusion in Session 5.

2. Related Work

There has been considerable effort devoted to create this system in the last decade. L.Pan, X.Shu and M.Zhang [3] proposed efficient key frame extraction algorithm that exploits Compressive Sensing and unsupervised clustering. J.Jiang et al. [4] proposed a new dimensionality reduction method called

compressive sensing with Gaussian mixture random matrix (CS-GMRM), in which a novel measurement matrix using Gaussian mixture distribution is constructed and is proved to satisfy the restricted isometry property.

X.Huang et al. [5] segmented the moving object through the robust principal component pursuit (PCP) for that the image is consisted with low-rank of the background regions and the sparsity of the foreground regions. Then, the data dictionary is created through KSVD to strengthen the sparse representation of the dictionary capabilities. S.Qaisar et al. [6] presented a brief background on the origins of the CS idea, reviews the basic mathematical foundation of the theory and then highlighted different areas of its application with a major emphasis on communications and net-work domain.

B.Kaung et al. [7] proposed an object detection model to simultaneously reconstruct the foreground, background, and video sequence using the sampled measurement. Then, they used the reconstructed video sequence to estimate a confidence map to improve the foreground reconstruction result. In paper [8], authors provided a comprehensive study and an updated review on sparse representation to supply guidance for researchers.

V.M.Patel and R.Chellappa reviewed the role of Sparse Representation (SR), Compressive Sensing (CS) and Dictionary Learning (DL) for object recognition. Algorithms to perform object recognition using these theories are reviewed [9]. In paper [10], authors addressed the problem of object detection by representing an extracted feature of an image using a sparse linear combination of chosen dictionary atoms.

Authors in [11] presented the comparison of recently proposed CS several methods as Haar transforms, hybrid CS-Haar, averaging and sub-sampling, and performing recognition to compress time series either directly in the compressed domain over the reconstructed signals. Authors in [12] proposed to decompose the motion field into sparse and non-sparse components for the motion boundaries and small universal noises. By exploiting the statistics on optical flow fields dataset, authors found that gradients of flow fields come from two sources: a sparse large motion-discontinuity component and small dense Gaussian component.

S.Hou, S.Zhou and M.A.Siddique [13] proposed about CS based algorithms that are investigated for Query by Example Video Retrieval (QEV) and a novel similarity measure approach. This system combined CS theory with the traditional discrete cosine transform (DCT), better compression efficiency for spatially sparse is achieved. X.Shu and N.Ahuja [14] proposed a three-dimensional compressive sampling (3DCS) approach to reduce the required sampling rate of the Compressive Imaging (CI) camera to a practical level.

In 3DCS, a generic three dimensional sparsity measure (3DSM) is presented, which decodes a video from incomplete samples by exploiting its 3D piecewise smoothness and temporal low rank property.

S.Narayanan and A.Makur [15] proposed to use a circulant CS matrix on image frames to obtain the CS measurements and then to perform motion estimation in the measurement domain. G.Chen and D.Needell [16] introduced the compressed sensing problem as well as recent results extending the theory to the case of sparsity in tight frames and the problem of dictionary learning, its origin and applications, and existing solutions. G.Li et al. [17] proposed a robust object tracking and generative action recognition method.

All reviews supported to develop this proposed system.

3. The Proposed System

This system includes three contributions as:

- (1) This system creates the first paddy plants health system using the sparse representation dictionary.
- (2) This system extracts sparse values representing motion, spatial frequency, edge and color contents from compressed features that can force to get accurate arguments.
- (3) This system can track and alert accurate paddy plant deficiency conditions in real time.

CS is an innovative concept that directly acquires signals in a compressed form if they are sparse in certain transform domains. This system creates the sparse dictionary using sparse representation coefficients getting from compressed domain features. Sparse representation can reduce existing noise reduction and dimensionality reduction methods because the relevance of CS that is a dimensionality reduction technique for series of sampled signals. Figure 1 shows the functional architecture of the proposed system.

This system creates the paddy plants healthy state system using the sparse dictionary for tracking in real time. Existing systems [1, 22-24] are using HOG, MBH and MFH features and creates the Bag of Words model to reconstruct and track the moving features. These old systems accuracy depends also on the code books classification methods. There are so many challenges such as increasing the memory requirements, higher complexity and less of higher visual level features mapping for deformable objects from sensing devices. Increasing features can degrade the accuracy for real time systems.

In this system, compressed domain features are firstly extracted from original video sequences. Then this system counts sparse coefficients in compressed domain by CS to create the sparse dictionary that is sparse representation coefficients classes of linear

representation system. Test samples can usually represent samples from the same objects of incoming videos. This system uses the sparse dictionary to analyze different situations from new coming video shots or sensing states for efficient tracking of deficiency conditions of paddy plants.

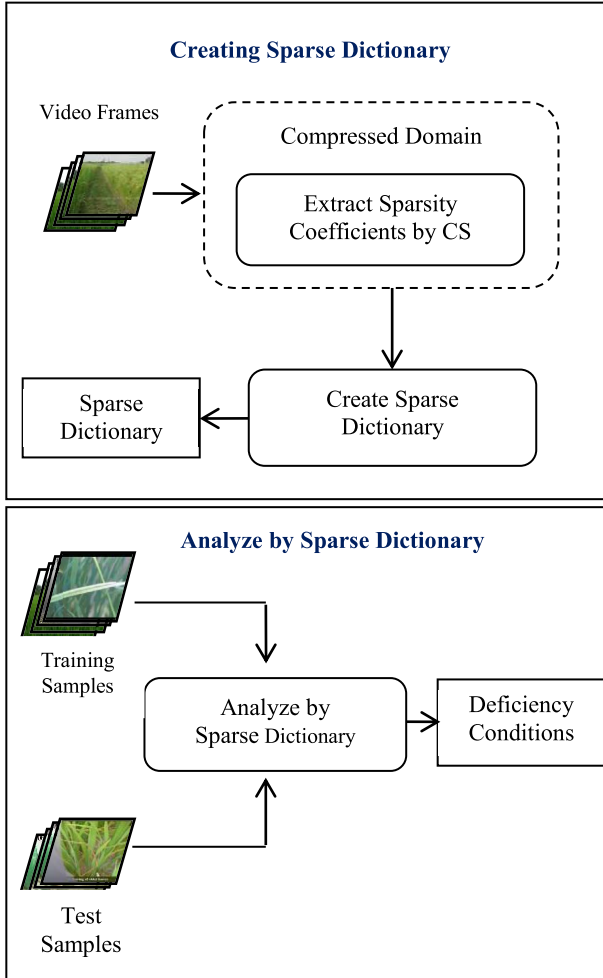


Figure 1. Functional Architecture of the Proposed System

3.1. Compressed Domain Features Extraction

The streaming video of sensing devices consists of compressed domain features as Motion fields and its intensity information of the ongoing scene. Motion fields are representing the Motion Vectors and intensity information is representing Discrete Cosine Transform (DCT) coefficients that form as residuals when motion estimation is performed.

Motion estimation is the process of estimating the best match block of current frame in the reference frame. There are three types of frames in video sequences as Intra frame (I-frame), Predicted Frame (P-

frame) and Bi-directionally predicted frame (B-frame). There is also prediction error that is the result of difference between motion vectors and transforms coefficients.

Figure. 2 shows the process of Motion Estimation. Motion Vectors and its residual coefficients are complementary to each other to form an accurate action.

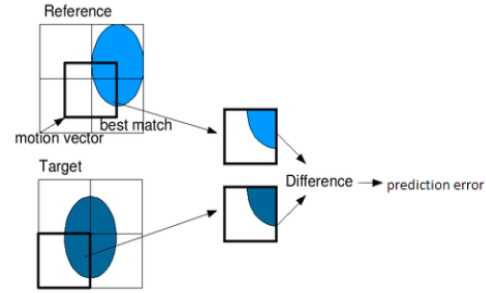


Figure 2. Motion Estimation Process [2]

This system accumulates all motion vectors and residuals of P frames from incoming video sequences as formulated in equation (1):

$$X = \sum_i^n |u_i| + |v_i| \quad (1)$$

where X consists of total motion values from P frames in a video sequence, u and v are representing the motion vectors and residuals. Motion data that do not represent real values are filtered by Gaussian approach to be more reliable in motion segmentation. This system uses these compressed domain features that can reduce the computational complexity because coefficients are extracted by partial decoding instead of fully compression.

3.2 Sparse Representation in Compressive Sensing

Compressive sensing (CS) is a novel signal sampling method to reconstruct a signal from a series of sampling measurements by finding the solutions to underdetermined linear systems. Nyquist- Shannon sampling theorem states that if the signal's highest frequency is less than half of the sampling rate, then the signal can be reconstructed perfectly [20]. There are two types of conditions as sparsity and incoherence to recover a signal.

Sparsity: Natural signals can be stored in compressed form if a large number of projection coefficients are small enough to be ignored. Most of elements are zero in sparse matrix or sparse array. If the signal is not sparse, then recovered signal is best reconstruction getting from S largest coefficients of signal. If the total number of elements are R , total

sparse elements are S and the left are dense elements $N = R - S$.

In most problems, signals are modeled by a small set of prototypes. In this system, the prototype signal representing motion values, $X = [x_1, x_2, \dots, x_n] \in R^n$ is used for training the dictionary $D = [d_1, d_2, \dots, d_m] \in R^{m \times n}$, which can be considered as an overcomplete basic matrix consisting of elementary signals called atoms. In the overcomplete dictionary, the number of samples is larger than the dimensions of the samples in the dictionary. This system uses the learned dictionary from compressed features matrix as:

$$x = D\alpha \quad (2)$$

The probe sample, $x \in R^{n \times m}$ that can be achieved by a linear combination of a few small number of dictionary atoms. $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_m]^T$ is the sparse coefficient matrix. The solution of α can be searched by the following problem:

$$\arg \min_{\alpha} \|\alpha\|_0 \quad \text{subject to} \quad x = D\alpha \quad (3)$$

where the equation (3) denotes the l_0 -norm which counts the number of non-zero entries in a vector. As it is NP hard, alternative solutions are often sought. If the solution of x is sparse enough, then the sparsest solution can be recovered via l_1 minimization as shown in equation (4):

$$\arg \min_{\alpha} \|\alpha\|_1 \quad \text{subject to} \quad x = D\alpha \quad (4)$$

where, the equation (4) provides the sparsest recovery [18]. If the measurements are contaminated in the noisy setting because of the error, e which obeys $\|e\|_2 < \varepsilon$ as follows:

$$y = D\alpha + e \quad \text{for} \quad \|e\|_2 < \varepsilon \quad (5)$$

where ε is the allowed error tolerance. This system finds the sparsest vectors using Orthogonal Matching Pursuit [21] that has average trade-off classification accuracy and processing time comparing with other sparse representation methods [8] and this equation is presented as follows:

$$\arg \min_{\alpha} \|\alpha\|_1 \quad \text{subject to} \quad \|D\alpha - y\|_2 < \varepsilon \quad (6)$$

Intuitively, the l_1 -norm is the convex relaxation function closest to the l_0 -norm.

3.3 Paddy Plants State Dictionary Learning

The dictionary learning (DL) method determines the proper representation of data via decreased dimensionality subspaces. An effective dictionary can lead to excellent reconstruction results and satisfactory applications, and the choice of dictionary is also significant to the success of sparse representation technique. Sparse dictionary not only provides a sparse representation but also constructs a sparse dictionary [26].

This system creates the sparse dictionary using the prototype signals representing the nutrient deficiency features as columns that are extracted from consecutive frames of a streaming video of paddy fields. This system finds the sparse representation vectors by l_1 -norm and then updates the dictionary by K-SVD. These two steps are iteratively preserved until a sufficient small residue point is reached.

This system uses the feature sets samples $X = [x_1, x_2, \dots, x_k], x_i \in R^d$ to find the dictionary $D \in R^{d \times n}$, $D = [d_1, \dots, d_n]$ and the representation matrix $R = [r_1, \dots, r_k], r_i \in R^n$. This system uses K-SVD algorithm because it is one of the most simplicity and effective algorithm among existing dictionary learning algorithms [19]. It is finding the best possible codebook to represent the data samples y by nearest neighbor. $\|X - DR\|_F^2$ is minimized by equation (6) and the representation r_i are sparse enough. Then the dictionary D is updated by the following optimization problem,

$$\arg \min_{D, R} \|X - DR\|_F^2 = \sum_{i=1}^d \sum_{j=1}^n \|x_{i,j} - Dr_{i,j}\|_2^2$$

$$\text{subject to} \quad \|r_{i,j}\|_0 \leq \varepsilon \quad (7)$$

where $r_{i,j}$ is the sparse representation for the j -th samples of class i , and ε indicates the maximum allowed nonzero entries in $r_{i,j}$. F denotes the Frobenius norm. This system can gradually update the dictionary about the healthy status of paddy plants in real time and drastically reduce the amount of memory needed to store the huge size of dataset.

4. Experimental Result

This system can highly provide to manage the nutrient deficiency in paddy plants. This system gives the facts for totally eleven deficiencies as feature vectors about health paddy conditions: Phosphorus that can help in fibrous root development. If there is phosphorus deficiency, the plant is in purple or brownish red discoloration on leaves. Nitrogen encourages the vegetative growth of paddy. If nitrogen deficiency occurs, older leaves become yellow and stunt in growing plants. Calcium promotes the activities of soil bacteria. Magnesium is the essential constituent in Chlorophyll molecule. Iron is necessary for chlorophyll synthesis. Manganese helps in uptake of Nitrogen. If iron, manganese and magnesium deficiency occurs interveinal yellowing and chlorosis of young leaves.

Copper is important for panicle development. Sulphur is constituent in straw and stalk. Boron helps in fertilization. If calcium and boron deficiency occurs, leaves be-come white and tips of young leaves roll. The other two conditions are zinc toxicity and aluminum toxicity of plants. If farmers know the conditions of plants, correct measures can be applied to the fields in time.

The training dataset is being constituted using the motion values resulting from streaming frames of sensing video about eleven kinds of paddy plant deficiency states. Incoming video is sampled with 25 frames per second with totally 5356 frames of width 128 and height 128. As requirements for implementation, this system uses MATLAB implementation on the processor core i7 and 4GB RAM.

The training dataset is normalized with zero mean and unit variance. 30 percent of the training dataset is used as testing dataset to test the new incoming video or state of feature occurrence. The testing dataset is 1601 signals with total dimensions of 1296. The resulting datasets are down sampled by PCA and LDA filtering methods.

Then the dictionary is learned by K-SVD for linear representation for a given set of signals in training dataset. K-SVD can search the best dictionary that can sparsely represent each signal [19]. The learned dictionary is a matrix that contains signals, each of dimensions n (1296). The dictionary was trained 3 times. There are five iterations each time. For each iteration, the number of atoms used to represent a signal is changing as 5 atoms per signal, 10 atoms per signal, 15 atoms per signal, 30 atoms per signal and 40 atoms per signal for testing.

The comparison of average representation error (RMSE) for three group of signal is displayed in Figure 3. All sample groups are under specific representation error.

Meanwhile, this system uses the l_1 -norm sparse representation method to find the precise sparse value about features of paddy plants. The sparse coefficients of signal are 1601 signals with 3755 dimensions. The average accuracy of the classification and the average speed of sparse coding for different number of signals are presented in Figure 4 and Figure 5.

When implementing this system with testing dataset, the accuracy level is about 97% on the average. According to the experimental result, average accuracy is higher when the number of samples increases although the processing time is a little longer.

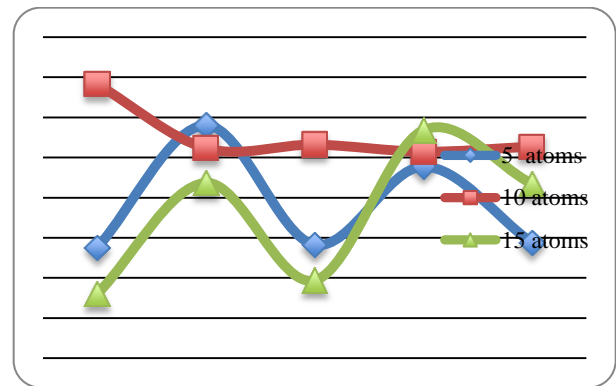


Figure 3. Comparison of RMSE of the Learned Dictionary

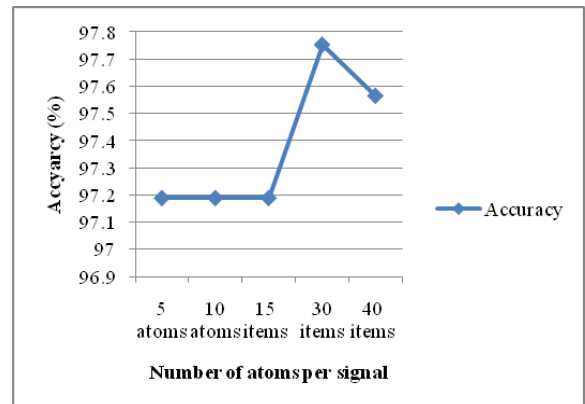


Figure 4. Average Accuracy of Different Items per Signal

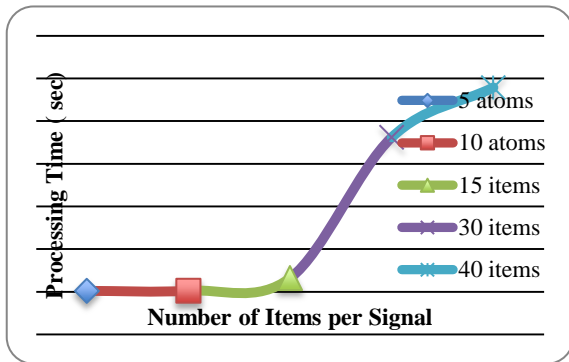


Figure 5. Average Speed of Different Items per Signal

5. Conclusion

Compressive sensing takes the advantage of the redundancy in many interesting signals that are not pure noises. The main idea of this system is that a signal can be decoded from incomplete linear measurements by seeking its sparsity in some domain. There are many ways to represent the signal, but normally the sparsest representation is preferred for simplicity and easy interpretability for sampling. This system typically starts with taking a weighted linear combination of samples about nutrient deficiencies of paddy plants from compressed domain of sensing video data to get compressive measurements. Then this system creates the sparse dictionary consisting of prototype signals about conditions of the plants that are used to express other signals for real time alerts. This system can also be applied for different object tracking systems in many areas. In the future, this system tends to expand to other related fields of study in application requirements for urban countries.

6. References

- [1] Zar Zar Tun and Khin Htar Nwe, "Compressed Domain Feature Analysis For Efficient Action Recognition", *International Conference on Disruptive Innovation (ICDC)*, MUST, Malaysia, 24-25 Sept, 2016.
- [2] Zar Zar Tun and Khin Htar Nwe, "Comparison of Different Motion Estimation Algorithms in Video Compression", *International Conference on Computer Applications (ICCA)*, Myanmar, 16-17 Feb, 2017.
- [3] L.Pan, X.Shu and M.Zhang, "A Key Frame Extraction Algorithm Based on Clustering and Compressive Sensing", *International Journal of Multimedia and Ubiquitous Engineering*, pp. 385-396, Vol.1.0, No.11, 2015.
- [4] J.Jiang, X.He, M.Gao, X.Wang and X.Wu, "Human Action Recognition via Compressive-Sensing-Based Dimensionality Reduction", *Optik - International Journal for Light and Electron Optics*, Volume 126, Issues 9-10, May 2015, Pages 882-887.
- [5] X.Huang, F.Wu and P.Huang, "Moving-object Detection Based on Sparse Representation and Dictionary Learning", *AASRI Conference on Computational Intelligence and Bioinformatics*, Volume 1, 2012, Pages 492-497.
- [6] S.Qaisar, R.M.Bilal, W.Iqbal, M.Naureen and S.Lee, "Compressive Sensing: From Theory to Applications, A Survey", *Journal of Communications and Networks*, October 2013, Page: 443-456.
- [7] B.Kaung, W.P.Zhu and J.Yan, "Object Detection Oriented Video Reconstruction Using Compressed Sensing", *Journal on Advances in Signal Processing 2015(1)*, November 2015.
- [8] Z.Zhang, Y.Xu, J.Yang and S.Zhang, "A Survey of Sparse Representation: algorithms and Applications", *The journal for Rapid Open Access Publishing, IEEE*, May 20, 2015.
- [9] V.M.Patel and R.Chellappa, "Sparse Representations, Compressive Sensing and Dictionaries for Pattern Recognition", *Processing of IEEE*, March 2011.
- [10] G.K.Vinay, S.M.Haque, R.V.Babu and K.R.Ramakrishnan, "Human Detection Using Sparse Representation", *IEEE International Conference on Intelligent Computing and Intelligent Systems*, December 2009.
- [11] R.Xu, O.P.Concha and M.Piccardi "Compressive Sensing of Time Series for Human Action Recognition", *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, December 2010.
- [12] Z.Chen, J.Wang and Y.Wu "Decomposing and Regularizing Sparse/Non-sparse Components for Motion Field Estimation", *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2012.
- [13] S.Hou, S.Zhou and M.A.Siddique "A Compressed Sensing Approach for Query by Example Video Retrieval", *Multimedia Tools and Applications*, Volume 72, Issue 3, October 2014, pp 3031-3044.
- [14] X.Shu and N.Ahuja "Imaging via Three-dimensional Compressive Sampling (3DCS)", *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011.
- [15] S.Narayanan and A.Makur "Camera Motion Estimation using Circulant Compressive Sensing Matrices", *International Conference on Information, Communications & Signal Processing (ICICIS)*, December 2013.
- [16] G.Chen and D.Needell "Compressed Sensing and Dictionary Learning", 29-October- 2014.
- [17] G.Li, F.Wang and W.Lei "Generative Human Action Tracking Based on Compressive Sensing", *International*

Journal of Signal Processing, Image Processing and Pattern Recognition, Vol.8 No.7, 2015.

[18] A.M. Bruckstein, D.L. Donoho, and M. Elad, "Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images", *SIAM Review*, 2009, vol. 51, no. 1, pp. 34-81.

[19] M. Aharon, M. Elad, and A.M. Bruckstein, "The K-SVD: An Algorithm for Designing of Overcomplete Dictionaries for Sparse Representation", *IEEE Trans On Signal Processing*, November 2006, Vol. 54, no. 11, pp. 4311-4322.

[20] E.J.Candes and M.B.Wakin, "An Introduction to Compressive Sampling", *IEEE Signal Processing Magazine*, March-2008.

[21] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit", *IEEE Transactions on Information Theory*, 2007, vol. 53, no. 12, pp. 4655-4666.

[22] V.Kantorov and I.Laptev, "Efficient Feature Extraction, Encoding and Classification for Action Recognition", *Computer Vision and Pattern Recognition (CVPR), IEEE*, 2014.

[23] J.Miao, X.Xu, R.Mathew and H.Huang, "Residue Boundary Histograms for Action Recognition in the Compression Domain", *IEEE International Conference on Image Processing*, Sept-2015, vol.25, pp.39-52.

[24] J.Uijlings, I.C.Duta, E.Sangineto and Nicu Sebe, "Video Classification with Densely Extracted HOG/ HOF/ MBH Features: An Evaluation of the Accuracy/ Computational Efficiency Trade-off", *International Journal of Multimedia Information*, March 2015, Volume 4, Issue 1, pp33-44.

[25] https://en.wikipedia.org/wiki/Compressed_sensing, last accessed 13-Oct-2017.

[26] https://en.wikipedia.org/wiki/Sparse_dictionary_learning, last accessed 13-Oct-2017.

Mobile and Distributed Computing

The Home Safety System Based on Competing Functions

Zaw Myint Naing Oo, Khin Kyawt Kyawt Khaing
University of Information Technology
zawmyintnaingoo@uit.edu.mm, khinkkkhaing@uit.edu.mm

Abstract

This paper presents a home safety system assured the safety of home appliances that can be integrated with the existing home automation. This system consists of generic rules, a fuzzy set of rules and inferences the home safety services according to the sensor input values. This system has some filters, that judge is the appropriateness of sensor values. This system will use the competing home appliances which are automatically controls near boundary on/off. So, this system has the special database system for competing appliances. This database system will link to the fuzzy logic decision making system. The Geo Fencing system will also be applied to watch the movement of object. IFTTT (If This Then That) service will provide the automotive function as a remote control. The Remote UI will be applied to monitor the condition of the home status.

Keywords- Home safety, Generic Rules, Database, Geo Fence, IFTTT, The Remote UI

1. Introduction

Intelligent home is an integrated system in a home that integrates multiple home services, where the technology and process used to create a building that can act intelligently so that a home becomes safer and more productive for users and more efficient for its owners. The proposed system realizes by the sensors value rules. This system needs to define the set of fuzzy rules. People are always worried about what would be the condition of their homes and offices when they are not there. Therefore, this proposed system is trying to make a system which would automatically provide the user to save and control the home appliances [1].

There are many types of safety problems that may arise within a home environment. These safety problems can be classified into three big categories: safety of home appliances, safety of indoor environment and safety of interaction between home users and home appliances. The occurrence of home safety problem always have three bad consequences: cause casualty or cause home property loss or both [2].

This system will define the generic rules based on the sensor values. The generic rule is the representation of

common senses in terms using the syntax of the fuzzy decision support system. The system will use some filters to judgment between sensors and embedded appliances, because they might be malfunctioned.

This system will use the fuzzy inference system to inference the home safety. Intelligent home is an integrated system in a home that integrates multiple home services, where the technology and process used to create a building that can act intelligently so that a home becomes safer and more productive for users and more efficient for its owners.

The Geo Fencing system will watch and act as a sentinel system, how the user is moving which way the user moves to the Geo Fence (from inside, outside, inside/outside cross direction). These three conditions will include for the Geo Fencing system.

The Remote UI will also provide the functions to monitor and to control the status of home appliances.

2. Related Works

The architecture of home safety system includes the Geo Fencing rules for intelligence fence, IFTTT acts like as remote control, sensor value rules for controlling the home appliances and the remote UI for monitoring the status of home. This proposed system used the fuzzy expert system. This system defined a set of fuzzy rules according to the input sensor values which can be obtained by the sensors. This proposed system acted as a sentinel, which knows everything at home situations. It can provide home safety functions, and can also save the electricity usage. On the weekdays and weekends, the system automatically works based on the rules to save the electricity usage. In this proposed system is a new technique of implementing home safety system that will give more safety for smart home appliances and electrical usage based on the rules [1].

Intelligent home management system has been developed which has the ability to turn on and turn off the room lights automatically, record the controlled electronic devices usage status, switching on and off air condition regulating device automatically, showing temperature room in the house, detect fire signs in the house and turned on the sprinklers in the home in case of fire, supervising the home through surveillance cameras, storing photos and surveillance records on home, detecting people movement in home, and providing

notification when someone entered home. System is implemented in prototype. The results show that the system can detect light intensity, flame, room temperature, movement of people, and home state and then the information is successfully sent to the server over the WiFi. The result can be read from server by using browser and there is a data logger in the server. Intelligent home management system prototype development covers hardware and software implementations [3].

Expert systems are normally used in various problem solving and decision making activities such as monitoring, diagnosing and various training related activities. Yashwant Singh Patel proposed a framework that is based on wireless sensors and expert system to solve day to day problem occurring in home appliances. Whenever problem occurs in any part of home appliance, the sensor detects that problem automatically and sends it for solution to the expert system, Various noise removal algorithms for removing noise from the received data can be applied for getting noise free data. The expert system finds the solution based on the type of problem and sends the solutions with various images through SMS or e-mail to user's mobile or mail-id [4].

The author proposed the intelligent control in smart home based on adaptive Neuro Fuzzy Inference System (ANFIS). This research proposed the use of K-means clustering algorithm in the division of the input space. Every cluster generates a membership function by approximation, the type of membership function is the bell, and then the optimization of the premise and consequent parameters in ANFIS model are realized through the combination of improved adaptive particle swarm algorithm and the least squares method. When the number of iterations that users set is reached, the satisfactory ANFIS model is obtained. The model also went through the simulation of controlling the electric curtains of the smart house in the Matlab platform. Theoretical analysis and simulation experiments show that this model can improve the learning ability of home control system [5].

3. Background Theory

3.1 Fuzzy Inference System

The primary objective of fuzzy logic is to map an input space to an output space. The way of controlling this mapping is to use IF-THEN statements known as rules. The order in which these rules are carried out is insignificant, since all rules run concurrently. Fuzzy logic is a powerful problem-solving methodology with a myriad of applications in embedded control and information processing. It provides a remarkably simple way to draw definite conclusions from vague, ambiguous, or imprecise information. In a sense, it resembles human

decision making with its ability to work with approximate data yet finds precise solution [6].

Fuzzy logic provides an approach to data fusion and reasoning for uncertain data by using the human expert knowledge. The Fuzzy Inference System (FIS) is as shown in figure 1. It is divided into three main components: the fuzzifier, the knowledge management and the defuzzifier [1].

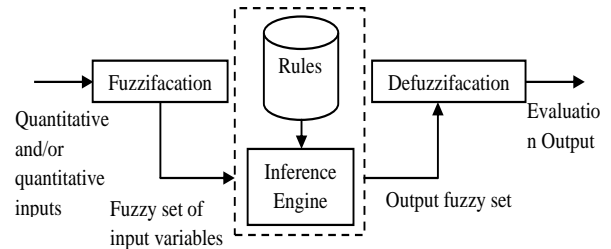


Figure 1. The Fuzzy Inference System

3.2. Geo fencing

Geo fencing is a technology used to monitor mobile objects (vehicles, persons, container, etc.), located by GPS. The geographic coordinates of the tracked object are automatically and regularly sent to a control center, via mobile phone networks. The set of geographic coordinates is used to constitute a virtual boundary (Geo Fence) around a geographic area. The system can determine whether the tracked object is located inside or outside the Geo Fenced area. This technology can also allow the detection of spatial proximity between the tracked mobiles and a specific Geo Fenced area [7].

3.3 IFTTT

IFTTT is a web based service that allows Internet users to create a chain-reaction from one web service application to another. Based on a user-defined conditional statement, called a recipe, the trigger of one web service application activates an action of another web service application. The IFTTT model can be applied to home automation devices where one device can trigger the action of another device. The IFTTT technology is described as shown in Figure 1. The Figure 2 describes how home automation devices would react on the user-define recipes. Two recipes are shown in Figure 3. First recipe is “If motion is detected in a room, then turn on the lights”. When the motion sensor in the room detects a movement, it sends a trigger to the central node. Based on the recipe and the trigger, the central node sends an action to the room lights to turn on. Second recipe is “If temperature and humidity changes in the garden, the turn on the irrigation system”. When the temperature and humidity sensor senses change, it sends the trigger to the central node. Then, the trigger is

interpreted by the central node that sends an action to the irrigation system. These recipes can be generated by remotely accessing the central node of the home automation system, or it can also be accessed within the home network. The central node acts as a router for the home devices to access the Internet and integrates all different types of data communication mediums. Therefore the central node offers a web interface to allow users to configure the different recipes, which can be accessed from computers, smartphones or tables [8].

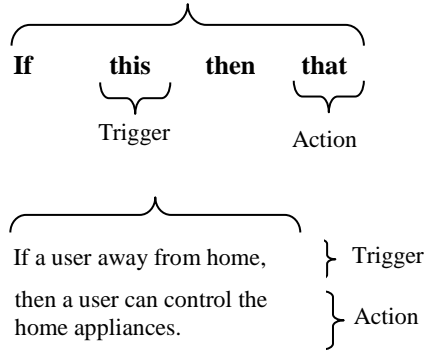


Figure 2. IFTTT Description

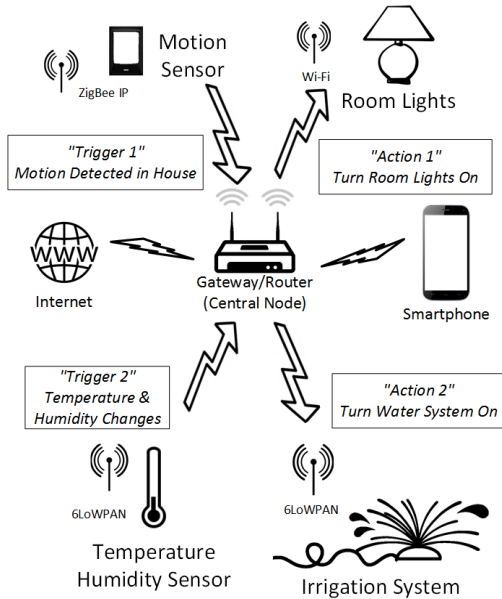


Figure 3. Home Automation Overview

3.4 Remote UI

Remote UI refers to Web 2.0. The user can create new services by combining the object provided services, it is called Web 2.0 or mashup. It can be specialized for the composition of services that enable accessing/controlling

smart things [10]. A mashup is a web application or a web page which usually uses application programming interfaces (APIs) in order to blend information from multiple sources to create compelling services. As more and more embedded devices (like smartphones and sensor equipped appliances) will be apply to provide their functions as services online, and an abundance of real objects will essentially become a part of ambient spaces (interoperating and communicating over TCP/IP networks), the need to create value-added services by composing numerous embedded-device enable services [9].

4. The Architecture of Home Safety System

The architecture of home safety system is shown in Figure 4. This architecture includes the special database system for the competing home appliances, some filters to judgement for sensors and home appliances to avoid malfunction, the Geo Fencing rules for intelligence fence, IFTTT acts like as remote control, sensor value rules for controlling the home appliances and the remote UI for monitoring the status of home.

This system will use generic rules. Firstly, the system needs to define a set of fuzzy rules according to the input sensor values. This system also needs to define competing functions for competing appliances. The input value can be obtained by the sensors. Secondly, it proceeds to perform the fuzzy decision support system. And then this system will be made defuzzification by using the Sugeno fuzzy inference method to get crisp output. The Sugeno fuzzy inference method can be computed by the weighted average method. According the crisp output, finally the system will save the home appliances according to the competing functions which located in the special database.

In this system, the size of Geo Fence size can range from a few tens of meters to several kilometers. The Geo Fencing areas can be defined by geometric shapes. The geographical areas are defined as circular area, rectangular area and ellipsoidal area.

This system defines the circular geographical area with a single point that represents the center of the circle and a radius. Coordinates from characteristic points of the shape are necessary to define the Geo Fence perimeter. These coordinates are used in equation (1), along with the inside or outside of the Geo Fence, which enables the computing of alerts. Sensor value rule uses the appropriate sensor values within the total range and the geo-fencing rules use fuzzy control logic, which is the IF THEN statements. The geographical circular area is described as shown in Figure 5. The function of geographical circular area is defined by equation (1).

$$F(x, y) = 1 - \left(\frac{x}{r}\right)^2 - \left(\frac{y}{r}\right)^2 \quad (1)$$

Where F is the function to determine the spatial characteristics of a point (x,y) relative to a geometric shape, r is the radius of a circle, x is the abscissa of a Cartesian coordination system with the origin in the center of the geographical area, y is the ordinate of a

Cartesian coordination system with the origin in the center of the geographical area. The function F defined in equation (1), determines whether a point is located inside, outside, at the center, or at the border of a geographical area.

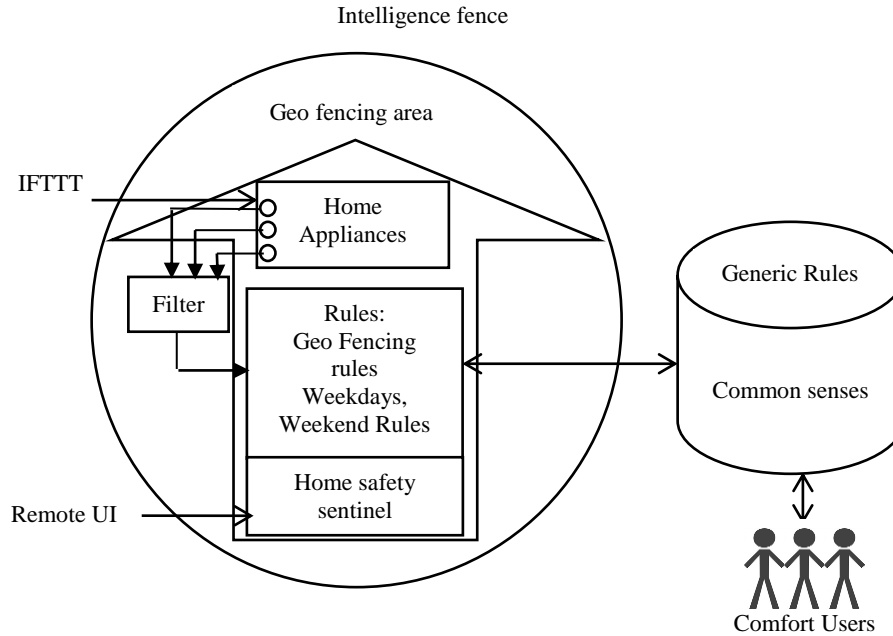


Figure 4. The architecture of home safety system

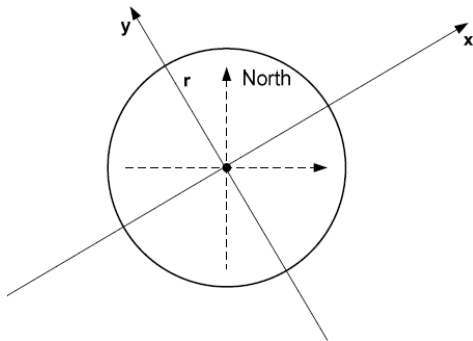


Figure 5. The geographical circular area

If the value of function F is equal to one, the location is at the center point of the geographical area. If the value of function F is greater than zero, the location is inside the geographical area. If the value of function F is equal to zero, the location is at the border of the geographical area. If the value of function F is less than zero, the location is outside the geographical area.

IFTTT is a web-based service that allows Internet users to create a chain-reaction from one web service application to another. Based on the IFTTT (IF-This-Then-That) model, this system will define a set of device communication protocols where devices' triggers and actions are combined to manage interactions for home safety. This system uses Web 2.0 for remote user

interface and creates new services by combining the object provided services.

5. Evaluation of Home Safety System

The sensor and embedded appliances might be malfunctioned. So, this system will define the generic rules and must have some filters to avoid the malfunctions. The generic rule is the representative of common sense. The scenario is using the temperature to define the rules. These rules will be putted into the special database. When the people are sleeping sometime, they use the blanket, because the room temperature is a little bit low, i.e., common sense. There has rules based on the common senses, about the temperature, humidity, the electricity usages and more interestingly the home appliances, we can hold the state of appliances then we will be doing some more interesting senses, competing appliances (e.g., the heater and air con).

In this system, we assume that the competing appliances, the air conditioner is set to 25 degree and the heater is also set to 25 degree. When the heater starts to heat, it takes time to give warm. When the room temperature is high, the air conditioner kick the heater, it takes time to low the room temperature. They may over

shift belong, and as soon as over shift. But, the temperature was higher than 25 degree, may be outside temperature as like 30 degree, at that time the air conditioner kick the heater, to cool down the temperature, (i.e competing appliances (competing functions)).

The following rules are the generic rules to use the competing functions. These rules are located into the special database.

Rule 1: if the heater is higher than 25 degree, then the air conditioner is cool down the temperature until 25 degree.

Rule 2: if the air conditioner is lower than 25 degree, then the heater is high temperature until 25 degree.

Rule 3: if the outside temperature is higher than 25 degree, then the air conditioner is cool down the temperature until 25 degree and the heater is just warm.

Rule 4: if the outside temperature is lower than 25 degree, then the heater is warm up to 25 degree and the air conditioner is still 25 degree.

This system will also use the rules for the Geo Fencing system. In this system, it uses the linguistic variables of fuzzy set for testing the Geo Fencing system which is shown in table 1.

Table 1. Linguistic Variables of Fuzzy Set

	Light	Air Con	Fan	Doors
Inside	On	On	On	On
Center	On	On	On	On
Border	On	On	On	On
Outside	Off	Off	Off	Off

This system defines the following rules to control the home appliances by using the geographical area.

Rule 1: If $F(x,y) = 1$ then the location is at the center point of the geographical area (control the home appliances)

Rule 2: If $F(x,y) > 0$ then the location is inside of the geographical area (control the home appliances)

Rule 3: If $F(x,y) = 0$ then the location is at the border of the geographical area (control the home appliances)

Rule 4: If $F(x,y) < 0$ then the location is outside of the geographical area (lock the home)

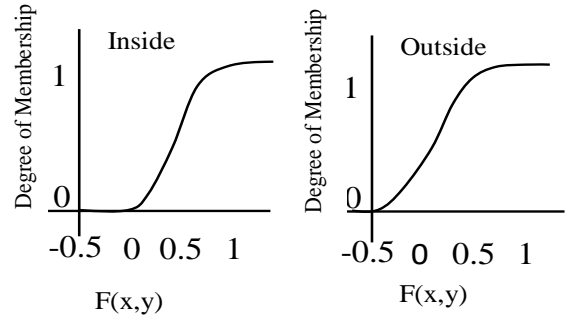


Figure 6. Fuzzy set of inside and outside of geographical area

Each input variable has membership functions as shown in figure 6. The output variable also has membership as on, off in table 1. These rules are applied to input and output of the Sugeno inference system in equation (2), which is weight average method to get crisp output for controlling the home appliances. The crisp output of the system is the weighted average of all rule outputs, computed as

$$\text{The crisp output} = \frac{\sum_{i=1}^N W_i Z_i}{\sum_{i=1}^N W_i} \quad (2)$$

Where, N is the number of rules, Z_i is the output level of rules and W_i is output degree of rules. According to the crisp output value from this equation (2), it will apply to control the home appliances.

The Figure 7 is showing the result of IFTTT service to control the appliances by using IFTTT service. The IFTTT service can create chains of conditional statements, which is called 'applet'. The following conditional statement is tested based on android location, which acts as a remote control for appliances.

If (EnteredOrExited) an area (OccuredAT) via Android (LocationMapUrl) then (Notify or Control the appliances)

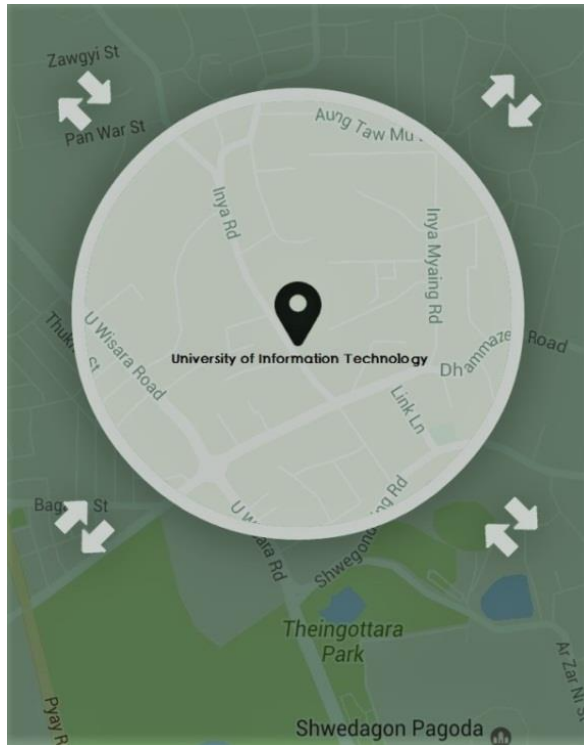


Figure 7. IFTTT service

If the user entered or exited at the specified area then send notification to the user and the user can control appliances which the user wants to switch on/off for electrical appliances.

6. Conclusion

This paper proposes a home safety system which helps us to assure the safety of home appliances and home environment. This system acts as a sentinel, which knows the movement of user to Geo Fence from inside, outside or cross direction. It can provide home safety functions. In this system has a special database for competing appliances. It is a new technique of implementing home safety system that will give more safety for smart home appliances based on the rules. This system will save cause casualty or cause home property loss or both. There exists several home safety systems. This system to be more effectively and safety for home. This system will be acted intelligently the home safety services as like the human manner. In

future work, this research plans to develop the detail of competing functions and movement of user.

7. References

- [1] Zaw Myint Naing Oo, Tha Pyay Win, "The Development of an Intelligent Fuzzy Expert System for The Home Safety System", The 15th International Conference on Computer Application 2017, Feb 16th -17th 2017, 13-18.
- [2] Zhengguo YANG, Azman Osman LIM, Yasuo TAN, School of Information Science, Japan Advanced Institute of Science and Technology, "Event-based Home Safety Problem Detection Under The CPS Home Safety Architecture," 2013 IEEE 2nd Global Conference on Consumer Electronics (GCCE)
- [3] Azka Ihsan Nurrahman, Kusprasapta Mutijarsa, "Intelligent Home Management System Prototype Design and Development," International Conference on Information Technology Systems and Innovation, Bandung-Bali, November 16-19, 2015 IEEE
- [4] Yashwant Singh Patel, Sneha Vyas, Atul Kumar Dwivedi, "A Expert System based Novel Framework to Detect and solve the Problems in Home Appliances by Using Wireless Sensors," 2015 1st International conference on futuristic trend in computational analysis and knowledge management (ABLAZE 2015)
- [5] Wanglei, SHAO Pingfan, "Intelligent Control in Smart Home based on Adaptive Neuro Fuzzy Inference System," 978-1-4673-7189-6/15/\$31.00©2015 IEEE
- [6] Mirza Mansoor Baig, Hamid Gholamhosseini, Michael J. Harrison, "fuzzy Logic Based Smart Anaesthesia Monitoring System in the Operation Theatre,"E-ISSN: 2224-266x, Issue 1, Volume 11, January 2012
- [7] Fabrice RECLUS, Kristen DROUARD, "Geofencing for Fleet and Freight Management," 2009 IEEE
- [8] Thomas Gonnot, Won-Jae Yi, Ehsan Monsef, Jafar Saniie, "Home Automation Device Protocol[HADP]:A Protocol Standard for Unified Device," Advances in Internet of Things, 2015, 5, 27-38
- [9] Nikos Vesyropoulos and Christos K. Georgiadis, "Customized QoS-based Mashups for the Web of Things: An Application of AHP," Computer science and information systems 12(1):115-13

Range Tree Based Indexing of Mobile Tracking System

Thu Thu Zan, Sabai Phyu
University of Computer Studies, Yangon
thuthuzan@ucsy.edu.mm, sabaiphyu@ucsy.edu.mm

Abstract

With advances in location-based services, indexing the need for storing and processing continuously moving data arises in a wide variety of applications. Some traditional spatial index structures are not suitable for storing these moving positions because of their unbalance structure. Searching an unbalanced tree may require traversing an arbitrary and unpredictable number of nodes and pointers. Presorting before tree structure is one of the ways of building a balanced two dimensional tree. In this paper, we proposed Presort Range tree that is suitable for moving objects with the dynamic range query. Moreover, with extending mobile technology, tracking the changing position of devices becomes a new challenge. The current location of each user would always be known at the server side whereas it would create a problem. If the mobile movements are small and frequent, at that time unnecessary updates would be performed at the server. In this paper, we also proposed Hybrid Update Algorithm to reduce the server update cost greatly.

Keywords- Location Update Policies, Location Based Service (LBS), Range Tree, Tracking, 2D Range Query

1. Introduction

Everyone who is in IT field says “Today is the age of three things: Cloud Computing, Internet of Things, and Mobile.” This word is true because there is no doubt that businesses can reap huge benefits from them [2].

In mobile technology, tracking moving objects are one of the most common requirements for many location management services. Since, the location of moving object changes continuously but the database location of the moving object cannot be updated continuously; therefore, an updating strategy for moving object is required.

In summary, in this paper we introduce the presort range tree for dynamic attributes whose main contributions are as follows.

- i. Presort Range tree structure is proposed for moving objects with the availability of dynamic range query.

- ii. Hybrid Update Algorithm is proposed that will help to get the current position of moving mobiles at client side and reduces the server update cost greatly.

We explain how to incorporate dynamic attributes in presort range tree and a model is added to deal with overall system. Finally, we made a comparison that will show the experimental result based on presort range tree and without tree. Furthermore, an experimental result of threshold value for proposed Hybrid Update algorithm is made with simulation.

2. The Range Tree

A tree data structure is a powerful tool for organizing data objects based on keys. It is equally useful for organizing multiple mobile objects in terms of hierarchical relationships. There is an assumption for mobile locations that no two points have the same x-coordinate and also y-coordinate. To construct spatial tree structure, the first thing is preprocessing the data into the data structure. Then, queries and updates on the data structure are performed. Then, we treat range query as 2 nested one-dimensional queries: $[x1,x2]$ by $[y1,y2]$. The first step is to ask for the points with x-coordinates in the given range $[x1,x2] \Rightarrow$ a set of subtrees. Then, instead of all points in these subtrees, only want those that fall in $[y1,y2]$. In figure, $P(u)$ is the set of points under u store those points in another tree $Y(u)$, keyed by the y-dimension.

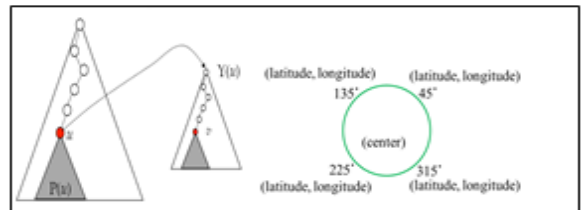


Figure 1: Structure of a Range Tree and Circular Range Searching

3. Location based services

Location based services (LBS) are services offered through a mobile phone and take into account the device's geographical location.

LBS typically provide information or entertainment. LBS largely depend on the mobile user's location. These services can be classified into two types: Pull and Push. In a Pull type, the user has to actively request for information. In a Push type of service, the user receives information from the service provider without requesting it at that instant [4].

3.1. Location Update Policies

Various location update strategies are available in the mobile computing. They are divided into specific strategies like (1) Distance Based Location Update (2) Time Based Location Update (3) Movement Based Location Update (4) Profile Based Location Update and (5) Deviation Based Location Update [1].

3.2. Getting Mobile Location Framework

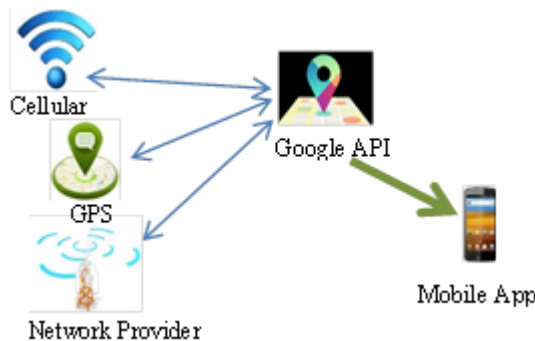


Figure 2: Getting location for Mobile Application

This system has to get current location as fast as it can so it has a framework shown in figure2. It helps to provide a more powerful location framework than usual. This framework is intended to automatically handle location provider's support, accurate location, and update scheduling. It includes the following features.

(a) GPS features → (GPS, AGPS):

- i. determines location using satellites.
- ii. does not need any kind of internet or wireless connection.
- iii. depending on conditions, this provider may take a while to return a location fix.

(b) Network provider → (AGPS, CellID, WiFi MACID):

- i. determines location based on availability of cell tower and WiFi access points.

- ii. results are retrieved by means of a network lookup.

(c) Cellular Network → (CellID, WiFi MACID):

- i. a special location provider for receiving locations without actually initiating a location fix.
- ii. although if the GPS is not enabled this provider might only return coarse fixes.
- iii. is mapped to the specific set of hardware and telecom provided capabilities

To shorten the time to first fix, or the initial positioning or increase the precision in situations when there is a low satellite visibility, the mobile network should be used. The best way is to use the “network” or “Cellular Network” provider first, and then fallback on “gps”, and depending on the task, switch between providers.

4. Related Works

There are a number of papers that describe about moving objects' index tree structure and mobile update policies. Most papers are focus on using one index structure and one update policy. Some discuss combination of index trees called hybrid tree structure and comparison of using one index structure and it.

Dongseop Kwon, Sangjun Lee, Sukho Lee proposed a novel R-tree based indexing technique called LUR-tree. This technique updates the structure of the index only when an object moves out of the corresponding MBR (minimum bounding rectangle). If a new position of an object is in the MBR, it changes only the position of the object in the leaf node. [5]. So, it remove unnecessary modification of the tree while updating the positions because this technique updates the index structure only when an object moves out of the corresponding MBR (minimum bounding rectangle).

Christian S. Jensen, Dan Lin, Beng Chin Ooi represented moving-object locations as vectors that are time stamped based on their update time. By applying a novel linearization technique to these values, it is possible to index the resulting values using a single B⁺tree that partitions values according to their timestamp and otherwise preserves spatial proximity. This scheme uses a new linearization technique that exploits the volatility of the data values, i.e., moving-object locations, being indexed. Algorithms are provided for range and _{NN} queries on the current or near-future positions of the indexed objects [4].

Yuni Xia, Sunil Prabhakar proposed a novel indexing structure, namely the Q+Rtree that is a hybrid tree structure which consists of both an R*tree and a QuadTree. In Q+R tree, quasi-static objects are stored in

an R*tree and fast-moving objects are stored in a Quadtree. By handling different types of moving objects separately, this index structure more accurately reflects the reality and results in better performance. In their work, no assumption is made about the future positions of objects. It is not necessary for objects to move according to well-behaved patterns and there are no restrictions, like the maximum velocity, placed on objects either [7].

Cheng, Pingzhi Fan, Xianfu Lei, and Rose Qingyang Hu made a location update scheme in which update occurs either when the movement threshold for MBLUs is reached or when the time threshold for TBLUs is reached. The movement counters and the periodic LU timer reset when an LU occurs. They used convex function of the movement threshold. That is, there is a value of the movement threshold that can minimize the signaling cost. They showed that the HMTBLU scheme always has higher signaling cost than the MBLU scheme [3].

Vicente Casares_Giner, Pablo Garcia-Escalera proposed a location update scheme by combining two dynamic strategies, movement based and the distance based [6]. They showed that results obtained from these analytical model show that, with little memory requirements in the mobile terminal very good performances can be obtained. However, it required that after each movement, the mobile terminal has to search the identity of the new visited cell in a cache memory.

5. Proposed Approach

This paper is integrated by two major components: client side and server side. The overall system model is built and hybrid update algorithm that will aid to get last current location and reduce server update cost is proposed at the client side. Presort range tree procedure for moving objects is included in the server side.

5.1. System Model

A model, searchable model is built to incorporate dynamic attributes in presort range tree and query processing. This includes a server and a collection of registered mobile objects. In order to keep the location information up to date, these objects regularly send their updated positions to the server. Unnecessary updates wouldn't be performed at the server because Hybrid Update Algorithm is applied to the client side. The require information query the server with range queries like "which mobiles are currently located within a disaster area?" To process such queries efficiently, the server maintains an index tree that, in addition to

speeding up the query processing, is also able to absorb all of the incoming updates.

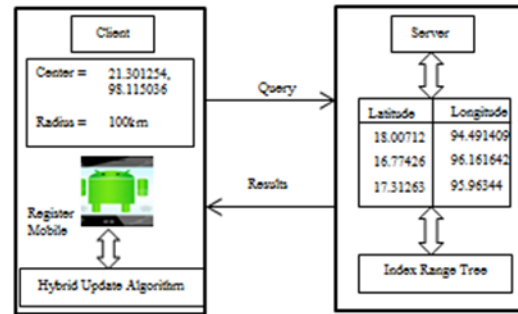


Figure 3: Client-Server System Model

5.2. Hybrid Update Algorithm

The distance-based update scheme seems to be simple location update strategy. In this scheme, each mobile host has to track the distance it moved since its last location update. When the distance exceeds the threshold (HD), the mobile host transmits an update message.

But it is complicated because of the variation of cell sizes and the need to compute the distance a mobile has moved.

The time based location update strategy is a simple strategy for location update. Here the mobile base station would update the location of user after a particular time period say T. However, the main drawback here would be sometimes if the user is stationary at that time unnecessary updates would be performed.

In this paper, hybrid location update algorithm is proposed based on time and distance so that it can significantly reduce location update overhead which improves the efficiency of mobility support mechanisms. The structure of mobile location update is shown in figure.

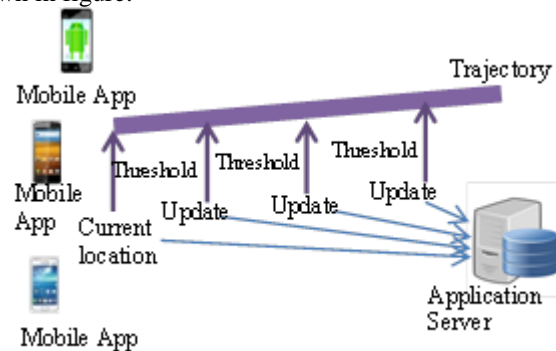


Figure 4: Mobile Location Update Structure

The advantage of the proposed algorithm is that it reduces location update traffic, with a minimum increase in implementation complexity.

Algorithm: Hybrid Location Update

Input: Database of mobile locations contains the locations of registered mobile with time

Output: current registered mobile location

1. $i=0$; dis_threshold; time_threshold; time_scheduler;
2. Read the current location (L_{xi}, L_{yi}, t_i) and previous location $(L_{xi-1}, L_{yi-1}, t_{i-1})$ of registered mobile location
3. If the time_scheduler > time_threshold && $\sqrt{(L_{xi}-L_{xi-1})^2 + (L_{yi}-L_{yi-1})^2} > \text{dis_threshold}$
- 4.3.1. Update the database with current location $(L_{xi-1}, L_{yi-1}, t_{i-1}) = \text{current location } (L_{xi}, L_{yi}, t_i), i+1$.
- 3.2. Total number of update=Total number of update+1;
5. Else current location $(L_{xi}, L_{yi}, t_i) = \text{previous location } (L_{xi-1}, L_{yi-1}, t_{i-1}), i=i+1$;

5.3. Proposed Presort Range Tree

The procedure of proposed presort range tree is the following;

Input: Lats=Array of two dimensional points sort on latitudes

Longs=Array of two dimensional points sort on longitudes

Procedure PRTree (Lats, Longs)

1. If Lats.length==1 then return new LeafNode(Lats[1]);
2. medium= [Lats.length/2];
3. Copy Lats[1....medium] to Lats_L and Lats[medium+1.... Lats.length] to Lats_R;
4. for $i=1$ to Longs.length do
5. if Longs[i].x <= Lats[medium].x then append Longs[i] to Longs_L ;
6. else append Longs[i] to Longs_R ;
7. root= new Node((Lats[medium].x),One D Range(Y));
8. root.left= PRTree(Lats_L , Longs_L);
9. root.right= PRTree(Lats_R , Longs_R);
10. return root;

5.3.1. Circular Range Search

After preprocessing of tree construction is done, the structure allows searching circular range query for mobile objects. To determines whether registered mobiles are in service area or not so that this system has to **get bounding coordinates** with center and service distance: (centerLat, centerLong, bearing, distance).

```
bearingRadians = Radians(bearing);
lonRads = Radians(centerLong);
latRads = Radians(centerLat);
maxLatRads = asin((sin(latRads) * cos(distance / 6371)
+ cos(latRads)
sin(distance / 6371) * cos(bearingRadians)));
maxLonRads = lonRads + atan2((sin(bearingRadians) *
sin(distance / 6371) cos(latRads)),(cos(distance / 6371) -
sin(latRads) * sin(maxLatRads)));
```

5.3.2. Example: Calculating Presort Range Tree with center and service distance

Firstly, sort the mobile locations by latitudes and longitudes.

Sort by X	Sort by Y
16.35099 96.44281	16.77923 96.03917
16.77923 96.03917	16.80958 96.12909
16.80958 96.12909	26.69478 96.2094
24.77906 96.3732	24.77906 96.3732
24.99183 96.53019	16.35099 96.44281
25.38048 97.87883	24.99183 96.53019
25.40319 98.11739	26.35797 96.71655
25.59866 98.37863	25.82991 97.72671
25.82991 97.72671	25.38048 97.87883
25.88635 98.12976	25.40319 98.11739
26.15312 98.27074	25.88635 98.12976
26.35797 96.71655	26.15312 98.27074
26.69478 96.2094	25.59866 98.37863

Then the Presort Range Tree is built and shows as the following;

```
25.40319 98.11739
LEFT: 16.80958 96.12909
LEFT: 16.35099 96.44281
RIGHT: 16.77923 96.03917
RIGHT: 24.99183 96.53019
LEFT: 24.77906 96.3732
RIGHT: 25.38048 97.87883
RIGHT: 25.88635 98.12976
LEFT: 25.59866 98.37863
```

RIGHT: 25.82991 97.72671
 RIGHT: 26.35797 96.71655
 LEFT: 26.15312 98.27074
 RIGHT: 26.69478 96.2094

The results of sample range search in centerLat, centerLng, distance: 26.693, 96.208, 1000km that are registered mobile locations to send notification as follows:

node (25.40319, 98.11739)
 RIGHT: node (24.99183, 96.53019)
 LEFT: node (24.77906, 96.3732)
 RIGHT: node (25.38048, 97.87883)
 RIGHT: node (25.88635, 98.12976)
 LEFT: node (25.59866, 98.37863)
 RIGHT: node (25.82991, 97.72671)
 RIGHT: node (26.35797, 96.71655)
 LEFT: node (26.15312, 98.27074)
 RIGHT: node (26.69478, 96.2094)

6. Simulation Results

The simulation considers an experimental result with threshold value for proposed Hybrid Update algorithm. These values inserted in the local database of the moving object. Then compute the distance and if the distance \geq a specific threshold, an update occur. In figure 6 that represent the actual and expected path through 9 minute at threshold = 2 miles. This result shows that central database needs to be update with actual location only five times at point a, b, c, d and e. Since the distance is greater than the value of threshold instead of updating the database every time.

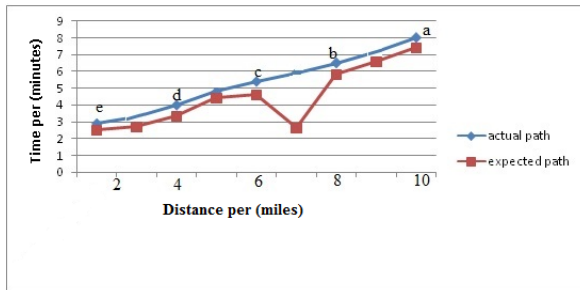


Figure 5. The actual and expected path at (threshold=2 miles)

The next experiment has been performed on a 2.60 GHz ASUS PC, with Intel (R) Core (TM) i7 CPU and 4 GB memory. For this experiment, we use the most popular testing framework in Java, JUnit. It is an open source testing framework which is used to write and run repeatable automated tests. The experiment was performed for computing query and execution time for range search, with number of data set points that are organized in two dimensions. To construct spatial tree

structure, the first thing is preprocessing the data into the data structure. Then, queries and updates on the data structure are performed. It has been used to compare the performance of both, Range tree and without Tree, for data set points in a 2-dimensional space. The execution time required by both approaches was different. Query found by both approaches with the same points. Better performance was achieved when the Range tree was used for larger number of data sets for range search. The more volumes of data tests, the less number of seconds needs in Range tree.

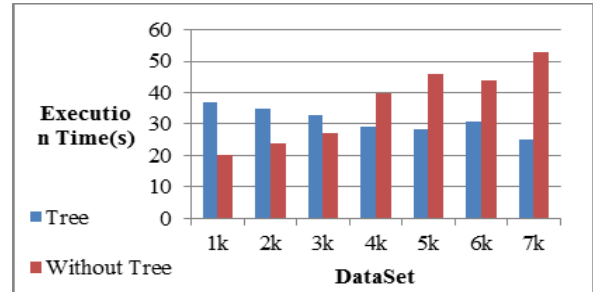


Figure 6. Execution Time (Preprocessing time +Query time) of Tree and Without Tree

7. Conclusion and Future Works

In this paper, the main service task is handling mobile objects based on index tree structure. The system maintains the moving mobile locations and circular range query is available from the server. Therefore, the system is done for monitoring of mobile objects, to be able to efficiently locate and answer queries related to the position of these objects in desire time. The system will helps to be tradeoff frequency of update due to the locations of mobile objects and reduce server update cost. It also support range query with dynamic object locations.

For future works, the proposed Hybrid Update approach will be applied to other index structures (e.g. the quad tree, the K-D-B tree). Moreover, this proposed system can be used to storing other moving objects such as temperature, vehicle location and so on. The results obtained from the other index tree structure can be compared to this paper's results.

8. References

- [1] A.Kalpesh A, S.Priyanka, "Various Location Update Strategies in Mobile Computing", International Journal of Computer Applications® (IJCA) (0975 – 8887) Proceedings on National Conference on Emerging Trends in Information & Communication Technology (NCETICT 2013)
- [2] Rundle, M.Huffington, "future of technology whitepaper", UK, 2015.

[3] Cheng, Pingzhi Fan, Xianfu Lei, And Rose Qingyang Hu, "Cost Analysis Of A Hybrid-Movement-Based And Time-Based Location Update Scheme In Cellular Networks", IEEE Transactions On Vehicular Technology, Vol. 64, No. 11, November 2015.

[4] Christian S. Jensen, Dan Lin, Beng Chin Ooi, "Query and Update Efficient B+-Tree Based Indexing of Moving Objects", VLDB 04 Proceedings of the Thirtieth international conference on Very large databases, Volume 30 pages 768-779.

[5] Dongseop Kwon, Sangjun Lee Sukho Lee, "Indexing the Current Positions of Moving Objects Using the Lazy Update

R-tree" Third International Conference on Mobile Data Management, IEEE, 2002.

[6] Vicente Casares_Giner, Pablo Garcia-Escalle, "An Hybrid Movement-Distance-Based Location Update strategy for Mobility Tracking", CICYT (Spain) for financial support under project number TIC2001-0956-C04-04.

[7] Yuni Xia Sunil Prabhakar, "Q+Rtree: Efficient Indexing for Moving Object Databases", Eighth International Conference on Database Systems for Advanced Applications (DASFAA '03), March 26-28, 2003, Kyoto, Japan.

Automatic Adjustment of Read Consistency Level of Distributed Key-value Storage by a Replica Selection Approach

Thazin Nwe¹, Tin Tin Yee¹, Myat Pwint Phyu¹, Ei Chaw Htoon², Junya Nakamura³
University of Information Technology, Myanmar¹, Computer University (Kyaing Tone), Myanmar²
Toyohashi University of Technology, Japan³
{thazin.nwe,tintinyee, myatpwintphyu}@uit.edu.mm¹, eichawhtoon@uit.edu.mm², junya@imc.tut.ac.jp³

Abstract

In distributed key-value storage systems, Apache Cassandra is known for its scalability and fault tolerance. In such systems, Cassandra is a peer-to-peer architecture which any user can connect to any node in any data center and can read and write data anywhere. Most of the systems usually select a fixed number of replicas for read/write requests in key-value storage. When the more replicas a read request chooses, it may increase the response time and reduce the system performance. In this paper, a consistent replica selection approach is proposed to automatically select number of consistent replicas by defining the read and write consistency level. This approach searches the nearest replicas and selects the consistent replicas depending on the current time, nearest arrival time, read/write latency and version for each read request. The proposed approach tends to achieve the read/write performance of client requests for key-value storage system by reducing the read/write execution time, latency cost and storage cost.

Keywords- Consistent Replicas, Consistency Level, Key-value storage

1. Introduction

Replication is a widely used technology in distributed key-value storage systems to achieve data availability, durability, fault tolerance and recovery. In these systems, maintaining data consistency of replication becomes a significant challenge. Although many applications benefit from strong consistency, latency sensitive applications such as shopping carts on e-commerce websites chooses eventual consistency. Eventual consistency is a weak consistency that does not guarantee to return the last updated value [5]. Eventually consistent systems are high operation latencies and thus in bad performance.

Achieving high throughput and low latency of responses to client requests is a difficult problem for cloud services. To fix these issues, a replica selection process needs to include mechanisms for filtering and

estimating the latency when processing requests. The replica selection process is inherently complicated.

Therefore, this paper proposes a replica selection approach for read access in distributed key-value storage systems. A key-value store is a simple database that uses an associative array as the fundamental data model where each key is associated with one and only one value in a collection. This relationship is referred to as a key-value pair.

This approach can determine the minimal number of replicas for reading request needs to contact in real time by defining the consistency levels (one, two, quorum, local quorum, etc.). Depending on these consistency levels, the system can choose the nearest consistent replicas using replica selection algorithms. By using these algorithms, the system will improve the read/write execution time on defining the consistency levels and reduce the read/write latency cost on choosing the nearest consistent replicas.

2. Related Works

Geo-distributed storage systems tend to forward client's requests towards the "close" replicas to minimize network delay and to provide the best performance. This task commonly occurs, e.g., in self organizing overlays. One of the primary tasks is to correctly compute or estimate the distance between the nodes; various systems has tackled this problem. Meridian et al. [4] is a decentralized, lightweight overlay network that can estimate the distance to a node in the network by performing a set of pings that are spaced logarithmically from the target. Kirill Bogdanov et al. [2] demonstrate the need for dynamic replica selection within a Geo-distributed environment in a public cloud. Second, they propose a novel technique of combining symbolic execution with lightweight modeling to generate a sequential set of latency inputs that can demonstrate weaknesses in replica selection algorithms. The sequential set of latency inputs is the consecutive latency that describes the network conditions.

According to [6, 7], there are two traditional mechanisms that can generally be used as how to implement consistency management in large scale

systems: an optimistic mechanism which does not immediately propagate changes and therefore tolerates replica content divergence, and Pessimistic mechanism prevents conflicts by blocking or aborting operations as necessary.

Harmony [1] is a system that can dynamically adjust replica consistency according to the application requirements. It proposes an estimation model to predict the stale read. By collecting read/write access frequency, network latency, most recent read/write access time and other information, it can predict the stale read ratio in real time and achieve the required consistency level with relatively good performance of elastically increase or decrease the number of replicas involved in each read request. Harmony uses a White box model, which decides the replicas numbers of each request by using mathematical formula derivation. To compute the number of replicas to be involved in a read operation necessary, this model finds the stale read rate smaller or equal to the defined threshold value. However, since there are so many factors that can impact the result and lots of those factors change in real time, such white box analysis may not get precise results. Besides, Harmony assumes the request access pattern meets Poisson process, however, different application' access patterns are different, which means Harmony has its usage limitation.

In most systems, it defines the rate of stale read that can be tolerated, and then try to improve system performance as much as possible while still not exceed such stale read rate. However ZHU, Y et al. [8] takes another mechanism, the longest response time is defined that it can tolerate and try to enhance the consistency level as much as possible within this time. The read/write access is broken into 6 steps: reception, transmission, coordination, execution, compaction and acquisition, and each of which can further break into smaller steps. Then a linear regression is used to predict the execution time and latency of the next request for each step. When a request comes, it maximizes the number of steps this request covers within the tolerated time, thus achieves the maximize consistency. However, the stale read rate of this system is unpredictable.

P.Bailis et al. [3] introduces Probabilistically Bounded Staleness (PBS) consistency. PBS describes two ways to estimate the staleness of data: version based and time based. Firstly, a closed-form solution is derived for version based data staleness. Then it models time based staleness and applies it in Dynamo style systems [11]. PBS uses Monte Carlo simulation to

describe the time based data staleness. The paper is inspired by PBS and uses the same Monte Carlo simulation to estimate the minimal replica number a read request needs to contact in order to get a specific fresh data rate. An adaptive replica selection algorithm [9] determines minimal number of replicas for each read request needs to select in order to achieve a specific consistency level by estimating the time interval between current read request and nearest write request. However, this algorithm doesn't consider the version based staleness. W.golab et.al [10], proposed the methods of quantifying the consistency in eventually consistent storage systems. That paper described the comparisons of the staleness methods for the stale read problems and issues of Probability of Bounded Staleness (PBS) that does not consider workloads where writes overlap in time with reads.

3. Read/Write consistency level of Cassandra

Cassandra offers tunable data consistency across a database cluster. This means a developer or administrator can decide exactly how strong (e.g., all nodes must respond) or eventual (e.g., just one node responds, with others being updated eventually).

This tunable data consistency is supported across single or multiple data centers, and a developer or administrator has many different consistency options from which to choose. Moreover, consistency can be handled on a per operation basis, meaning a developer can decide how strong or eventual consistency should be per SELECT, INSERT, UPDATE, and DELETE operation.

Cassandra provides automatic data distribution across all nodes that participate in a "ring" or database cluster. There is nothing programmatic that a developer or administrator needs to do or code to distribute data across a cluster. The data is transparently partitioned across all nodes in either a randomized or ordered fashion, with random being the default. Cassandra also provides built-in and customizable replication, which stores redundant copies of data across nodes that participate in a Cassandra ring. This means that if any node in a cluster goes down, one or more copies of that node's data are available on other machines in the cluster.

Unlike complicated replication schemes in various RDBMSs or other NoSQL databases, replication in Cassandra is extremely easy to configure. A developer or administrator simply indicates how many data copies are desired, and Cassandra takes care of the rest. Replication options are also provided that allow for data to be automatically stored in different physical racks (thus ensuring extra safety in case of a full rack

hardware failure), multiple data centers, and cloud platforms [14].

Data consistency is the synchronization of data on all its replicas in the cluster. The number of replicas that need to acknowledge the write request to the client application is determined by write consistency level and the number of replicas that must respond read request before returning data to the client application is specified on the reading consistency level. Consistency levels can be set globally or on a per-operation basis. Few of the most used consistency levels are stated below:

- ONE

A response from one of the replica nodes is sufficient.

- Quorum

A response from a quorum of replicas from any data center. The quorum value is found from the replication factor by using the formula. $Quorum = (Replication\ Factor / 2)$.

- All

All nodes play equal roles; with node communicative with each other equally. There is no master node so there is no single point of failure and all the data has copies in other nodes which secures the data stored. It is capable of handling large amounts of data and thousands of concurrent users or operations per second across multiple data centers.

4. Proposed System

In this architecture, a client writes a file to the replicas as the write consistency level. On Cassandra, the read and write consistency levels (e.g., one, two, quorum, local quorum; etc.) can be defined. In a cluster with a replication factor of three, and the read/write the consistency level of quorum, the two of the three replicas have to respond to read/write requests. Consistency level describes the behavior seen by the client. Writing and reading at quorum level allows strong consistency.

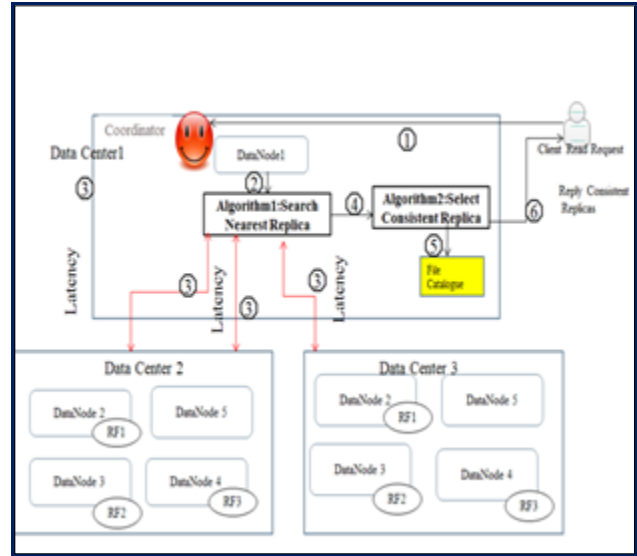


Figure 1. Consistent Replica Selection Architecture for reading request in key-value storage

Erasure coding (EC) is a method of data protection in which data is broken into fragments and encoded with redundant data pieces and stored across a set of different locations or storage media. The goal of erasure coding is to enable data that become corrupted at some point in the disk storage process to be reconstructed by using information about the data that's stored elsewhere in the array.

In figure 1, the write requests are incoming to the coordinator node. The coordinator node performs the erasure-encoding that divides the data block into m fragments and encode them into n fragments. For example, the incoming data is 5MB; it is split into same size for each MB. And two more 1MB parity pieces are added for redundancy. Therefore, the original block is divided into five fragments and then stored on five separate Cassandra key-value storage nodes and two redundant nodes. In this case, the proposed system writes totally $7 * 7$ (write consistency level * fragments) into a cluster. The fragments created are saved by consistent hashing [12] on different quorum nodes. The acknowledgement of successful writes is sent to the coordinator node.

Secondly, when the client reads a file, it sends a read request to the coordinator Node. The Coordinator Node collects the list of DataNodes that it can retrieve data by using the replica selection algorithm described in the next section. When sufficient fragments have been obtained, the Coordinator Node decodes the data and supplies it to the read application request. In this case

where a client reads from the cluster the file with the read consistency level of five. Therefore, the coordinator of the read request retrieves 5*5 (read consistency level *fragments) from data nodes. If two of five nodes fail, the data cannot be lost.

5. Algorithm Definition

The replica selection algorithm has two parts. It includes (i) searching nearest replica and (ii) selecting consistent replica. In algorithm_1, the coordinator node sends the request message to each replica and latencies of different replicas are listed in the read latency map. And it chooses the lowest latency of replica from this map.

In algorithm_2, the replica selection algorithm in the coordinator node chooses the consistent replica from nearest replicas.

1. **Input:** Replicas RF= { RF₁, RF₂,...RF_n }
2. **Output:** Nearest Replica NR
3. **Set** latencyCost= MAX_VALUE;
4. **Set** lowestLC []=null; //Initialize return lowest latency replica
5. **For** each r in RF // RF=Replicas
6. **Begin**
7. **Set** latencyCost=getLatencyCost(RF_r, job);
8. **If**(latencyCost<=MAX_VALUE)Then// MAX_VALUE =threshold values
9. MAX_VALUE =latencyCost;
10. lowestLC.add (RF_r);
11. **End**
12. **End for**
13. **Return** lowest LC //nearest replica NR

Algorithm 1: Search nearest Replica

Search nearest Replica part executes in two stages. First, all replicas are sorted based on their physical location, so that all replicas in the same rack and then the same datacenter as the source are at the top of the list. Second, the latencies are computed from the local node (originator of the query) to all other nodes. If the latency cost is greater than a threshold of the closest node, then all replicas are sorted based on their latency costs. Finally, the top replicas from the list are chosen.

Firstly, total numbers of replica are listed as input (line1). The threshold value is set at the latency cost of line3. In line8, the coordinate node contacts every other replica with request messages. The round trip time it takes from the request until the reply is passed through $T_{total} = RTT_{request}/2 + T_{processing} + RTT_{reply}/2$. T_{total} is used to

get the latency cost of computing data nodes in algorithm1. These costs are used when the local node needs to forward client requests to other replicas.

And then total times taken from different replicas are listed in latencyCost (line8). Finally algorithm1 returns the list of lowest latency cost of the replicas in lowestLC as output to client. (line14).

1. **Input:** Nearest Replica NR= {NR₁,NR₂,...NR_n}
2. **Output:** Consistent Replicas
3. **For** each Nearest Replica NR_i
4. **Begin**
5. **Set** RCL=2//ConsistencyLevel.QUORUM
6. **Set** noOfConsistentRead=0
7. **While**(noOfConsistentRead<=RCL)
8. **If**(stalerate<=maxStaltrate)Then
9. consistentRead.add(NR_i)
10. noOfConsistentRead++;
11. **Return** consistentRead;
12. End for
13. End

Algorithm 2: A Consistent Replica Selection

In algorithm_2, the set of the nearest replicas is collected as input that comes from output of algorithm_1 by computing latency costs. And then algorithm_2 sets the read consistency level (RC) that the client will need the most up-to-date information. Read/Write latencies of different replicas are listed in history file on the coordinator node.

This algorithm determines the number of consistent replica nodes, one read request should select in real-time, according to calculate arrival times of nearest update request and the processing order of read request and write request in different replicas.

For computing stale rate of algorithm_2, a quorum system obeys PBS k-staleness consistency if with probability $1-p_{sk}$; at least one value in any read quorum has been committed within k versions of the latest committed version when the read.

$$p_{sk} = \left(\frac{\binom{N-W}{R}}{\binom{N}{R}} \right)^k \quad (1)$$

In eq. 1, When N=3, R=W=1, this means that the probability of returning a version within 2 versions is 0.5, within 3 versions is 0.703, within 5 versions is > 0.868, and within 10 versions is > 0.988.

When N=3, R=1, W=2 (or, equivalently, R=2, W=1), these probabilities increase: k=1 -> 0.6, k=2 -> 0.8, and k=5 > 0.995.

A quorum system obeys PBS (k,t)-staleness consistency if, with probability $1 - p_{sk}$, at least one value in any read quorum will be within k versions of the latest committed version when the read begins, provided the read begins t units of time after the previous k versions commit. A quorum system obeys PBS k-staleness consistency with probability $1 - p_{sk}$ where p_{sk} is the probability of non-intersection with one of the last k independent quorums.

$$p_{st} = \frac{\binom{N-W}{N}}{\binom{N}{R}} + \sum_{c \in \{W, N\}} \frac{\binom{N-c}{N}}{\binom{N}{R}} \cdot [P_w(c+1, t) - P_w(c, t)]$$

(2)

The above equation makes several assumptions. Reads occur instantly and writes commit immediately after W replicas have the version. T-staleness in real systems depends on write latency and propagation speeds.

6. Analysis of read/writes execution time

The read/write execution time of consistency level is tested by using Cassandra cluster on VMware Ubuntu 14.04 LTS i386. The processor is Intel(R) Core(TM) i7-4770 CPU @ 3.40 GHz. Installed memory (RAM) is 4.00GB as shown in table1.

Table1. Hardware Specification and Virtual Environment

Operating System	VMware Ubuntu 14.04 LTS i386
RAM	4.00GB
Hard-disk	195GB
Processor	Intel(R) Core(TM) i7-4770 CPU @ 3.40 GHz
Cassandra	version: 1.0.6

The staff data from Ministry of Higher Education is used on Cassandra cluster. Staff information is described by Unicode in "staff.csv".

When importing data from the csv to Cassandra, java hector code truncate the input csv data with a comma (",") line by line. And then the output csv data are exported on Cassandra.

Figure2 shows Cassandra supports Unicode, but, Hbase does not support it. Therefore, staff data can be tested on Cassandra cluster. Unicode is the international accepted standard by the World Wide Web Consortium, the main international standards organization for the World Wide Web. And it also makes that it is extremely easy to translate the Wikipedia's interface. And Unicode

fonts support 11 languages that use the Myanmar script: Burmese, 2 liturgical languages: Pali and Sanskrit, 8 minority languages: Mon, Shan, Kayah, four Karen languages and Rumai Palaung [13]. It was officially released by Myanmar Natural Language Processing (NLP) Research Center joining existing Myanmar Unicode 5.1.

```

rowkey: sample1
> (column=country, value=, timestamp=1505188206076000)
> (column=addressLine1, value=, timestamp=1505188206076001)
> (column=addressLine1, value=, timestamp=1505188206076002)
> (column=alias, value=၀၀၁၀၀၀၀၀၀, timestamp=1505188206076003)
> (column=armyForce, value=၀, timestamp=1505188206076004)
> (column=city, value=၀၀, timestamp=1505188206077000)
> (column=city1, value=, timestamp=1505188206085000)
> (column=country, value=၀၀၀၀၀, timestamp=1505188206085001)
> (column=country1, value=၀၀ Min Thant, timestamp=1505188206085002)
> (column=dateOfBirth, value=, timestamp=1505188206085003)
> (column=degree, value=, timestamp=1505188206085004)
> (column=duty, value=၀၀၀၀၀၀၀, timestamp=1505188206086000)
> (column=email, value=၀၀၀, timestamp=1505188206086001)
> (column=enddate, value=၀၀၀၀၀, timestamp=1505188206086002)
> (column=englishName, value=၀/၀၀၀၀၀), timestamp=1505188206086003)
> (column=eyeColor, value=, timestamp=1505188206086004)
> (column=hairColor, value=၀/၀, timestamp=1505188206086005)
> (column=height, value=၀၀၀၀၀၀, timestamp=1505188206086006)
> (column=job, value=၀၀၀၀၀၀၀၀၀၀၀, timestamp=1505188206086007)
> (column=mark, value=၀၀၀၀၀၀၀၀၀, timestamp=1505188206086008)
> (column=phoneNumber, value=၀၀၀၀၀၀၀, timestamp=1505188206086009)

```

Figure2. Unicode on Cassandra Cluster

Ubuntu 14.04 LTS is installed on three servers and one client by Cassandra clusters. There are 203 rows and 40 columns from csv file are inserted into one of Cassandra servers and replicate it interconnected other servers. And Replication Factor (RF), Read Consistency Level (RCL) and Write Consistency Level (WCL) are defined by changing the consistency level (one, two and quorum) and tested by Java hector code. Write execution time of servers and read time of the client according to consistency level are shown in figure3. According to this figure, the read/write execution time of consistency level (quorum) is better than consistency level (one and all).

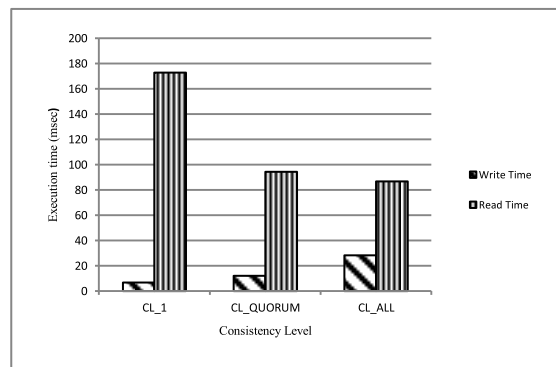


Figure3. Read/Writes execution time of consistency level (one, two and quorum)

7. Conclusion

The paper presents the performance of the consistency level (one, all and quorum) for read/write requests in Cassandra key-value data storage. In defining these consistency levels, a replica selection

approach is proposed for choosing the consistent replicas in different clusters by searching the nearest replica and selecting the consistent replica. For a specific application, its read/write access pattern, network latency and system load always change dynamically. Therefore, at different time, to reach the same consistency level, the impact on system performance is different. In this approach, the arrival time of read/writes request latencies, timestamp and versions are used to choose the consistent replicas near the clients. And this approach can determine the minimal number of replicas for reading request needs to contact in real time and thus improve the system performance as a result of reduced read/write execution time, latency cost and disk storage cost.

8. Future Work

In the future, the proposed algorithms will be validated on Cassandra clusters. And predicted t-visibility and latency will be compared with measured values and will compute the stale read rate for consistent replicas by adding more nodes and more dataset size on Cassandra and MongoDB distributed key-value data storage. And read/writes execution time, latency cost and storage cost of the system will be compared with the existing system.

9. References

- [1] H. Chihoub, S. Ibrahim, G. Antoniu and M. S. Perez, "Harmony: Towards Automated Self-Adaptive Consistency in Cloud Storage", IEEE International Conference on Cluster Computing, September 24-28; Beijing, China , 2012.
- [2] K. Bogdanov, M. Pe'ón-Quir'os, Gerald Q. Maguire Jr., Dejan Kosti'c, "The Nearest Replica Can Be Farther Than You Think", ACM 978-1-4503-3651-2/15/08, 2015.
- [3] P. Bailis, S. Venkataraman, J. M. Hellerstein, M. Franklin and I. Stoica. "Probabilistically Bounded Staleness for Practical Partial Quorums", Proceedings of the VLDB Endowment. 5, 8 , 2012.
- [4] B. Wong, A. Slivkins, SIRER and E. G. Meridian, "A lightweight network location service without virtual coordinates", in ACM SIGCOMM Computer Communication Review, vol. 35, ACM, pp. 85–96, 2005.
- [5] W. Vogels, "Eventually consistent", CACM, 52:40–44, 2009.
- [6] Y. Saito and H. M. Levy, "Optimistic replication for internet data services", in International Symposium on Distributed Computing, pages 297–314, 2000.
- [7] Y. Saito and M. Shapir, " Optimistic replication", ACM Comput. Surv., 37(1):42–81, 2005.
- [8] Y. Zhu and J. Wang. Malleable, "Flow for Time-Bounded Replica Consistency Control", OSDI Poster, October 8-10; Hollywood, USA , 2012.
- [9] Z. Ye and Weijian, "An Adaptive Replica Selection Algorithm for Quorum based Distributed Storage System", International Journal of Grid and Distributed Computing Vol.9, No.5, 2016.
- [10] W. Golab, Muntasir R. Rahman. et.al, "Eventually Consistent: Not What You Were Expecting?", Volume 12 Issue 1, January 2014 CM.
- [11] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall and W. Vogels, "Dynamo: Amazon's Highly Available Key-value Store", ACM 978-1-59593-591-5/07/0010, 2007.
- [12] P. Garefalakis , P. Papadopoulos, I. Manousakis, and K. Magoutis, "Strengthening Consistency in the Cassandra Distributed Key-Value Store", International Federation for Information Processing 2013.
- [13] [https://wikivisually.com/wiki/Burmese_\(language\)](https://wikivisually.com/wiki/Burmese_(language))
- [14] A. Basith, "Introduction to Apache Cassandra", April 22, 2017.

Workshop Session

An Approach of Accessing Small Files on HDFS for Cloud Storage

Khin Su Su Wai, Julia Myint, Tin Tin Yee

University of Information Technology, Yangon, Myanmar

khinsusuwai@uit.edu.mm, juliamyint@uit.edu.mm, tintinyee@uit.edu.mm

Abstract

Hadoop distributed file system (HDFS) was originally designed for large files. When the large number of small files is accessed, NameNode often becomes the bottleneck. The access and storage efficiency is low for the mass small files. In order to solve this issue in HDFS, the approach of accessing small files on HDFS is proposed. In this approach, all correlated small files will be merged into a larger file based on the agglomerative hierarchical clustering mechanism to reduce NameNode memory space. Moreover, HDFS does not take the correlation among files into account and it does not provide any prefetching mechanism to improve the I/O performance. A prefetching mechanism will be proposed to improve the efficiency of accessing small files. This approach will provide small files for cloud storage.

Keywords- Hadoop distributed files system (HDFS), NameNode, Small files, prefetching, and hierarchical agglomerative clustering

1. Introduction

Cloud computing has become increasingly popular as the next infrastructure for hosting data and deploying software and services. Distributed file system is the foundation of cloud computing and provides reliable and efficient data storage for upper applications.

Today most popular storage system for cloud computing such as Google File System (GFS) and Hadoop Distributed File System (HDFS) are widely used and well known. However, HDFS is more light-weighted and open-source platform.

Hadoop is a software framework which is an open source that supports big data in distributed environment. Hadoop creates a cluster of machines and coordinates the work among them. It has two major components HDFS and Map Reduce. HDFS has master-slave architecture, with a single master called the NameNode and multiple slaves called DataNodes. NameNode manages the metadata and regulates client accesses. The metadata is maintained in the main memory of the NameNode to ensure fast access to the client, on read/write requests. DataNodes provide block storage and service read/write requests from clients, and

perform block operations by contacting with NameNode.

The consumption of memory in NameNode is decided by the number of files stored in HDFS. Each file Metadata requires 150 bytes of space. DataNode is responsible for saving the real and replicated data. Each file is split into several blocks with the size of 128MB. These files are replicated in HDFS based on the configuration. The DataNode keeps on sending the heartbeat signal to the NameNode at regular intervals to indicate its existence in the system. The heart-beat consists of DataNode's capacity, used space, remaining space and some other information. Client can upload, download, update, and delete by contacting with NameNode.

A small file is a file whose size is less than the HDFS block size. Although the size of a file is less than the HDFS default block size, HDFS creates it as one block. When the large number of small files is stored, HDFS is inefficient because of high memory usage and unacceptable access cost. Hadoop does not provide optimal performance for small files processing. Furthermore, HDFS does not take the correlation files and it does not support any prefetching mechanism to improve the I/O performance. In this proposed approach, merging and prefetching will be presented to overcome small file problems in HDFS.

The rest of the paper is organized as follows. Section 2 is an illustration of related works about the proposed topic. Section 3 is the research methodology of the proposed system. Section 4 proposes the proposed system. Section 5 presents the expected result. Section 6 concludes the paper.

2. Related Works

MENG Bing and et al. [1] provided a solution to reduce NameNode memory consumption, by TLB-Map File. TLB-MapFile merges massive small files into large files by MapFile mechanism to reduce NameNode memory consumption and add fast table structure (TLB) in DataNode, and to improve retrieval efficiency of small files. A challenging work is to build up suitable TLB refresh cycle.

Zhipeng Gao and et al. [2] defined Logic File Name (LFN) and proposed the Small file Merge Strategy Based LFN (SMSBL). SMSBL is a new idea and a new

perspective on hierarchy; it improves the correlation of small files in the same block of HDFS effectively based different file system hierarchy. This system solved small file problem in HDFS and has appreciable high hit rate of prefetching files. The proposed system needs to combine SMSBL with other great solution to improve performance of HDFS.

A new structure for HDFS (HDFSX) is presented by Passent M EIKafrawy and et al. [3] to avoid higher memory usage, flooding network, requests overhead and centralized point of failure of the NameNode. In the other word, the performance analysis of the systems is needed to be developed.

Tao Wang and et al. [4] defined a user access task. The correlations among the access tasks, applications and access files are constructed by the improved PLSA, and the research object is transferred from file-level to task-level. Then, an effective strategy is proposed to improving small file problem in distributed file system. The strategy merges small files in term of access tasks and selects a prefetching targets based on the transition of the tasks. This strategy reduces the MDS workload and the request response delay.

Parth Gohil and et al. [5] focused on a MapReduce approach to handle small files. This approach improves the performance of Hadoop in handling of small files by ignoring the files whose size is larger than the block size of Hadoop. This also reduces the memory required by NameNode to store these files. So, it requires very less amount of memory than original HDFS but it requires some more memory than HAR and Sequence.

Extended Hadoop Distributed File System (EHDFS) is used by Tanvi Gupta and et al. [6]. This paper focuses on increasing the ‘efficiency’ of the indexing mechanism for handling ‘Small files’ on HDFS. This also added the concept of ‘Avatar node’ that eliminates the single point of failure.

Kyoungsoo Bok and et al. [7] proposed a distributed cache management scheme that considers cache metadata for efficient accesses of small files in Hadoop Distributed File Systems (HDFS). The proposed scheme can reduce the number of metadata managed by a NameNode. Many small files are merged and stored in a chunk. It also reduces unnecessary accesses by keeping the requested files using clients and the caches of data nodes and by synchronizing the metadata in client caches according to communication cycles.

Yonghau Huo and et al. [8] used additional hardware named SFS (Small File Server) between users and HDFS to solve the small file problem. This approach includes a file merging algorithm based on temporal continuity, an index structure to retrieve small files and a prefetching mechanism to improve the performance of file reading and writing.

3. Background Theory

3.1. Hierarchical Clustering

Hierarchical clustering is a widely used data analysis tool. The idea is to build a binary tree of the data that successively merges similar groups of points. This tree provides a useful summary of the data. Hierarchical clustering only requires a measure of similarity between groups of data points. The hierarchies also involve ordering relations.

A hierarchical classification can be illustrated in several ways. The result of hierarchical clustering is a tree-based representation of the objects, which is also known as dendrogram. The dendrogram is a multilevel hierarchy where clusters at one level are joined together to form the clusters at the next levels. This makes it possible to decide the level at which to cut the tree for generating suitable groups of a data objects.

In data mining and statistics, hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters. Strategies for hierarchical clustering generally fall into two types: agglomerative and divisive. In this system, the agglomerative hierarchical clustering is focused to merge many small files. A measure of dissimilarity between sets of observations is required in order to decide which clusters should be combined. In most methods of hierarchical clustering, this is achieved by use of an appropriate matrix and a linkage criterion which specifies the dissimilarity of sets as a function of the pairwise distances of observations in the sets.

3.1.1. Agglomerative Hierarchical Clustering:

Agglomerative hierarchical clustering is a clustering algorithm that builds a cluster hierarchy from the bottom-up. It starts by adding a cluster for each of the data points to be clustered, followed by iterative pair-wise merging of clusters until only one cluster is left at the top of the hierarchy. The choice of clusters is decided to merge at each iteration base on a distance metric.

The dissimilarity values between one file and another have to be calculated in advance. In this paper, the Euclidean distance measure is used to cluster the related files. The clustering is based on distance matrix. Only the half of the matrix is needed because the distance between objects is symmetric. The Euclidean distance measure is:

$$d(i, j) = \sqrt{(|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \dots + |x_{ip} - x_{jp}|^2)} \quad (1)$$

This formula defines data objects i and j with a number of dimension equal to p. The distance between the two data objects d(i,j) is expressed as given the

above formula x_{ip} is the measurement of object i in dimension p .

An agglomerative clustering algorithm is described in the single-linkage clustering. The single-linkage clustering is the minimum distance between elements of each cluster. Two clusters are merged if there is at least one edge which connects them.

4. Proposed System

HDFS has been adopted to support the Internet applications because of its reliable, scalable and low-cost storage capability. It is a file system that supports for cloud storage. However, it does not present good storage and access performance when processing a huge number of small files. Firstly, all correlated small files will be clustered into a large file to reduce NameNode memory consumption. The small file is a file, whose size is less than 75% of default block size (128MB). Secondly, a prefetching mechanism will be introduced to improve the efficiency of accessing small files. The new HDFS structure will support a new approach. The proposed system architecture is shown in figure 1.

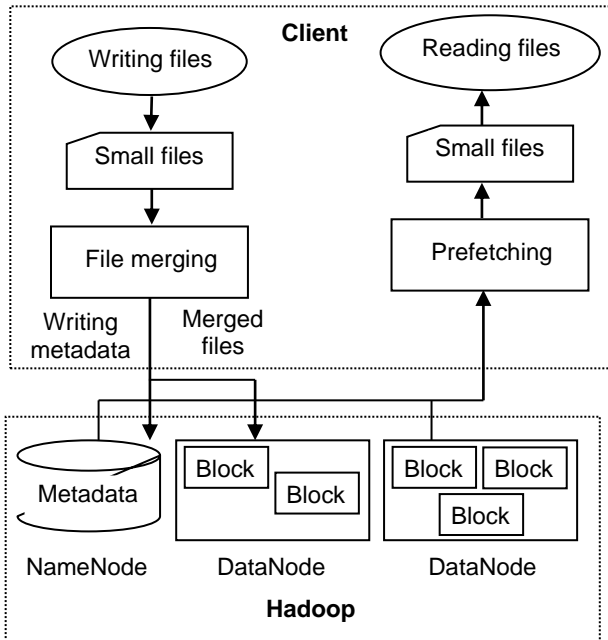


Figure 1. Proposed system architecture

4.1 File Merging Operation

File merging operations are carried out in processing layer. The related small files are merged into a large file. NameNode only maintains the metadata of merged files and does not store the original small files. File merging reduces the length of the metadata of many small files.

The data access and latency can be improved. Algorithm of file merging is as follow.

Algorithm of file merging

Input: The number of small files

Output: The number of clusters hierarchy

Method:

- (1) Calculate the Euclidean distance matrix between Small files
- (2) Repeat
- (3) Merge a pair of file with single-linkage based on the distance matrix
- (4) Update the distance matrix
- (5) Until size of cluster is greater than or equal to default block size

In this approach, the files will be ignored to merge as they are already larger than the threshold value. The default threshold for this system is set to (0.75) 75% of default block size (128 MB). If the user accesses the files, the system will check the size of file. If the file is a small file, the system will put it in the hierarchical structure. The small files are nested in large cluster of files. These larger clusters are joined until its size is less than the default block size. Thus, a small file belongs to many clusters depending on its size and threshold value. Otherwise, the file is directly operated in the original HDFS.

The following table is the sample input to trace a hierarchical clustering of distances in sizes between files.

Table 1. Sample input size of small files

File	Size(MB)
S1	40
S2	10
S3	50
S4	20
S5	60
S6	30

The Euclidean distance matrix of small files in table 1 is shown in table 2.

Table 2. Euclidean distance matrix of small files

	S1	S2	S3	S4	S5	S6
S1	0	30	10	20	20	10
S2	30	0	40	10	50	20
S3	10	40	0	20	10	20
S4	20	10	20	0	40	10
S5	20	50	10	40	0	30
S6	10	10	20	20	30	0

According to the file merging algorithm, the sample input from table 1 will have two clusters output. The file merging dendrogram of small files is shown in figure 3.

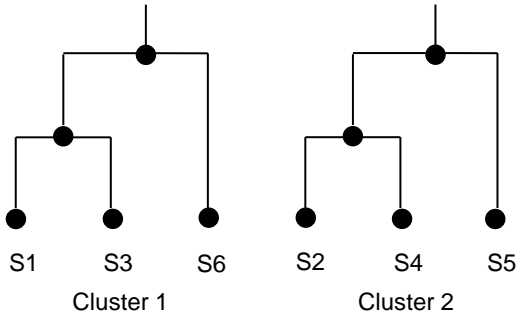


Figure 3. File merging dendrogram of small files

4.2 Prefetching

Prefetching scheme is widely used for improving the access efficiency. Prefetching can avoid disk I/O cost and reduce the response time by considering the access locality and fetching data into cache before they are requested.

The correlated small files are merged in a larger one file. When the user reads a small file, the system would need to find in hierarchical structure. Otherwise, the user has to read the file by connecting the DataNode. Prefetching schema helps in metadata caching and fetches small files. This technique optimizes fetching small files and reduces the communication cost when reading huge number of small files.

In the proposed paper, metadata and blocks of correlated files will be prefetched from NameNode and DataNodes respectively. The metadata prefetching reduces access latency on metadata server. The blocks prefetching is used to reduce visible I/O cost.

5. Expected Result

The proposed algorithm will be implemented and tested in simulated cloud environment. In the proposed paper, the strategy of merging small files will reduce the usage of metadata stored on NameNode. The processing time will also be optimized by comparing the original Hadoop and existing approach. Client will achieve better performance on accessing small files into HDFS.

6. Conclusion

In this paper, the small file problem of HDFS is focused. The memory space will be reduced to store the metadata of many small files in NameNode by the proposed agglomerative hierarchical merging approach. The number of small files will be eliminated in the HDFS by merging into a large file. The proposed

method for small files will provide the infrastructure of Hadoop. As a future work, many experiments have to be done in order to get the efficiency of proposed merging algorithm. Proposed prefetching mechanism has to be verified as a future work to improve the access of massive small files.

7. References

- [1] MENG Bing and GUO Wei-bin and FAN Gui-sheng, "A Novel Approach for Efficient Accessing of Small Files in HDFS: TLB-MapFile", 2016 IEEE SNPD 2016, Shanghai, China, May 30-June 1, 2016.
- [2] Zhipeng Gao, Yinghao Qin and Kun Niu, "AN EFFECTIVE MERGE STRATEGY BASED HIERARCHY FOR IMPROVING SMALL FILE PROBLEM ON HDFS", 2016 IEEE.
- [3] Passent M EIKafrawy, Amr M Sauber and Mohamed M Hafez, "HDFSX: Big Data Distributed File System with Small Files Support", 2016 IEEE.
- [4] Tao Wang, Shinhong Yao, Zhengquan Xu, Lian Xiong, Xin Gu and Xiping Yang, "An effective strategy for improving small file problem in distributed file system", 2015 IEEE, 2015 2nd International Conference on Information Science and Control Engineering.
- [5] Parth Gohil, Bakul Panchal and J. S. Dhobi, "A Novel Approach to Improve the Performance of Hadoop in Handling of Small Files", 2015 IEEE.
- [6] Tanvi Gupta and Prof. SS Handa, "An Extended HDFS with an AVATAR NODE to handle both small files and to eliminate single point of failure", 2015 IEEE, 2015 International Conference on Soft Computing Techniques and Implementations- (ICSTI) Department of ECE, FET, MRIU, Faridabad, India, Oct 8-10, 2015.
- [7] Kyongsoo Bok, Hyunkyo Oh, Jongtae Lim and Jaesoo Yoo, "An Efficient Cache Management Scheme for Accessing Small Files in Distributed File Systems", 2017 IEEE, BigComp 2017.
- [8] Yonghau Huo, Zhihao Wang, XiaoXiao Zeng, Yang Yang, Wenjing Li, ZHONG Cheng, "SFS: A Massive small file processing middleware in Hadoop", IEICE-The 18th Asia-Pacific Network Operations and Management Symposium (APNOMS) 2016.

Resource-based Data Placement Strategy for Hadoop Distributed File System

Nang Kham Soe, Tin Tin Yee, Ei Chaw Htoon

University of Information Technology, Yangon, Myanmar

nangkhamsoe@uit.edu.mm, tintinyee@uit.edu.mm, htoon.eichaw@gmail.com

Abstract

Big-Data is a term for data sets that are so large or complex that traditional data processing tools are inadequate to process or manage them. Apache Hadoop is an open-source software framework for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. The default Hadoop data placement strategy works well in homogeneous cluster. But it performs poorly in heterogeneous clusters because of the heterogeneity (in terms of processing, memory, throughput, I/O, etc.) of the nodes capabilities. It may cause load imbalance and reduce Hadoop performance. Therefore, Hadoop Distributed File System (HDFS) has to rely on load balancing utility to balance data distribution. The utility consumes the cost of extra system resources and running time. As a result, data can be placed evenly across the Hadoop cluster. But it may cause the overhead of transferring unprocessed data from slow nodes to fast nodes because each node has different computing capacity in heterogeneous Hadoop cluster. In order to solve these problems, a data/replica placement algorithm based on storage utilization and computing capacity of each data node in heterogeneous Hadoop Cluster is proposed. The proposed policy can balance the workload as well as reduce overhead of data transmission between different computing nodes.

Keywords- HDFS, Data Placement Policy, Load Balancing.

1. Introduction

Hadoop is one of the platforms for big data. Hadoop includes two parts, namely, the MapReduce and Hadoop Distributed File System (HDFS). MapReduce [4] is a software framework to process large data. It includes two phases: Map phase and Reduce phase. Data is split into small parts so that many map tasks can process them simultaneously. The results of map tasks are shuffled and merged by reduce tasks. HDFS is a storage part of Hadoop. It has two kinds of nodes, where a namenode serves as the master node and datanodes as slave nodes.

In HDFS, each data file is stored as a sequence of blocks, the blocks of a file are replicated for reading performance and fault tolerance. HDFS uses a uniform

triplication policy (i.e. three replicas for each data block) to improve data locality and ensure data availability and fault tolerance in the event of hardware failure [3]. The placement policy of replicas is critical to HDFS performance and reliability. For the common case, the triplication policy in HDFS works well in term of high reliability and high performance. The HDFS Replica Placement Policy (RPP) is a rack-aware policy. The drawback of the policy is that it cannot evenly distribute replicas to cluster nodes. As a result, such data placement policy can noticeably reduce heterogeneous environment performance and may cause increasingly the overhead of transferring unprocessed data from slow nodes to fast nodes.

Data placement problem needs to be considered to obtain an optimal placement solution that balances workload in the cluster. Although HDFS provides a balancing utility to address the issue of unbalanced HDFS cluster, which does not consider on computing capacity of each node. This paper presents data placement policy based on storage utilization and computing capacity to distribute data as even as possible with keeping same rules of existing HDFS RPP. As a result, there is no need to run balancer tool and reduce overhead of data transmission between different computing nodes. The proposed policy assigns the data blocks to cluster nodes in accordance with their storage and computing capacity.

2. Background

HDFS [8] is designed to reliably store very large files across machines in a large cluster. It has two kinds of nodes, where a namenode serves as the master node and datanodes as slave nodes. The namenode[5] maintains the metadata of the file system, which stores the directory structure, file descriptions and a block map which identifies the location of each block replica in the cluster. Each datanode is responsible for storing the actual data blocks on each machine, and handling incoming read and write requests. Each datanode also periodically sends a heartbeat message to the namenode to report machine and block status. The namenode receives such messages because it is the sole decision maker of all replicas in the system. Each application creates a HDFS client to access the file system.

HDFS stores each file as a sequence of blocks. The blocks of a file are replicated for fault tolerance. The

default HDFS replica placement policy is rack-awareness. The purpose of this replica placement policy is to improve data reliability, availability, and network bandwidth utilization. For the common case, the replication factor is three by default, the data placement policy is to put one replica on one node in the local rack, another on a different node in the local rack, and the last on a different node in a different rack. Such default data placement strategy assumes that the computing capacity and storage capacity of each node in the cluster is the same [1]. In a heterogeneous environment, the difference in nodes computing capacity may cause load imbalance and overhead of data transmission. The reason is that different computing capacities between nodes cause different task execution time, so the faster nodes finish processing local data blocks earlier than slower nodes do. At this point, the master assigns non-performed tasks to the idle faster nodes, but these nodes do not own the data needed for processing. The required data should be transferred from slow nodes to idle faster nodes through the network. Because of waiting for the data transmission time, the task execution time increases. It causes the entire job execution time to become extended. Moving large number of data affects Hadoop performance.

The rest of the paper is organized as follows. Section 3 describes related works. Section 4 introduces the HDFS RPP and presents the proposed policy in detail. Expected results are presented in Section 5, and we conclude in Section 6.

3. Related Works

Some data placement policies have been proposed in Hadoop framework for load balancing problem in recent years. In [3, 6], a new replica placement policy is proposed for HDFS, which addresses the load balancing issue by evenly distributing replicas to cluster nodes. The policies mainly focus on storage utilization of each node in homogeneous cluster environments where all cluster nodes have the same computing capabilities. Hadoop provides a mechanism to rebalance data manually [9]. Rebalancing is necessary in a Hadoop cluster, as it avoids under-utilization or over-utilization of data nodes. Data rebalancing is done by a rebalancing server. The drawback of this method is that it has to be invoked manually whenever a new data node is added to the Hadoop cluster or when the cluster is imbalanced. The researchers in [2] proposed a data placement algorithm to resolve the unbalanced node workload problem. The algorithm is based on different computing capacities of nodes to allocate data blocks, thereby improving data locality and reducing the additional overhead to enhance Hadoop performance. The computing capacity of each node is based on the average execution time of a single task in that node. Reference

[7] proposed a method that first computes the nodes computing capability based on the log information about the history tasks. Then data is divided into different sized blocks according to the nodes computing capacity. Further the dynamic data migration policy aims at the transfer of data from slow DataNode to headmost DataNode during execution time. The computing capability is defined as the time cost which is spent to execute a unit size of data.

4. Data/Replica Placement in HDFS

When a user submits a job to Hadoop for some required process, it needs to specify the location of input as well as output files in HDFS. As the client write file, this file is split into data blocks. To handle this process, a HDFS client first asks the namenode to update the metadata. The namenode responds with a write permission indicating the datanode on which the blocks of the file should be written. Number of datanodes depends on the replication factor of that file. By default replication factor is three in HDFS, therefore location of three datanodes are returned by namenode. Then the list of datanodes forms a pipeline. The client then writes the data block onto the first datanode in the pipeline and forwards it to the second datanode. Similarly, after the block is stored in the second datanode, forwards it to the third datanode. The data placement process is complete after all replicas of the blocks have been written as depicted in Figure 1.

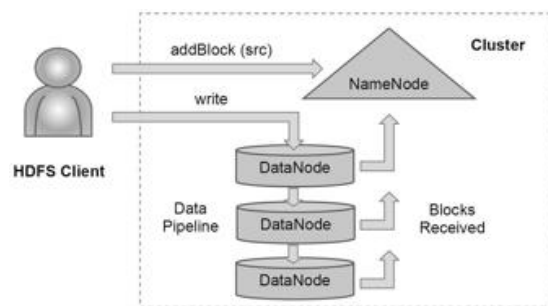


Figure 1. Data Placement in HDFS

4.1. Proposed Replica/Data Placement Policy

In a heterogeneous cluster, the storage and the computing capacity for each node are not the same. Therefore, the proposed data placement policy first calculates storage utilization, computing capacity of each node in the cluster and stores in nodeStatus table. When the Hadoop startup, the nodeStatus table is created in NameNode using algorithm 1 and updated everytime NameNode receives heartbeat message from DataNodes. The proposed system uses the number of finished tasks of each data node for a given time interval as computing capacity.

-To calculate storage utilization of the k^{th} DataNode $SD(k)$, uses formula (1)

$$SD(k) = \frac{SU(k)}{ST(k)} \% \quad (1)$$

($SU(k)$ denotes used space of the k^{th} DataNode and $ST(k)$ denotes total storage capacity of the k^{th} DataNode).

-To calculate average storage utilization of the j^{th} rack $SR(j)$, uses formula (2)

$$SR(j) = \frac{\sum_{k=0}^{DN-1} SD(k)}{DN} \quad (2)$$

(DN denotes the number of available data nodes of the j^{th} rack).

-To get the least storage utilization of the rack sui_Rack , uses formula (3)

$$\text{Min}(SR(j), SR(j+1), \dots, SR(r-1)) \quad (3)$$

(r denotes the number of racks in the cluster).

Table 1. Notations used in Proposed Data Placement Policy

Notation	Description
RF	Replication factor of the file. (maximum replication factor is 3 in this system.)
TargetNode	A array holds the target data nodes.
LocalNode	The DataNode where the client initiates a write
highestComputingNode	The DataNode has the maximum number of finished task for a given time interval in the rack.
nodeFlag	It is to be used to know the node is whether in local or remote rack.
nodeStatus Table	The table stores storage utilization, computing capacity of each node in the cluster
reqBlocks	The number of blocks for requested file to be distributed.
sui_Nodes	An array holds the datanodes which storage utilization are less than $SR(j)$
sui_Rack	the least storage utilization of the rack in the cluster

Algorithm 1: Making nodeStatus Table

Step 1: For each node do

Step 1.1: Calculate storage utilization $SD(k)$ using formula (1).

Step 1.2: Get the computing capacity of each node by retrieving the number of finished tasks at a given time interval.

Step 1.3: Fill storage utilization and computing capacity in the nodeStatus Table.
End for.

Algorithm 2: Proposed Data Placement Policy

Input: RF, reqBlocks

Output: targetNode[reqBlocks][RF]

Step 1: nodeFlag=0

Step 2: For each block do

Step 2.1: If targetNode.size==0 then

Step 2.1.1: Choose localNode.

Step 2.1.2: targetNode.add(localNode).
End If

Step 2.2: while targetNode.size<RF

Step 2.2.1: If nodeFlag==1 then

Find $SR(j)$ for the remote rack using formula (2).

//to calculate $SR(j)$, get storage utilization of each datanode from nodeStatus Table.

For each DataNode do

If $SD(k) < SR(j)$ then

sui_Nodes[] = DN(k)

End If

End For

Choose the highestComputingNode from sui_Nodes[].

Step

2.2.1.2:

//get computing capacity of each datanode from nodeStatus Table.

End If

Else

Find $SR(j)$ for the local rack using formula (2).

Step 2.2.2:

Step

2.2.2.1:

//to calculate $SR(j)$, get storage utilization of each datanode from nodeStatus Table.

For each DataNode do

If $SD(k) < SR(j)$ then

sui_Nodes[] = DN(k)

End If

Step

2.2.2.2:

Choose the highestComputingNode from sui_Nodes[].

//get workload of each datanode from nodeStatus Table.

If nodeFlag==0 then Set nodeFlag to 1.

End If

Step

2.2.2.3:

End If

targetNode.add(highestComputingNode).

End while

End For

Step

2.2.2.4:

Step 2.2.3:

Step 3:

return targetNode.

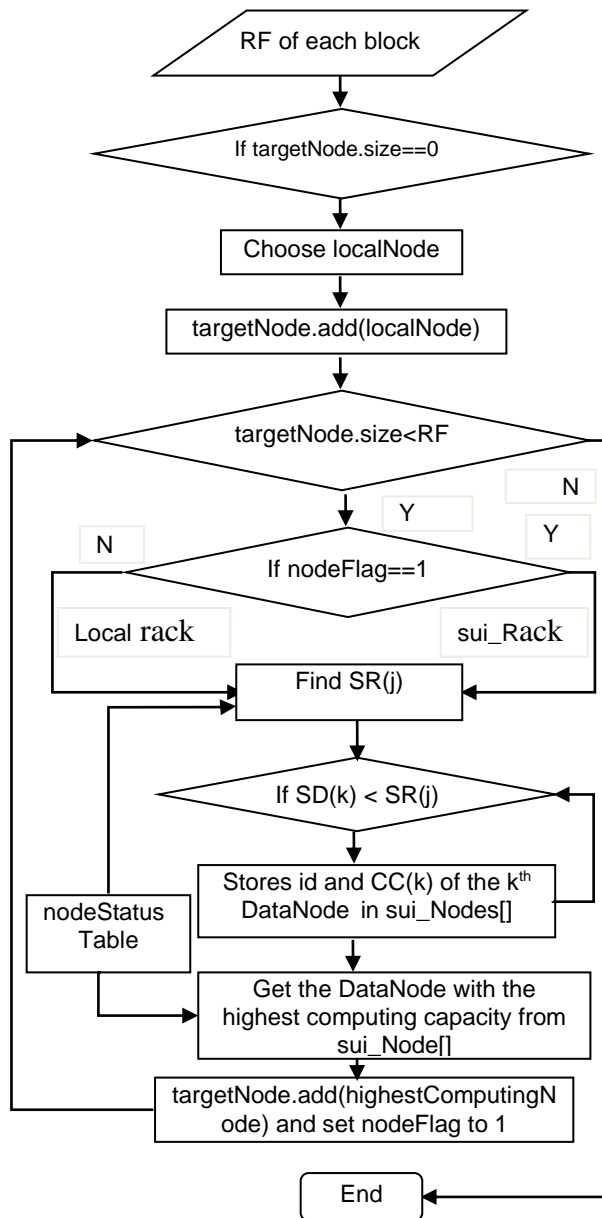


Figure 2. Flow Chat of Proposed Data Placement Policy

6. Conclusion

The default data placement strategy assumes that the computing capacity and storage capacity of each node in the cluster is the same. The HDFS Replica Placement Policy cannot evenly distribute replicas to cluster nodes. As a result, such data placement policy can noticeably reduce heterogeneous environment performance and may cause increasingly the overhead of transferring unprocessed data from slow nodes to fast nodes. Therefore, a data placement strategy based on storage utilization computing capacity of each node is proposed.

By considering the storage utilization of each data node for data placement, it can reduce load unbalance problem. Moreover, it can reduce the overhead of data transmission from the slow node to the fast node during execution time by placing data based on computing capacity. As a result, the proposed paper can improve the performance of Hadoop.

7. References

- [1] Avishan Sharafi, Ali Rezaee, "Adaptive Dynamic Data Placement Algorithm for Hadoop Heterogenous Environment", JACET Journal of Advance in Computer Engineering and Technology,2(4) 2016.
- [2] C.-W. Lee et al., A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments, Big Data Research(2014), <http://dx.doi.org/10.1016/j.bdr.2014.07.002>
- [3] Ibrahim Adel Ibrahim*, Wei Dai *, Mostafa Bassiouni, "Intelligent Data Placement Mechanism for Replicas Distribution in Cloud Storage Systems", IEEE International Conference on Smart Cloud, 2016.
- [4] J. Dean, S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", in Proceedings of Sixth Symposium on Operating System Design and Implementation (OSDI'04), San Francisco, CA, December 2004.
- [5] Q. Zhang, Sai Qian Zhang*, Alberto Leon-Garcia*, Raouf Boutaba , "Aurora: Adaptive Block Replication in Distributed File Systems", IEEE 35th International Conference on Distributed Computing Systems, 2015.
- [6] Wei Dai *, Ibrahim Ibrahim*, Mostafa Bassiouni, "A New Replica Placement Policy for Hadoop Distributed File System, IEEE 2nd International Conference on Big Data Security on Cloud, IEEE International Conference on High Performance and Smart Computing, 2016, IEEE International Conference on Intelligent Data and Security
- [7] Y. Fan, W. Wu, H. Cao, H. Zhu, X. Zhao, W. Wei, "A heterogeneity-aware data distribution and rebalance method in Hadoop cluster", 2012 Seventh ChinaGrid Annual Conference.
- [8] https://hadoop.apache.org/docs/r2.8.0/hadoop-projectdist/hadoop-hdfs/HdfsDesign.html#Data_Replication
- [9] Hadoop Data Balancer Administrator Guide, (01-20 2014),<https://issues.apache.org/jira/secure/attachment/12369201/BalancerAdminGuide.pdf>

Parallel PAM Clustering Algorithm for Learning Analytics

Nway Yu Aung, Swe Zin Hlaing

University of Information Technology, Yangon, Myanmar

nwayuaung@uit.edu.mm, swezin@uit.edu.mm

Abstract

Learning Analytics (LA) is defined as an area of research and application and is related to academic analytics, action analytics, and predictive analytics. This paper focuses the handling huge amount of data for better analysis. The challenges facing LA are regarding the need to increase the scope of data capture so that the complexity of the learning process can be more accurately reflected in analysis. This paper focuses on handling huge amount of data for better analysis. Partition Around Medoids (PAM) algorithm is one of the partition clustering algorithms. It tackles the problem in an iterative. However, it is not widely used for large data because of its high computational complexity. Parallelization technique can solve this problem. So, this paper proposed parallel PAM algorithm which is implemented by using Spark framework. This paper showed that the partition algorithm on Spark is slightly better than execution time tradition PAM algorithm.

Keywords- Clustering, PAM, Spark

1. Introduction

Nowadays, Information and technology are more developed. Data are generated from various sources such as social media, Internet of Things, multimedia, sensor networks etc... This huge amount of data can be used in many fields. For example, health care, bioinformatics, public administration, educations and many more. But, handling tremendous amount of data is difficult in term of storage, processing, retrieval. Application of some preprocessing techniques can make the data comprehensible to form the data analytics. Clustering, an unsupervised data mining technique which can effectively applied in such situations. The clustering was a process that divided the abstract object into same objects classes. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in other clusters. Clustering is used to find the shape of data set, and also in detection of anomalies. Nowadays, the clustering algorithms include partitioning methods, hierarchical methods, density-based method, model-based method and grid-based methods. Medoids algorithms are stronger than means algorithms because

medoids are less affected by noisy values. Partition Around Medoids (PAM) algorithm is one of the medoids algorithms, which attempts to determine k partitions for n objects, after an initial selection of k representatives, the algorithm repeatedly tried to make a better choice of cluster representatives and analyze all of the possible pairs of objects. But, partition around medoids algorithm has several drawbacks. This algorithm works successfully for small data sets but it does not work well for huge data sets. So, this paper intends to parallel partition Around Medoids algorithm which is proposed to handle to big data.

The data size is increasing at an exponential rate which makes it difficult to be handled on single machine environment. So, the existing algorithm need to optimize to run in distributed environment. Apache Hadoop emerged as one of the most powerful, scalable, fault tolerant platforms for this purpose. Hadoop provides two major abstractions for data storage and processing. Hadoop Distributed File Systems (HDFS) is used for distributed storage and MapReduce parallel programming is used for distributed processing. However, MapReduce programs are very much sensitive to iterations, as in each round the data is written back to the file system. Multiple read and writes to the file system increases to IO cost. PAM algorithm is iterative in nature. So, MapReduce based is quite costly.

Moreover, to get the optimal number of clusters, the algorithm needs to execute multiple times. Apache Spark, a very recently developed framework, is a better alternative. Spark does in-memory execution, which is faster in comparison to multiple read and write to the disk as in case of MapReduce. Hence, the execution time is optimized in Spark. It has been experimentally proved that spark works 100 times faster than Hadoop MapReduce when data is in memory; also the speed is 10 times when data is accessed from the disk. Spark can run on Hadoop, Mesos, standalone, or in the cloud. The proposed algorithm is implemented using Spark. The organization of the remaining paper is as follows. Section 2 reviews some of the related work. Section 3 discusses the proposed work. Experiment and results are in section 4. Conclusion of the paper is in Section 5.

2. Related Work

In the recent years, many clustering algorithms for big data have been proposed which are based on distributed and parallel computation. Cui, Xiaoli, et al. proposed optimized big data K-Means using MapReduce in which they claimed to counter the iteration dependence of MapReduce jobs [5].

Longhui Wang [1] has focused on Parallel algorithm based on Spark Cloud Computing Platform. They discussed MAX-MIN Ant System algorithm (MMAS) is parallelized to solve Traveling Salesman Problem (TSP) based on Spark cloud computing platform that combine MMAS with Spark MapReduce to execute the path building and the pheromone operation in a distributed computer cluster.

Jia LI [2] have proposed based on the bootstrap trails and implemented as an intelligent bootstrap library (IBL) on Spark to support efficient data clustering which obtain the trade-off between clustering efficiency and result quality.

Neha Bharill [3] has discussed the design of partitioned clustering algorithm and its implementation on Apache Spark. Partitioned based clustering algorithm called Scalable Random Sampling with Iterative Optimization Fuzzy c-Means algorithm (SRSO-FCM) which is implemented on Apache Spark to handle the challenges associated with Big Data Clustering.

Tapan Sharma [4] has simultaneously run multiple K-means algorithms with different initial centroids and values of k in the same iteration of MapReduce jobs. For initialization of initial centroids, Scalable K-means++ MR jobs have implemented and also run a validation algorithm of simplified Silhouette Index for multiple clustering outputs again in the same iteration of MR jobs. And then, the behavior of the above clustering algorithms which run on big data platforms like MapReduce and Spark jobs. Spark has been chosen as it is popular for fast processing particularly where iterations are involved.

Feng Bo [6] [7] proposed a new improved partition around medoids algorithm. This algorithm builds minimum spanning tree and then splits it to get k initial clusters with the relevant cluster centers. Experimental results show that the finding initial centers are closed to the desired cluster centers, and the improved algorithm achieved the stable clustering results and higher clustering accuracy.

3. Proposed System

3.1. PAM Algorithm

Traditional Partition Around Medoids algorithm was proposed by Kaufman and Rousseeuw in 1987. The

PAM algorithm is based on the search for k representative objects or medoids among the observations of the data set. After finding a set of k medoids, clusters are constructed by assigning each observation to the nearest medoids. Next, each selected medoids m and each non-medoids data point are swapped and the objective function is computed. The objective function corresponds to the sum of the dissimilarities of all objects to their nearest medoids. The SWAP step attempts to improve the quality of the clustering by exchanging selected objects (medoids) and non-selected objects. If the objective function can be reduced by interchanging a selected object with an unselected object, then the swap is carried out. This is continued until the objective function can no longer be decreased. The goal is to find k representative objects which minimize the sum of the dissimilarities of the observations to their closest representative object.

Algorithm PAM

Input:

$O = \{O_1, O_2, \dots, O_n\}$

K=number of desired clusters

Output:

K=set of clusters

PAM algorithm:

Randomly select k medoids from O

Repeat

 For each O_j not a medoid do

 For each medoid O_i do

 Compute square error

function S_{ij}

Find i,j where S_{ij} is the smallest

If $S_{ij} < 0$ then

 Replace medoid O_i with O_j

Until $S_{ij} \geq 0$

For each $O_i \in D$ do

 Assign O_i to K_h where $\text{dis}(O_i, O_j)$ is the smallest over all medoids;

In the above algorithm, O means objects (1,2,...,n). S_{ij} mean square error function and $\text{dis}(O_i, O_j)$ is the distance between object i and object j.

3.2. Apache Spark

Apache Spark is a popular open-source platform for large-scale data processing that is well-suited for iterative machine learning tasks. It is a lightning-fast cluster computing technology, designed for fast computation. Spark stores dataset in memory, which makes it a 100x faster than Hadoop. Also, it is a framework well suited to machine learning algorithms.

Components of Spark:

Resilient Distributed Datasets and the Spark Core:

The Spark Core is the foundation and provides basic I/O functionalities, task dispatching and scheduling. RDDs are basically a collection of partitioned data. These are generally created by referencing datasets in storages such as Cassandra, HBase et al., or by applying transformations such as map, reduce, and filter etc. on existing RDDs.

Spark SQL:

Spark SQL, a component on the Core, introduces a new data abstraction called DataFrame, for providing support for structured data. It provides a language to manipulate DataFrames in Java, Python or Scala.

Spark Streaming:

Spark streaming rests on the Core as well and leverages on top of the Core which is proven to be ten times faster than Hadoop’s disk-based Apache Mahout due to the distributed memory-based Spark architecture. It implements common algorithms to simplify large scale machine learning pipelines, like logistic or linear regression, decision trees or k-means clustering.

MLlib Machine Learning Library:

This is a machine learning framework on top of the Core which is proven to be ten times faster than Hadoop’s disk-based Apache Mahout due to the distributed memory-based Spark architecture. It implements common algorithms to simplify large scale machine learning pipelines, like logistic or linear regression, decision trees or k-means clustering.

GraphX:

It is a graph-processing framework on the Core, and provides an API for graph computation that can model the Pregel abstraction, providing an optimized runtime.

3.3. Proposed Algorithm

One major drawback of traditional PAM is not suitable for huge amount of data. The proposed system resolved this problem by parallel technique. For parallelization, this system will work on Spark Computing platform. In this proposed system, PAM algorithm works parallelization by using Spark framework. PAM algorithm firstly selects one of the representative objects as medoids for every cluster and makes partitions of the other objects to the nearest cluster based on the distance with the chosen representative objects. Then it repeatedly tries to get a better choice of cluster representatives until the process comes to a convergence. PAM algorithm takes as input dissimilarity D and produces as output a set of cluster centers or medoids. These medoids identify the clusters. Initialize cluster medoids.

$$m_1, m_2, \dots, m_n$$

Let n be the number of clusters and m denote any size n collection of the element x_i . Compute the minimum distance between data point x_i and medoids m_j .

$$D(x_i, m_j) = \min_{j=1,2,\dots,n} D(x_i, m_j) \tag{1}$$

Where $i \in j$ Re-compute medoids m_j

$$x_j \in j, m'_j = x_i \text{ and } x_i = m'_j \tag{2}$$

$$m_j = \min(\sum_{x_i \in j} D(x_i, m_j)) \tag{3}$$

The algorithm repeats works until medoids do not change. This system used learning analytics data sets to check the quality of spark based clustering algorithm.

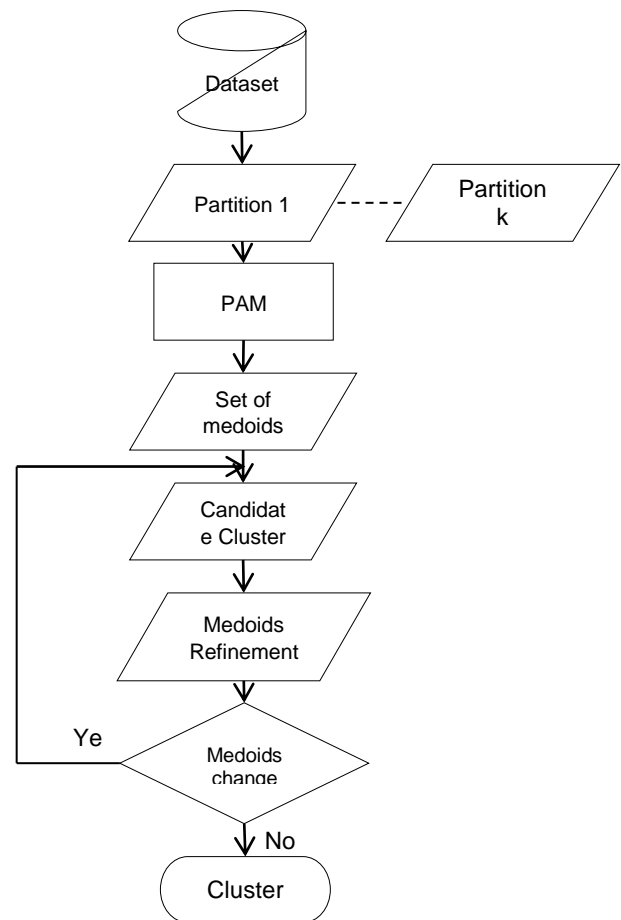


Figure 1. System flow diagram

4. Experimental Result

The proposed algorithm was simulated on UBUNTU 16.04 operating system. The Hadoop Version 2.7.0 was installed. Spark 2.0.0 run on top of Hadoop. Apache Hadoop and Spark are open source software. Figure 3 show the execution time of three algorithms: traditional PAM, PAM-Hadoop and PAM-spark.

In PAM-Hadoop, HDFS is stored the sample file and the initial center. After reading the initial cluster center, the other samples were divided to the most similar cluster parallel in Mapper function. In mapper function, (key, value) output pair is the cluster and the sample file. And then, clustering is again in Reducer function to identify the final center. In reducer function, (key, value) output pair is the cluster and the new center. The iteration is running that the result is compared with the new center which is not changed.

In PAM-spark, initial data are divided into partitions and distributed among various nodes in a cluster of computer. The data chunks are stored in Hadoop Distributed File System (HDFS). HDFS keeps 3 replicas (default) of each data chunk which minimizes the chances of data loss due to node failure. The replication factor can be increased or decreased according to requirement of the programmer. For each block in HDFS, the abstraction provided by spark, the Resilient Distributed Datasets (RDD) contains one partition. The original data is partitioned and distributed among the various nodes in the cluster, and then picked and worked upon in parallel by Spark. This system used the algebra dataset. These dataset contains 19 attributes and referenced from the website (<https://pslcdatashop.web.cmu.edu>). The data sets cannot actually be considered big data. However, they are large enough to judge the efficiency.

In this comparison, PAM-spark and PAM-Hadoop are faster than traditional PAM algorithm. Moreover, PAM-spark was much faster than PAM-Hadoop because of the advantages of Spark over Hadoop.

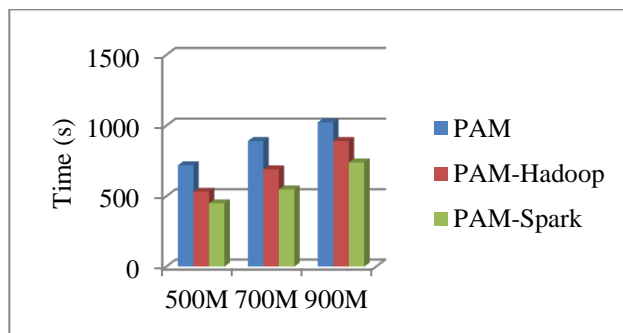


Figure 2. Time comparison between three algorithms

5. Conclusion

This paper presented partition around medoids algorithm implemented on Spark. Experimental results proved that the proposed algorithm outperforms the traditional partition algorithm implemented in Spark. Moreover the algorithm scales gracefully on increasing the data size and adds more machines to cluster. But, the proposed algorithm mainly focuses on time comparison. The proposed system should not consider the cluster quality. Initial cluster and medoids are main factor of considering cluster quality. In this algorithm, initial medoids choose randomly. The future work extends this algorithm for choosing initial medoids by using Bat algorithm. Moreover, the quality of clusters will be considered in another work. Also, the system will consider real world data analysis to improve the quality of teaching and learning.

6. References

- [1].Longhui Wang, Yong Wang, and Yudong Xie, "Implementation of a Parallel Algorithm Based on a Spark Cloud Computing Platform", *Directory of open access Journal volume 8, issue 3*, Switzerland, 2015.
- [2] Jia LI, Dongsheng LI, and Yiming ZHANG, "Efficient Distributed Data Clustering on Spark", *IEEE International Conference Cluster Computing*, Chicago, IL, USA, 2015.
- [3] Neha Bharill, Aruna Tiwari, and Aayushi Malviya, "Fuzzy Based Clustering Algorithms to Handle Big Data with Implementation on Apache Spark", *IEEE Second International Conference on Big Data Computing Service and Applications*, Oxford, UK, 2016.
- [4] Tapan Sharma, Dr. Sunil Mathur, and Dr. Vinod Shokeen, "Multiple K Means++ Clustering of Stallite Image Using Hadoop MapReduce and Spark", *International Journal of Advanced Studies in Computer Science and Engineering, IJASCSE volume 5 issue 4*, 2016.
- [5] Cui, Xiaoli and Zhu, Pingfei and Yang, Xin and Li, Keqiu and Ji, Changqing, "Optimized big data k-means clustering using MapReduce", *The Journal of Supercomputing Volume 70, Issue 3*, 2014.
- [6] Feng Bo, Hao Wenning, Chen Gang, Jin Dawei, Zhao Shuining "An Improved PAM algorithm for optimizing Initial Custer Center", *IEEE 3rd*

International Conference on Software Engineering and Service Science (ICSESS), Beijing, China, 2012.

[7] Mark Van der Laan, Katherine Pollard, Jennifer Bryan, "A new Partitioning around medoids algorithm". *Journal of Statistical Computation and Simulation*, 2013.

[8] Isaac B.Muck, Vasil Hnatyshin, Umashanger Thayasivam, "Accuracy of class prediction using Similarity functions in PAM". *IEEE International Conference on Industrial Technology (ICIT)*, Taipei Taiwan, 2016.

[9] Abhishek Bhattacharya, Shefali Bhatnagar, "Big data and Apache Spark: A Review", *International Journal of Engineering Research and Science (IJOER)*, 2016.

[10] Jiawei Han, Micheline Kamber, Jian Pei, *Data Mining, Concepts and Techniques*, Third Edition, 2011.

[11] <https://spark.apache.org>.

An Integrative Access Control with an Attributes-based Event Handler for Data Protection in Cloud Storage

Phyo Wah Wah Myint, Swe Zin Hlaing
University of Information Technology, Yangon, Myanmar
phyowahwah@uit.edu.mm, swezin@uit.edu.mm

Abstract

For many enterprises, cost savings and reducing the security risk are very important for data exchanging between all the involved parties. Cloud computing allows users to use different services which save storage and maintenance costs. The personal data has to be protected against unauthorized accesses in cloud storage. This paper shows an integrative access control with an attributes-based event handler for data protection. This access control is also an improvement of ciphertext policy attribute-based encryption (CP-ABE). In this access control scheme, a data owner firstly chooses an attribute-based access policy before he encrypts his data, and then encrypts the data. Moreover, an attributes-based event handler is integrated in this scheme before decryption phase. Every authorized/unauthorized user has to be checked by this handler. If and only if a user can pass this handler, he/she can decrypt the data. This handler checks every user by four cases. All of the four cases in this handler depend on both attributes-based policy and session timer of user during his/her requests for ciphertext. This integrative access control intends to give a full right on data owner to define access policy and to get a fine-grained access control for data protection in cloud storage.

Keywords- cloud storage, attribute-based access control (ABAC), attribute-based encryption (ABE), ciphertext policy ABE (CP-ABE)

1. Introduction

Modern societies and organizations are more and more complex, dynamic and flexible. The management of private and confidential information is a major problem for dynamic organizations. Data owners and organizations are motivated to outsource more and more sensitive information into the cloud servers. Protecting cloud from unauthorized users and other threats is a very important task for security providers who are in charge of the cloud. The secure cloud is always a reliable source of information. Access control is one of the most fundamental requirements in cloud computing. Access control has the paramount responsibility of providing smooth and easy access to authorized users as well as, at

the same time, preventing access to any unauthorized users. It is also a mechanism by which services know whether to honor or deny requests. There are some existing systems on access control in cloud which are centralized in nature. For securing data storage in cloud, it needs to use decentralized access control scheme that supports user authentication, key generation and management as well as multi authority data storage and retrieval. This paper focuses on the integration of access control and an attributes-based event handler for data protection in cloud storage. Moreover, it refers to the improvement of ABE schemes such as Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE) for cloud computing. Section 2 describes the related work including the ABE, literature review and their problem statements. Section 3 describes proposed integrative access control with an attribute-based event handler for data protection in cloud storage. Section 4 describes analysis on access control techniques over CP-ABE and proposed system. Section 5 includes conclusion. Section 6 describes limitation and further extensions and finally describes the references.

2. Related Work

2.1. Attribute-based Encryption (ABE)

Attribute-based encryption (ABE) is a public-key based one-to-many encryption that allows users to encrypt and decrypt data based on user attributes [1] in which the secret key of a user and the ciphertext are dependent upon attributes. The decryption of a ciphertext is possible only if the set of attributes of the user key matches the attributes of the ciphertext. Decryption is only possible when the number of matching is at least a threshold value. Collusion-resistance is crucial security feature of Attribute-Based Encryption. An adversary that holds multiple keys should only be able to access data if at least one individual key grants access. Another modified form of classical model of ABE is Key-Policy ABE (KP-ABE) as in Figure.1. The KP-ABE scheme can achieve fine-grained access control and more flexibility to control users than ABE scheme. Another modified form of ABE is called Ciphertext Policy ABE (CP-ABE) as in Figure. 2. CP-ABE improves the disadvantage of KP-ABE that

the encrypted data cannot choose the decryptor who can decrypt it. It can support the access control in the real environment. [1] [3].

2.2. Literature Review of Ciphertext Policy Attribute-based Encryption

Researchers have described the problems occurred in ABE schemes in various ways. Shucheng Yu, Cong Wang, KuiRen, and Wenjing Lou proposed a combining technique of attribute-based encryption (ABE), proxy re-encryption, and lazy re-encryption technique to achieve a fine-grained data access control in cloud computing [5]. Tengfei Li, Liang Hu, Yan Li, Jianfeng Chu, Hongtu Li, and Hongying Han studied ABE schemes of data access control in cloud storage environment. They listed some unsolved issues of existing access control schemes for cloud storage to provide some future developed direction about the further improvement [2]. Pradnya P. Shelar and Prof. Manisha M. Naoghare surveyed on efficient CP-ABE and secure data access control for multi authority cloud storage with data mirroring. They proposed a revocable multi-authority CP-ABE scheme to design the data access control scheme [4]. John Bethencourt, Amit Sahai, Brent Waters proposed ciphertext policy attribute based encryption (CPABE) by additional consideration for a delegation on an essential attribute structure [6]. Luan Ibraimi, Muhammad Asim, Milan Petkovic, Brent

Waters proposed An Encryption Scheme For A Secure Policy Updating [7]. But, it could not be efficient to be enough secure. G. Wungpornpaiboon1 and S. Vasupongayya proposed Two-layer Ciphertext-Policy Attribute-Based Proxy Re-encryption for Supporting PHR Delegation [8]. In [8], the encryption layer is divided into two layers such as inner and outer layer. The inner layer is possessed by data owner and outer layer is to satisfy the access structure for delegator. Jianwei Chen and Huadong Ma proposed Efficient Decentralized Attribute-based Access Control for Cloud Storage with User Revocation[9]. The authors proposed to consider the user revocation associated attributes set [9]. Le Qun Mo and Fu Yong Lin proposed a dynamic re-encrypted ciphertext-policy attributed-based encryption scheme for cloud storage [10]. The authors proposed to consider for re-encryption the ciphertext by using re-key in case of attribute revocation or delegation by delegator. Jiguo Li, Wei Yao, Yichen Zhang, Huiling Qian and Jinguang Han proposed Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing [11]. In [11], the authors proposed a fine-grained access control (ABE) scheme with efficient user revocation for cloud storage system. The issue of user revocation can be solved efficiently by introducing the concept of user group. When any user leaves, the group manager will update users' private keys except for those who have been revoked [11].

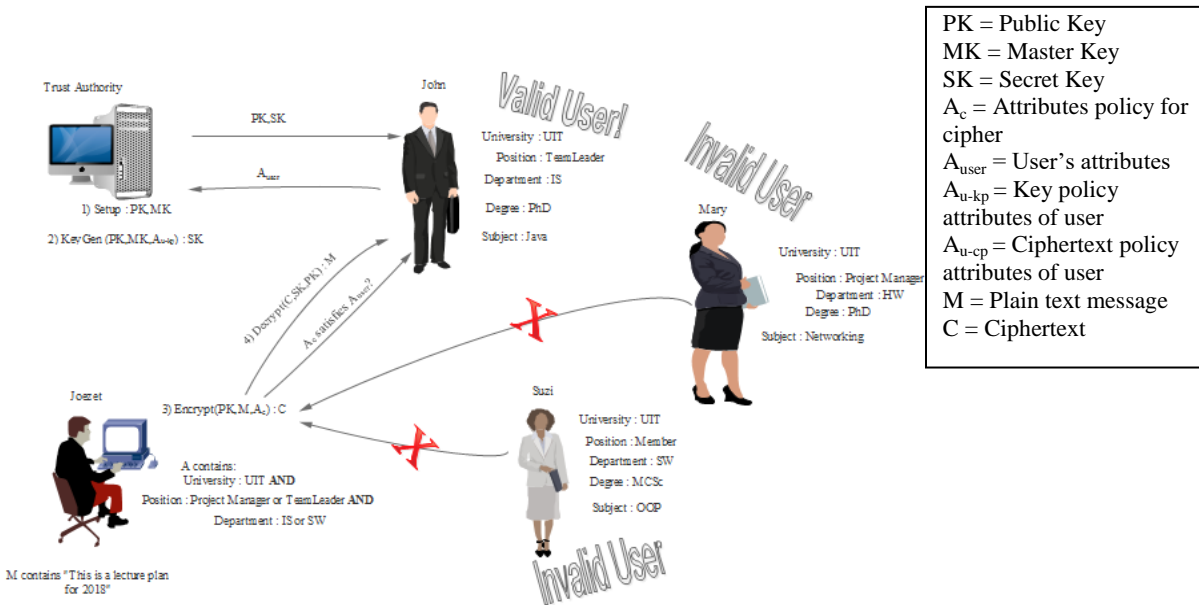


Figure 1. Key policy attribute-based encryption (KP-ABE) Illustration

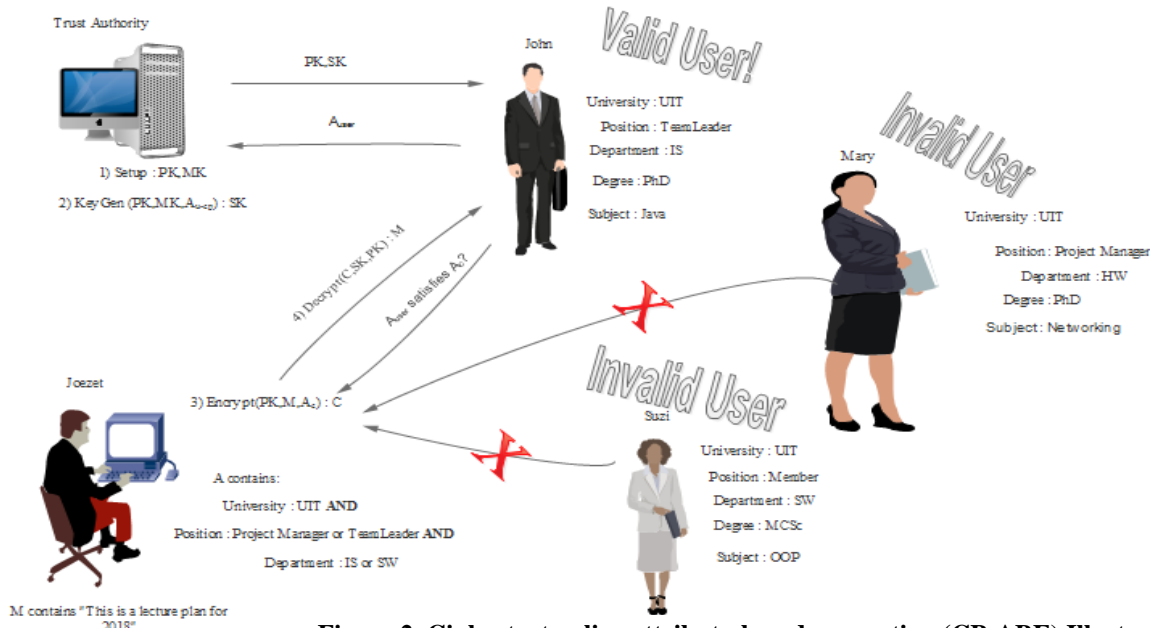


Figure 2. Ciphertext policy attribute-based encryption (CP-ABE) Illustration

2.3. Problem Statement

The problem with attribute based encryption (ABE) scheme is that data owner needs to use every authorized user's public key to encrypt data. The application of this scheme is restricted in the real environment because it uses the access of monotonic attributes to control user's access in the system [1]. The problem with KP-ABE scheme is that data owner cannot decide a user who can decrypt the encrypted data. It can only choose descriptive attributes for the data, it is unsuitable in some application because a data owner has to trust the key issuer [1] [3]. CP-ABE has still limitations in terms of specifying policies and managing user attributes [1] [2]. CP-ABE is one of the ongoing works in security research areas for data protection in cloud storage.

3. Proposed Integrative Access Control with an Attributes-based Event Handler for Data Protection in Cloud Storage

3.1. System Structure

In proposed system structure, there are four entities as follows [12]:

- Trusted Authority (TA): An entity which is trusted by all other participating entities in this system. It is trusted in the sense that it securely generates and stores master key and users' secret keys. It securely transmits those secret keys to the users upon valid requests.
- Data Owner (DO): The entity who owns data and encrypts those data.
- Data User (DU): The entity who would like to access encrypted data with proper authorization.
- Cloud Storage Provider: The entity that will provide storage service to store encrypted data.

3.2. System Design

In proposed integrative access control scheme, it emphasizes on defining access policies by the data owner's right. It assumes that a data owner can also update his access policy to grant/deny users who access the data. To check the access ability to the stored data, an attributes-based event handler is used to grant/deny the ciphertext. If user can pass this handler during his session for requesting the ciphertext, he can decrypt the ciphertext and can access data. So, any adversary has to challenge this access control with handler. This handler strongly filters any adversary. The detailed work of this handler will be explained in section 4.2. A workflow of this system design is shown in Figure.3 and Figure.4.

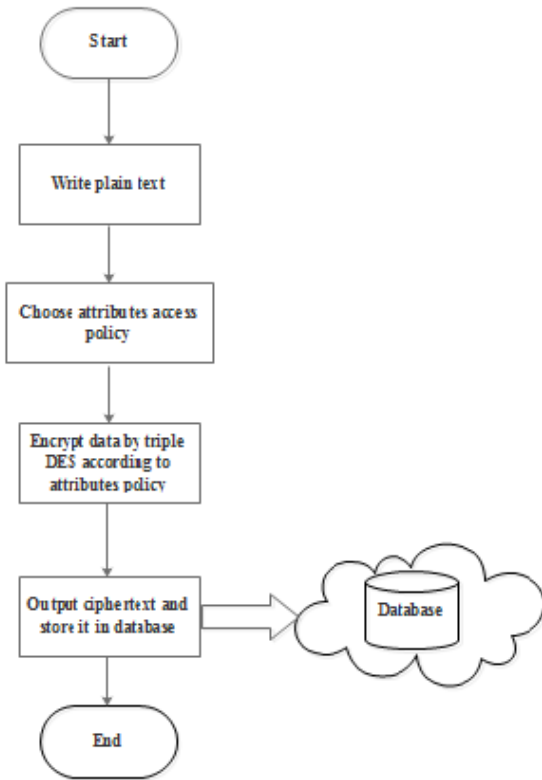


Figure 3. System flow for data owner site

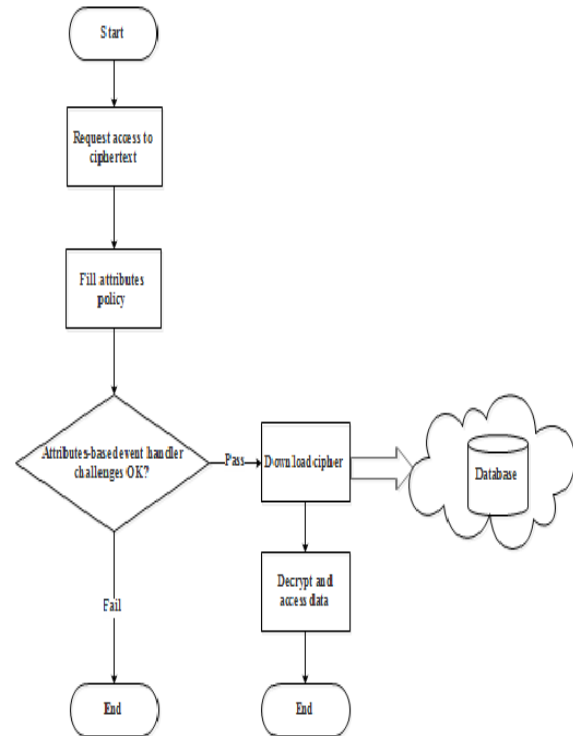


Figure 4. System flow for data user site

4. Analysis on access control techniques over CP-ABE and proposed system

4.1. Access Control by CPABE Technique

In the CP-ABE toolkit testing, it can only use for available attributes access structure in toolkit [6]. This tool is written by C language and tested on OpenSUSE Linux OS. In CP-ABE, there are four basic functions such as setup, key generation, encryption and decryption. The command lines are cpabe-setup, cpabe-keygen, cpabe-enc and cpabe-dec respectively.

The example of cpabe-setup command line is below.

- `srv1:~/Desktop/cpabe-0.11/test # cpabe-setup`

In setup phase, the system starts up and outputs public key and master key.

In the following keygen command lines, the attributes of user susu are 'research_field is Crypto, department is HW, qualification is MCTech, age is 28, office is 205 and position is tutor'. The attributes of user zarzar are 'research_field is DataMining, department is SW, qualification is MCSsc, age is 32, office is 105 and position is tutor' respectively.

- `srv1:~/Desktop/cpabe-0.11/test # cpabe-keygen -o susu_priv_key pub_key maser_key research_field Crypto department HW`

qualification MCTech 'age = 28' 'office = 205' position tutor

- `srv1:~/Desktop/cpabe-0.11/test # cpabe-keygen -o zarzar_priv_key pub_key maser_key research_field DataMining department SW qualification MCSsc 'age = 32' 'office = 105' position tutor`

In key generation phase, a secret key is generated for each user according to attributes policy.

The example of cpabe-enc command line is below.

- `srv1:~/Desktop/cpabe-0.11/test # cpabe-enc pub_key filename.pdf 'department and 2 of (age=32, SW,IS)'`

In encryption phase, the plaintext file filename.pdf is encrypted according to attributes policy '(department is SW or IS) and age is 32' and then plaintext is overwritten as filename.pdf.cpabe file extension. It cannot be read without decryption.

The examples of cpabe-dec command lines are below.

- `srv1:~/Desktop/cpabe-0.11/test # cpabe-dec pub_key susu_priv_key filename.pdf.cpabe`
- `srv1:~/Desktop/cpabe-0.11/test # cpabe-dec pub_key zarzar_priv_key filename.pdf.cpabe`

In decryption phase, any authorized user can decrypt the ciphertext and read the plaintext by using a secret key according to attributes policy. In the above two cpabe-dec

command lines, susu is not an invalid user so she cannot decrypt the filename.pdf.cpabe file and zarzar is a valid user because of her owned attributes so she can decrypt the ciphertext and read the plaintext filename.pdf successfully. In this toolkit, CP-ABE has still limitation in updating policy cases and revoking attributes/users cases.

4.2. Proposed Integrative access control with an Attributes-based Event Handler

In proposed access control scheme, it is written by C# language. In this system design, there are two sites such as data owner and data user sites. For a data owner, he needs to choose attributes access policy as he likes to encrypt the plaintext message. He can also update the attributes access policy for the cipher. For data user site, he firstly requests the ciphertext to access data. So, he needs to give his credentials including attributes for authorization. Here, an attributes-based event handler checks any authorized/unauthorized user before decryption phase. In this handler, each attribute filling is limited by each timer (milliseconds) control and what result he types (right/wrong) control. There are four cases

to check the adversary. The first one case ‘Session Timeout & Policy Fail’ occurs where an adversary fails the session timer control for each attribute filling step and types the wrong result for attributes-based policy in ciphertext. The second one case ‘Session Timeout & Policy Pass’ occurs where an adversary fails the session timer control for each attribute filling step and types the right result for attributes-based policy in ciphertext. The third one case ‘Session Gain & Policy Fail’ occurs where an adversary gains the session timer control for each attribute filling step and types the wrong result for attributes-based policy in ciphertext. The last one case ‘Session Gain & Policy Pass’ occurs where an adversary gains the session timer control for each attribute filling step and types the right result for attributes-based policy in ciphertext. Among these four cases of handler, only the last case ‘Session Gain & Policy Pass’ can pass the handler and download the ciphertext and can decrypt it. If and only if the user who can pass the handler, he can access the plaintext message. The system architecture is shown in Figure 5.

In Figure 5, the notations are as follows:

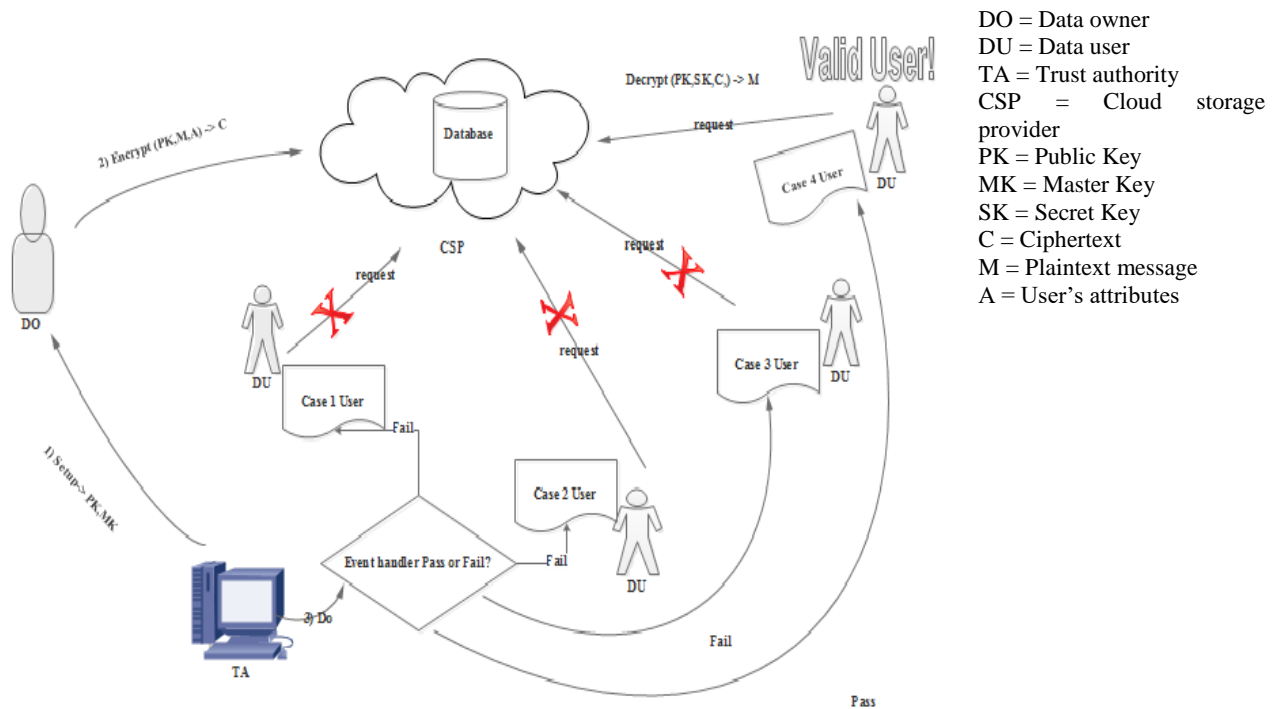


Figure 5. Proposed system architecture

4.3. Comparison between CP-ABE and proposed integrative access control with an attributes-based event handler scheme

In CP-ABE access control technique, the keys are generated by attributes access policy. It uses an advanced encryption standard (AES) for encryption. It is suitable for projects in real environment because data owner has a full access control. It has still limitations for attributes management and user revocation problems for multiple domain authority in cloud storage. It is still now in many research areas for data protection in cloud storage.

In proposed integrative access control scheme, the keys are also generated by attributes access policy. It uses triple data encryption standard (triple DES) for encryption. An attribute-based access control is used to define authorized/unauthorized user before both encryption and decryption phases as in the existing CP-ABE methodologies. An attributes-based event handler is also used to check any users. It is more secure and strong detection for user before decryption phase.

5. Conclusion

In this paper, the proposed integrative access control scheme uses an attribute-based access control with an event handler for detection of user credentials. It refers to help the CP-ABE over existing methodologies for data protection in cloud storage. It also focuses on the full control of data owner according to his/her access policy. An attribute-based event handler is also a good supporter for data confidentiality before decryption phase. This proposed integrative access control can give data confidentiality and availability.

6. Limitation and Further Extension

At this moment, only a text message is tested by proposed technique. It will be extended to test another messages (files). As a future work, it is going on to analyze the encryption techniques among CP-ABE researches for data protection in cloud storage.

7. References

[1] Minu George, Dr. C.Suresh Gnanadhas, Saranya.K, "A Survey on Attribute Based Encryption Scheme in Cloud Computing", International Journal of Advanced Research in Computer and Communication Vol. 2, Issue 11, November 2013.

[2] Tengfei Li, Liang Hu, Yan Li, Jianfeng Chu, Hongtu Li, and Hongying Han, "The Research and Prospect of Secure Data Access Control in Cloud Storage Environment", Journal of Communications Vol. 10, No. 10, October 2015.

[3] C. Vinoth, G.R.Anantha Raman, "A Survey on Attribute Based Encryption Techniques in Cloud Computing", International Journal of Engineering Sciences & Research Technology, January 2015.

[4] Pradnya P. Shelar, Prof. Manisha M. Naoghare, "A Survey on Efficient CP-ABE and Secure Data Access Control for Multi Authority Cloud Storage with Data Mirroring", International Journal of Innovative Research in Computer and Communication Engineering Vol. 3, Issue 10, October 2015.

[5] Shucheng Yu, Cong Wang, KuiRen, and Wenjing Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", in Proc. of INFOCOM'10, 2010.

[6] John Bethencourt, Amit Sahai, Brent Waters, "Ciphertext-Policy Attribute-Based Encryption", www.cs.utexas.edu/~bwaters/publications/papers.

[7] Luan Ibraimi, Muhammad Asim, Milan Petkovic, Brent Waters, "An Encryption Scheme For A Secure Policy Updating", in Proc. of the Security and Cryptography(SECRYPT) International Conference, 2010.

[8] G. Wungpornpaiboon, S. Vasupongayya, "Two-layer Ciphertext-Policy Attribute-Based Proxy Re-encryption for Supporting PHR Delegation", 978-1-4673-7825-3/15/\$31.00 ©2015 IEEE.

[9] Jianwei Chen and Huadong Ma, "Efficient Decentralized Attribute-based Access Control for Cloud Storage with User Revocation", IEEE ICC - Selected Areas in Communications Symposium, 2014.

[10] Le Qun Mo, FuYong Lin, "A dynamic re-encrypted ciphertext-policy attributed-based encryption scheme for cloud storage", IEEE Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, 2014.

[11] Jiguo Li, Wei Yao, Yichen Zhang, Huiling Qian and Jinguang Han, Member, IEEE, "Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing", DOI 10.1109/TSC.2016.2520932, IEEE Transactions on Services Computing.

[12] Phyto Wah Wah Myint, Swe Zin Hlaing, Ei Chaw Htoon, "An Encryption Access Control Scheme for Flexible Policy Updating in Cloud Storage", in Proc. of 14th International Conference on Computer Applications' Feb, 2017, pp-28-33.

Availability Modelling for SDN switch in Cloud based Infrastructure

May Thae Naing, Aye Myat Myat Paing
University of Information Technology, Yangon, Myanmar
maythae@uit.edu.mm, ayemyatmyatpaing@uit.edu.mm

Abstract

Attaining continuity and high availability of data transactions for cloud computing services are necessary for SDN architecture. The high-speed and complicated network of hosts and network devices often meet with a variety of failures due to links or system components. This failure affects the availability of the system. The proposed system uses a two-level availability model that is used to evaluate the availability of SDN concept in cloud based infrastructure. This paper offer availability solution for software defined network (SDN) in cloud computing Infrastructure and then describe the markov model for availability in SDN switches. Moreover, the impact of software and hardware failures on the overall availability of SDN switches is evaluated by SHARPE Tool.

Key Words- Cloud computing, Software Defined Networks, Availability

1. Introduction

Cloud computing has emerged as a widely accepted computing paradigm built around core concepts such as elimination of up-front investment, reduction of operational expenses, on-demand computing resources, elastic scaling, and establishing a pay-per-usage business model for information technology and computing services. There are different models of cloud computing that are offered today as services like Software as a Service (SaaS), Platform as a Service (PaaS), Network as a Service (NaaS) and Infrastructure as a Service (IaaS) [1]. In spite of all recent research and developments, cloud-computing technology is still evolving. Several remaining gaps and concerns are being addressed by alliances, industry, and standards bodies.

Software-Defined Networking (SDN) is an emerging networking paradigm that gives hope to change the limitations of current network infrastructures. First, it breaks the vertical integration by separating the network's control logic (the control plane) from the underlying routers and switches that forward the traffic (the data plane). Second, with the separation of the control and data planes, network switches become simple forwarding devices and the control logic is implemented in a logically centralized controller (or network operating system), simplifying policy

enforcement and network (re)configuration and evolution [2], [3].

The key concept of the cloud computing is virtualization. Virtualization is the abstraction of the physical resources needed to complete a request and underlying hardware used to provide service. And also, the idea of the SDN is adopted from the concept of virtualization, where controls and managements of software subsystems are completely decoupled from hardware infrastructure. The decoupled components of the SDN are separated into three layers of the SDN architecture; (i) Data plane: SDN enabled network devices on a data plane reside at the bottom of the SDN architecture as the underlying physical layer, (ii) Control plane: network operating systems and hypervisors on the control plane resides at the middle layer to provide a bare virtualized environment; and (iii) Management plane: network applications running on the management plane resides at the upper-most layer. This virtualization approach brings three key attributes to the SDN: logically-centralized intelligence, programmability and high-level abstraction. Nevertheless, there are still many issues to use SDNs [4]. In fact, physically centralized network infrastructure still requires adequate levels of system availability and reliability.

High availability refers to the ability of a system to perform its function continuously (without interruption) for a significantly longer period of time than the reliabilities of its individual components would suggest. High availability is mostly often achieved through fault tolerance. Therefore, the effort in the proposed system will offer availability model by a comprehensive evaluation of the SDN in cloud infrastructure. To evaluate the model using SHARPE tool simulation is presented.

This paper organizes as follows: Section II describes the related work of the proposed system, Section III presents the two-level availability model, and Section IV describes the case study for the model. Finally, Section V concludes the paper.

2. Related Work

One of the main reasons of hesitating to adopt SDNs is the concern on availability. There are a few works on the availability of SDNs. In paper [5], the authors considered the impact of SDN application failures on the

controller reliability. In paper [6], the authors proposed a stochastic model focusing only on the controller of a SDN rather than the whole SDN. In paper [7], the authors presented experimental results to improve the reliability and availability of core networks using SDN/Openflow. In paper [8], the authors proposed an approach to provide high-availability applications using a SDN. In this paper, hierarchical models will be presented for the availability of a SDN. In paper [9], the authors proposed a stochastic model focusing a stochastic availability model with the incorporation of hardware failures and software failures. They used RAID1 architecture for storage system.

In paper [10], the authors formalized a two-level availability model that is able to capture the global network connectivity without neglecting the essential details. It has highlighted the considerable impact of operational and management (O&M) failures on the overall availability of SDN. Moreover, its results showed that the impact of software and hardware failures on the overall availability of SDN can be significantly reduced through proper over provisioning of the SDN controller(s). The paper [11] provided similar availability to the traditional IP backbone networks. It also used a two-level availability model which is able to capture the global network connectivity without neglecting the essential details and which includes a failure correlation assessment should be considered. It also presented the implementation on M'obius of the Stochastic Activity Network (SAN) availability model of the network elements and the principal minimal-cut sets of a SDN backbone network and the corresponding traditional backbone network.

3. Two-Level Availability Model

A two-level hierarchical model [12] is introduced to evaluate the dependability of SDN in a global network. In this example, the dependability is measured in terms of steady state availability, in the following referred to as availability. The two-level hierarchical modelling approach consists of

- upper level: a structural model of the topology of network elements and controllers
- lower level: dynamic models (some) of network elements

The approach seeks to avoid the potential uncontrolled growth in model size, by compromising the need for modelling details and at the same time modelling a (very) large scale network. The detailed modelling is necessary to capture the dependencies that exist between network elements and to describe multiple failure modes that might be found in some of the network elements and in the controllers. The structural model disregards this and assumes independence

between the components considered, where a component can be either a single network elements with one failure mode or a set of elements that are interdependent and/or experience several failure modes and an advanced recovery strategy. For the former we need to use dynamic models such as a Markov model or Stochastic Petrinet (e.g., Stochastic Reward Network [13]), and for the latter structural models such as reliability block diagram, fault trees, or structure functions based on minimal cut or path sets.

The objective of the modeling approach is to evaluate the availability of SDN.

4. Model Case Study

In this evaluation, this paper consider the network topology depicted in Figure 1. The service provider access the SDN controller through the northbound APIs. The service provider uses them for configuring the network resources and adding or removing tenants. The tenants use the northbound APIs to create subnets, to define policies. The controller controls the network switches and gateways through OpenFlow. We distinguish between the core network and the edge. The latter comprises the access switches to which end hosts are connected and the gateways that connect the service provider's network to external networks.

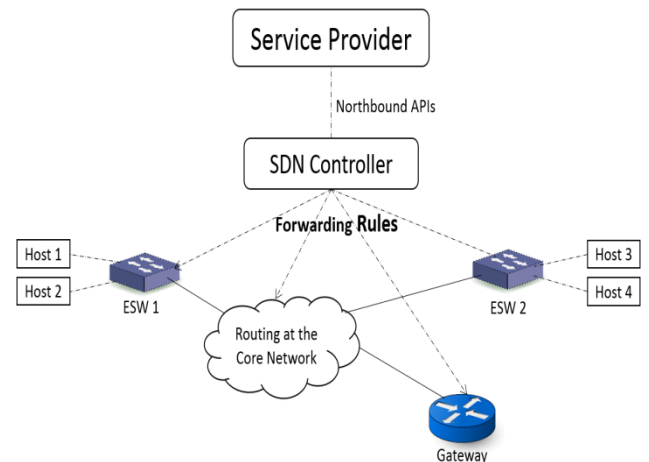


Figure 1. Topology for the model case study

5. Markov Model of the SDN switches

In the following, this paper present the markov model of the network element: SDN switches. Figure 2 shows the model of the switch in an SDN. The states related to the control hardware failures are not contained in this model, since all the control logic is located in the controller. In any case, we assume 1+1 redundancy of the SDN switch, which is a common best practice in any

architecture. Multiple failures are not included in the model.

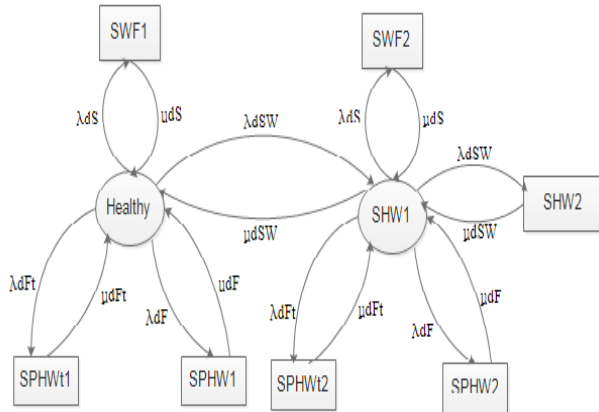


Figure 2. Markov model of SDN switches

Table 1. State Variables for SDN switch

State	Up/Down	Description
Healthy	up	System is fault free
SHW1	up	Hardware failure of one switch
SHW2	down	Hardware failure of both switches
FPHW1	up	Permanent hardware failure of one switch in forwarding plane
SPHW2	down	Permanent hardware failure of both switches in forwarding plane
FPHWt	down	Transient hardware failure of both switches in forwarding plane
SWF1	up	Transient hardware failure in forwarding plane
SWF2	down	Software failure of one switch
		Software failure of both switches

Table 2. Model parameters used in the case studies

Intensity	Time	Description
$1/\lambda dF = 6$	[months]	expected time to next permanent forwarding hardware failure
$1/\mu dF = 12$	[hours]	expected time to repair permanent forwarding hardware
$1/\lambda dFt = 1$	[week]	expected time to next transient forwarding hardware failure
$1/\mu dFt = 3$	[minutes]	expected time to repair transient forwarding hardware
$1/\lambda dSW = 6$	[months]	expected time to next control hardware failure
$1/\mu dSW = 12$	[hours]	expected time to repair control hardware
$1/\lambda dS = 1$	[week]	expected time to next software failure
$1/\mu dS = 3$	[minutes]	expected time to software repair

Table 3. Model parameters for the SDN switch

Intensity	Description
$\lambda F = \lambda dF$	intensity of permanent hardware failures
$\mu F = \mu dF$	repair intensity of permanent hardware failures
$\lambda Ft = \lambda dFt$	intensity of transient hardware failures
$\mu Ft = \mu dFt$	restoration intensity after transient hardware failures
$\lambda sS = 0$	intensity of software failure

All the model parameters are defined in Table II. In an SDN switch, the failure/repair intensities of (permanent/transient) hardware failures are the same because failures with the same cause and the same intensities. However, we assume that the software on an SDN switch will be much less complicated than on a traditional IP router because the control logic has been moved to the controllers.



Figure 3. Evaluating result for the model case study

6. Conclusions

This paper offer availability solution for software defined network (SDN) in cloud computing Infrastructure. A two-level availability model that includes structural and dynamic models has been formalized and for the dynamic level Markov model of the single network elements have been proposed. The numerical analysis used to evaluate the availability model. Finally, this proposed system will evaluate through both analytical and simulation tool (SHARPE).

7. References

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," September 2011.
- [2] N. Mckeown, "How SDN will Shape Networking," October 2011. [Online]. Available: <http://www.youtube.com/watch?v=c9-K5OqYgA>.
- [3] S. Schenker, "The Future of Networking, and the Past of Protocols," October 2011. [Online]. Available: <http://www.youtube.com/watch?v=YHeyuD89n1Y>.
- [4] D. Kreutz, F. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

- [5] B. Chandrasekaran and T. Benson, "Tolerating SDN Application Failures with LegoSDN," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, ser. HotSDN '14. New York, NY, USA: ACM, 2014, pp. 235–236. [Online].

- [6] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of Software Defined Networking," *Computer Networks*, Apr. 2015.

- [7] M. Tamura, T. Nakamura, T. Yamazaki, and Y. Moritani, "A study to Achieve High Reliability and Availability on Core Networks with Network Virtualization," *Technical Report*, vol. 15, no. 1, Jul. 2013.

- [8] S. Dwarakanathan, L. Bass, and L. Zhu, "Application Level HA and QoS Using SDN," *NICTA Technical Report*, Tech. Rep., 2015.

- [9] K. Han, T. A. Nguyen, D. Min, and E. M. Choi, "An Evaluation of Availability, Reliability and Power Consumption for a SDN Infrastructure Using Stochastic Reward Net," *Advances in Computer Science and Ubiquitous Computing*, vol. 421, 2016, pp 637-648.

- [10] G. Nencioni, B. E. Helvik, A. J. Gonzalez, P. E. Heegaard, and A. Kamiński, "Availability Modelling of Software-Defined Backbone Networks," submitted to the 2nd Workshop on Dependability Issues on SDN and NFV (DISN 2016).

- [11] G. Nencioni, B. E. Helvik, P. E. Heegaard, "Implementing the Availability Model of a Software-Defined Backbone Network in M²obius," submitted to the 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), 2017.

- [12] P. E. Heegaard, B. E. Helvik, G. Nencioni, and J. Wafner, "Managed dependability in interacting systems," in *Principles of Performance and Reliability Modeling and Evaluation*, L. Fiondella and A. Puliafito, Eds. Springer, 2016.

- [13] G. Ciardo and K. S. Trivedi, "A decomposition approach for stochastic reward net models," *Perf. Eval*, vol. 18, pp. 37–59, 1993.

Computation Offloading Decision in Mobile Cloud Computing: Enhance Battery Life of Mobile Device

Mi Swe Zar Thu, Hsu Mon Kyi

University of Information Technology, Yangon, Myanmar

swezar@uit.edu.mm, hsumonkyi@uit.edu.mm

Abstract

Functionality on mobile device is ever richer in daily life. Mobile devices have limited resources like battery life, storage and processor, etc. Nowadays, Mobile Cloud Computing (MCC) bridges the gap between the limited capabilities of mobile devices and the increasing user demand of mobile applications by offloading the computational workloads from local devices to the remote cloud. Deciding to offload some computing tasks or not is a way to solve the limitations of battery life and computing capability of mobile devices. Application offloading is energy efficient only under various conditions for determining where/which code should be executed. This paper presents a Computational Offloading Decision Algorithm (CODA), to save the battery life of mobile devices, taking into account the CPU load, state of charge, network bandwidth and transmission data size. The system can take decision which method should be offloaded or not based on different context of the mobile device to obtain minimum processing cost. Numerical study is carried out to evaluate the performance of system. Experimental result will demonstrate that the proposed algorithm can significantly reduce energy consumption of mobile device as well as execution time of application.

Keywords- Mobile Cloud Computing; Computation Offloading; Wireless Network Bandwidth; Energy.

1. Introduction

In our daily life, mobile devices have become common entity. However, the battery lifetime is still a major concern of the modern mobile devices. From the users' perspective, they need better performance of their mobile devices, which reflects on longer battery life and shorter processing time of any kind of services. These mobile devices provide us with much exciting applications which require large computing power, memory, network bandwidth and energy to run applications as multimedia, GPS navigation, real time games etc. which also adds energy consumptions constantly. Energy or Battery is the only resource in mobile devices that cannot be restored immediately and needs external resources to be renewed.

Computation offloading is a way to overcome this obstacle. Many works have been done in Mobile Cloud Computing (MCC), mainly focus on the code partitioning and offloading techniques, assuming a stable network connection and sufficient bandwidth. However, the context of a mobile device, e.g. network conditions and locations, changes continuously as it moves throughout the day. To tackle the issues mentioned above and improve the service performance in mobile cloud computing, we propose a Computation Offloading Decision Algorithm (CODA) that takes the advantages of both nearby cloudlet, local mobile device cloud, and public cloud computing services in the remote to provide an adaptive and seamless mobile code offloading service. The objective of the proposed system is to take offloading decision into the account of total execution cost of each method, device profiler and network profiler to provide better performance and less battery consumption.

2. Related Work

There have been many attempts to improve energy and CPU efficiency in mobile devices. These approaches enable to reduce application execution time on mobile devices and decrease the energy consumption of CPU. These attempts could be classified into two approaches: fine-grained and coarse grained tasks offloading schemes.

The first one relies on application developers to modify the code to handle partitioning, state migration, and adaption to various changes in network conditions. C.Eduardo [4] proposed Mobile Assistance Using Infrastructure (MAUI) profiler that measures the device characteristics at initialization time. It continuously monitors the program and network characteristics. Because these can often change and a state measurement may force MAUI to make the wrong decision.

The second approach assumes that the full process/program or full Virtual Machine (VM) is migrated to the remote servers. Then programmers do not have to modify the application source code to take advantage of computation offloading. B.G.Chun [2] presented a technique known as Clone Cloud to reduce the burden of Mobile Devices. Clone Cloud is a system

which automatically converts applications of the mobile devices by partially offloading it into the virtual clone (phones) present in the cloud. The author test Clone Cloud in HTC G1 device. As a result, Clone Cloud technique is helpful in reducing execution time and consumption of energy on mobile devices. A.Ellouze [1] presented Mobile Application Offloading (MAO) algorithm triggered by two conditions: the current CPU load and State of Charge (SoC) of the battery, assess its performance in terms of rejected jobs and the amount of energy savings achieved. To reduce application execution time on mobile devices, the round-trip time between the mobile terminal and the server is a key parameter that contains the level of interactivity of the applications that can be offloaded. The users and cloudlets may change their locations and become disconnected from each other. This will cause offloading failure. S M A.Karim [9] proposed an intelligent and dynamic algorithm to offload computation to the cloud focus on offloading computation based upon the communication topology, device energy and user inputs. That algorithm saves more time, compared to a previous approach, and also reduces device energy usage by moving energy hungry processes to the cloud. J.Oueis [7] discussed offloading algorithm which incorporates a multitude of parameters in the offloading decision process while reducing the mobile handset energy consumption and keeping a good user quality of experience. The purpose of this paper is to study how to deploy an offloadable application in a more optimal way, by dynamically and automatically determining which parts of the application tasks should be processed on the cloud server and which parts should be left on the mobile device to achieve a particular performance target (low energy consumption, low response time, etc.).

The remainder of this paper is organized as follows. Section 3 presents background theory of mobile cloud system. The overview of proposed system is presented in section 4. Then, proposed system architecture is explained in section 4.1 and followed by MAUI solving model is presented in section 4.2. This paper discusses Computational Offloading Decision Algorithm (CODA) in Section 4.3. After that, section 5 presents performance evaluation result. Finally, section 6 concludes the paper.

3. Background Theory

Computation offloading, as one of the main advantages of MCC, is a paradigm/solution to improve the capability of mobile services through migrating heavy computation tasks to powerful servers in clouds. Computation offloading yields saving energy for mobile devices when running intensive computational services,

which typically deplete a device's battery when are locally run.

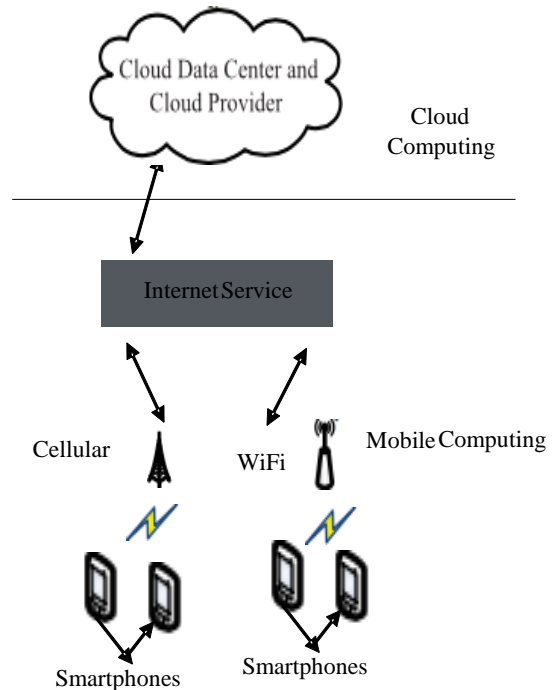


Figure 1. Mobile Cloud Computing

Nowadays, there has been a significant amount of research focusing on computation offloading because virtualization techniques enable cloud computing environments to remotely run services for mobile devices. These research themes are mostly related to explore ideas/ways to make computation offloading feasible, to draw optimal offloading decisions, and to develop offloading infrastructures. There are many factors that can adversely affect the efficiency of offloading techniques, especially bandwidth limitation between mobile devices and servers in cloud and the amounts of data that must be exchanged among them. Many algorithms have already been proposed to optimize offloading strategies to improve computational performance and/or save energy. These algorithms and techniques mostly analyze a few system parameters including network bandwidths, computation capability, available memory, server loads, and the amounts of exchanging data between mobile devices and cloud servers to propose offloading strategies. The real world performance benefits and battery gains are achieved by offloading certain amount of computational work from a mobile device (An android device in our case) to a cloud server. The system found a clear gain in terms of the load

on the CPU of the device as well as the battery life consumption.

The concept of Mobile Computational Offloading provides a solution for the execution of resource-hungry applications. Computation Offloading occurs at the code level in which an application is partitioned or analyzed before its development. While offloading computation to a cloud server, there are two important factors to keep in mind. The first factor is the size of the computation being performed and the second factor is the amount of data that needs to be sent and received for the computation to be successful.

Mobile Cloud Computing system consists of three parts, cloud, mobile clients, and wireless network. Cloud offers applications as services. Mobile clients access application services through proxies using wireless communication and proxies communicate with the cloud servers over fixed wired network.

4. System Overview

In this section, the components of proposed offloading system architecture are presented in detailed. This system is built on MAUI architecture to perform the offloading on method level. According to its three main components, the framework of proposed system is considered as in the following steps.

(1) **Offloading monitor:** is responsible for collecting real-time information on mobile devices, mobile networks and cloud servers.

(2) **Offloading planner:** is the decision making component of our framework. According to the collected information stored in the offloading monitor, it decides what services must be run locally and what services must be offloaded.

(3) **Offloading engine:** is responsible to execute mobile services based on decisions made by the offloading planner.

4.1. Proposed System Architecture

The proposed system is considered according to the conditions of the cloud and devices, such as CPU load, available memory, remaining battery power on devices, bandwidth between the cloud and devices.

The system decides the offloading method of application which is run on local device or remote cloud based on above conditions. The system components design and relations between each component are described below.

(1) **Device Profiler:** The device profiler includes factors like energy consumption and other processing factors related to mobile devices. The profiler gathers the hardware information of device and passes it onto the MAUI solver for prediction. The factor includes 1)

the average CPU usage 2) memory usage and 3) battery level.

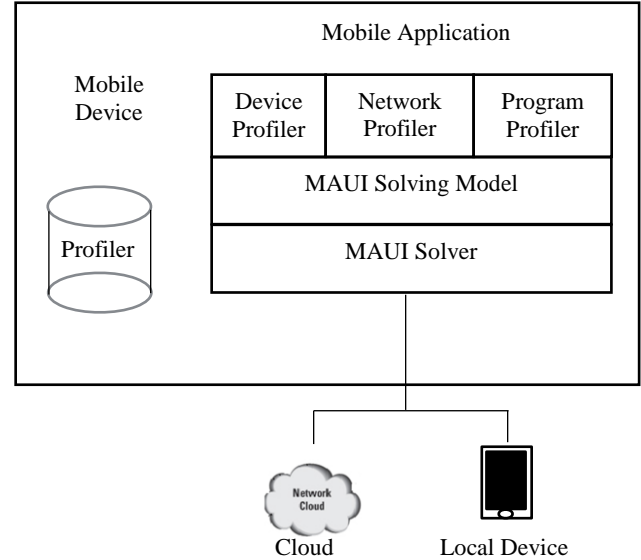


Figure 2. Proposed system architecture

(2) **Network Profiler:** The network profiler includes network properties like latency, bandwidth and so on. It also monitors: (1) cellular connection state and its bandwidth, (2) WiFi connection state and its bandwidth. This data is passed to MAUI solver.

(3) **Program Profiler:** The program profiler includes program characteristics 1) the overall instructions executed, 2) the execution time, 3) the memory allocated, 4) the number of calls of this method and 5) the method execution location (e.g. local, cloud). It is used in the MAUI solver for prediction. MAUI solving model is discussed in the following section.

All these parameters are measured and combined and then they are sent to the MAUI solver for further processing.

MAUI Solver: The MAUI solver uses the informatory report sent by the profilers. Based on the data in the report, MAUI solver uses the MAUI solving model to formulate a mathematical formula which is solved to give either 0 or 1 as an answer. This answer determines the partitioning strategy for method offloading. For each method invocation, a problem is solved, and it is offloaded to the cloud if the answer comes out to be 1, otherwise it is processed locally on the mobile device.

$$S_k = \begin{cases} 0 & \text{mobile} \\ 1 & \text{cloud} \end{cases}$$

This objective function is defined for each mobile device and the offloading decision of certain parts of an application to the cloud or not, depends on the following factors: total preprocessing time, current CPU load, State of Charge (SoC) and network bandwidth.

4.2. MAUI Solving Model

In this section, the details of MAUI solving model and the Computational Offloading Decision algorithm (CODA) are presented. Before presenting the MAUI solving model, the preprocessing step of the system is explained.

In the preprocessing step of the system, we calculate the total computing cost of application on mobile device. The decision problem is to find a solution of selecting where to execute the task and how to offload so that the overall execution time and energy consumption. Let us suppose that we have n number of methods which can be offloaded, $m_1, m_2 \dots m_n$.

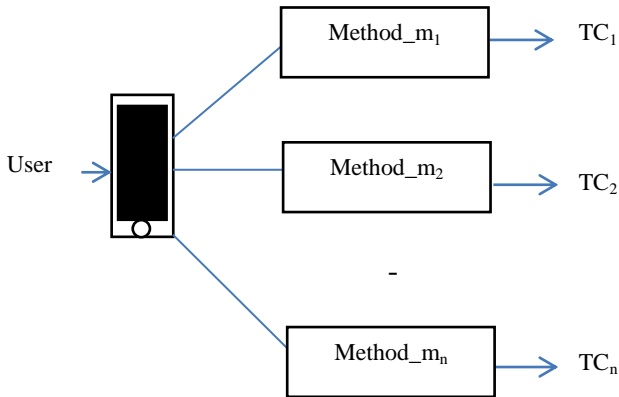


Figure 3. Preprocessing step for execution time of each method

Before presenting the MAUI solving model, the notations and symbols are described in Table 1.

Table 1. Notations and symbols used in MAUI solving model

Symbol	Description
W_n	WiFi network state
C_n	Cellular network state
SoC	State of Charge
TC	Total Execution Time of method
m_1, m_2, \dots, m_n	Methods of Application

m_{i_mem}	Memory usage of method i
m_{i_CPU}	CPU load of method i
m_{i_code}	Code size of method i
$E_{Transfer}$	Round-trip time on remote execution
$T_{Offload}$	Total Cost on offloading application
T_{Local}	Total Cost on local device
S	State of Offloading

The system offloads each of the methods based on several properties i.e. for specific method i, its memory cost m_{i_mem} , CPU load m_{i_CPU} and Code size m_{i_code} . Moreover, the system must be considered the offloading transfer costs for remote execution. Transfer cost includes local device to cloud side cost m_{i_send} and send back to the local device $m_{i_receive}$. Then, the model decides whether the method i is executed locally ($m_i = 0$) or remotely ($m_i = 1$).

The cost function is represented as follows:

(i) Round Trip Time Cost for Remote Execution

$$E_{Transfer} = m_{i_send} + m_{i_execution} + m_{i_receive} \quad (1)$$

(ii) Total Cost for Offloading method m_i on Remote Cloud Side

$$T_{Offload} = \alpha_{ir} E_{Transfer} + \alpha_{mem} m_{i_mem} + \alpha_{CPU} m_{i_CPU} \quad (2)$$

where α_{ir} , α_{CPU} and α_{mem} the weight factors of each cost that the system can adjust the portion of the cost to different scenarios.

(iii) Total Cost for method m_i on Local Device

$$T_{Local} = \alpha_{ir} m_{i_execution} + \alpha_{mem} m_{i_mem} + \alpha_{CPU} m_{i_CPU} \quad (3)$$

(iv) Minimum Cost function of the MAUI solving model

$$\min_{m_i \in \{1, 2, \dots, n\}} Cost(T_{local}, T_{Offload}) \quad (4)$$

4.3. Computational Offloading Decision Algorithm (CODA)

In this section, we explain Computational Offloading Decision Algorithm (CODA) of the system as shown in below.

In the proposed offloading decision algorithm, all the context parameters and profiles are collected from the system components at runtime to provide offloading decision making policies. In preprocessing step, estimate execution cost on client device. Firstly, the

algorithm will check the total execution time (TC) of each method is getting from preprocessing step, if time cost is more than system's constraints the algorithm also need to check the network bandwidth. There are different types of bandwidth which is depended on user choice. As soon as the SoC of the battery gets below 20% of the total capacity of the battery the algorithm is activated to test if the current application can be offloaded. If it is not the case, the application is run on mobile.

Algorithm 1: Computational Offloading Decision Algorithm (CODA)

Input: Set of Context parameter are WiFi network, Cellular network, State of Charge and Total execution time of method. tasks - method $m_1, m_2 \dots m_n$

Output: minimize energy consumption of mobile device

```

1: procedure GetDecision (context, tasks)
2: para[] ← context
3: task[] ← tasks
4: local cost ← estimate execution cost on client device
5: check TC
6: if TC is not valid then
7: check network state
8: else if network is Cn then
9: return decision← minCost(local, offload)
10: else if network is Wn then
11: check Soc
12: if Soc if less 20% then
13: return decision (local execution, null)
14: else
15: return decision← minCost(local, offload)
16: else
17: return decision (local execution, null)

```

5. Performance Evaluation

In this section, the performance of offloading scheme in MCC is demonstrated. Table 2 shows the hardware specification of mobile device. Local device is based on this specifications and cloud side simulation use the cloudSim simulator.

Table 2. Hardware specifications of mobile device

Hardware Components	Specification
Android type and OS	Samsung Galaxy S (Android 2.3)
Memory	512 MB
CPU	1 GHz Cortex-A8

In this experiment, we evaluate the performance of the system based on the CODA algorithm. The input workload size is changed from 10 MB to 50 MB. The output energy consumption level is changed depending

on the input size. We apply CODA algorithm to select the best decision under the current context such as workload size, the device information, network state and execution time of each method to obtain the minimum energy consumption. In this experiment, assume that the network condition is stable. Figure 4 shows the average energy consumption while the input workload size changes. The X-axis represents input size and Y- axis shows an average energy consumption of different situation.

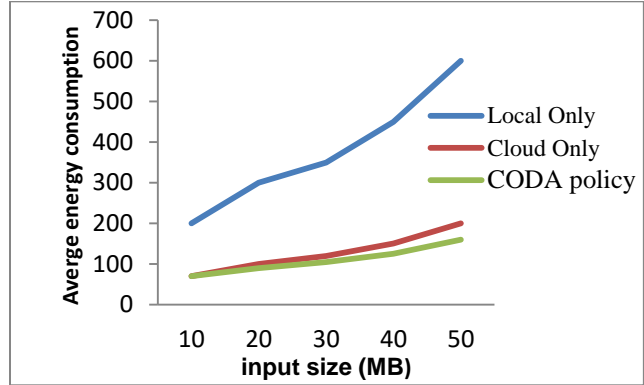


Figure.4. Average energy consumption on various workloads

6. Conclusion

The mobile cloud computing is one of the mobile technology trends in the future because it combines the advantages of both Mobile Computing and Cloud Computing, thereby providing optimal services for mobile users. A vast number of works have been done on the mobile code offloading process which is one of the vital parts of MCC. Even though it's numerous existing frameworks availability, there is much scope which is left for enhancing the offloading process to make it more feasible and attractive. In this paper, we proposed a Computation Offloading Decision Algorithm (CODA) for better performance and less battery consumption.

7. References

[1] A.Ellouze, M.Gagnaire, and A.Haddad,"A mobile application offloading algorithm for mobile cloud computing",, 3rd IEEE International Conference on Mobile Cloud Computing, Service and Engineering , 2015.

[2] B.G.Chun, "Clonecloud: elastic execution between mobile device and cloud", 6th ACM conference on Computer System, 2011, pp.301-314.

[3] B.Zhou, A.V.Dastjerdi, "A context sensitive offloading scheme for mobile cloud computing service", *8th IEEE International Conference on Cloud Computing*, 2015.

[4] C.Eduardo, "MAUI: making smartphones last longer with code offload", *8th ACM International Conference on Mobile System, Application and Services*, 2010, pp.49-62.

[5] D.Kovachev, and R.Klamma. "Framework for Computation Offloading in Mobile Cloud Computing" *International Journal of Interactive Multimedia and Artificial Intelligence*, 2012, pp .6-15.

[6] F.Mehmeti, and T.Spyropoulos, "Performance analysis of mobile data offloading in heterogeneous networks", *IEEE Transaction on Mobile Computing*, vol. 16, no 2, 2017, pp. 482-497.

[7] J.Oueis, E.C.Strinati, and S.Barbarossa, "Multi-parameter decision algorithm for mobile computation offloading", *IEEE*

Wireless Communications and Networking Conference, 2014, pp. 3005-3010.

[8] M.S.Z.Thu, H.M.Kyi, E.C.Htoon, "Computation Offloading Decision in Mobile CloudnComputing: Challenges of Mobile Devices", *15th International Conference on Computer Applications*, 2017, pp.23-27 .

[9] S.M.A.Karim, and John J.Prevoost,l "Efficient mobile computation using the cloud.", *3rd IEEE International Conference on Future Internet of Things and Cloud* , 2015.

[10] T.Truong-Huu, C.K.Tham, and D.Niyato, "To Offload or to Wait: An Opportunistic Offloading Algorithm for Parallel Tasks in a Mobile Cloud", *6th IEEE International Conference on Cloud Computing Technology and Science*, 2014.

Land Use Classification using Deep Convolutional Neural Network

Su Wai Tun, Khin Mo Mo Tun

University of Information Technology, Yangon, Myanmar

suwaitun@uit.edu.mm, khinmomotun@uit.edu.mm

Abstract

One of the challenging issues in high-resolution remote sensing images is classifying land-use scenes with high quality and accuracy. Land use classification is required to measure land and its impact on ecosystem. Deep learning is a powerful state-of-the-art technique for image processing including remote sensing images. Land use is classified for environmental monitoring, urban planning and resource management. This proposed system will use in the UC Merced land-use data set. The preprocessing the image can make the improving of image positional accuracy, reducing the storage space, the improving the spectral qualities of image. The pretrained CNN is initially used to learn deep and robust features. Then, the feature extractor of CNN maps the features and the fully connected layers of CNN are used to obtain excellent results.

Keywords- Classification, Deep Learning, Land Use

1. Introduction

Land-use classification with remote sensing image is always a hot issue in remote sensing technology, which refers to a process that classifies each pixel in remote sensing image into realistic land-use objection. Along with the rapid increase of remote sensing image data and the gradual improvement of resolution, land-use classification with remote sensing image technology plays a more and more important role in urban planning, environmental protection, resource management, mapping and other fields. Deep learning has attracted the interest of many researchers, and becomes a wave of big data and artificial intelligence. Deep neural network simulates the multilayer structures of human brain, abstracts the original data to get features which are applicable for classification. Nowadays, deep learning has achieved great success in recognition of handwritten character, speech and other fields, and offered new thought for land-use classification with remote sensing image. In this paper a approach for land-use classification with remote sensing image based on CNN is proposed, which is verified by the remote sensing data UC-Merced data set.

2. Related Works

In literature, [1] land use classification method based on stack autoencoder has been proposed by Anzi Ding,

Xinmin Zhou. This method is tested in GF-1 images with 4 spectral bands and spatial resolution of 8 m. They show that the method based on SAE is more accurate in classification result than support vector machine and back propagation neural network. In [2], Anqi Wang, Peng Liu and Chao Xie have proposed Markov random field texture classification method. This method is used in German TerraSAR-X radar data. [3]Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel have proposed land cover classification method based on Deep recurrent neural networks. This proposed model has validated on two different data set showing that this framework efficiently deals with both pixel- and object-based classifications. [4]Deep convolutional neural network for land-cover classification method has proposed by Grant J. Scott, R. England, William A. Starns, Richard A. Marcum and Curt H. Davis.

3. Theory Background

The CNN is a trainable multilayer architecture composed of multiple feature-extraction stages. Each stage consists of three layers: 1) a convolutional layer, 2) a nonlinearity layer, and 3) a pooling layer. The architecture of a CNN is designed to take advantage of the two-dimensional structure of the input image. A typical CNN is composed of one, two, or three such feature-extraction stages, followed by one or more traditional, fully connected layers and a final classifier layer. Each layer type is described in the following sections.

3.1 Convolutional Layer

The input to the convolutional layer is a three-dimensional array with r two-dimensional feature maps of size $m \times n$. Each component is denoted as $x_{m,n}$, and each feature map is denoted as x^i . The output is also a three-dimensional array $m_1 \times n_1 \times k$, composed of k feature maps of size $m_1 \times n_1$. The convolutional layer has k trainable filters of size $l \times l \times q$, also called the filter bank W , which connects the input feature map to the output feature map. The convolutional layer computes output feature $z^s = \sum_{t=1}^q W_t^s x^i b_{si}$ where $*$ is a two-dimensional discrete convolution operator and b is a trainable bias parameter.

3.2 Non Linearity Layer

In the traditional CNN, this layer simply consists of a pointwise nonlinearity function applied to each component in a feature map. The nonlinearity layer computes the output feature map $a^s = f(z^s)$, as $f(\cdot)$ is commonly chosen to be a rectified linear unit (ReLU) $f(x) = \max(0, x)$.

3.3 Pooling Layer

The pooling layer involves executing a max operation over the activations within a small spatial region G of each feature map: $P_G^s = \max_{i \in G} a_i^s$. To be more precise, the pooling layer can be thought of as consisting of a grid of pooling units spaced s pixels apart, each summarizing a small spatial region of size $p * p$ centered at the location of the pooling unit. After the multiple feature-extraction stages, the entire network is trained with back propagation of a supervised loss function such as the classic least-squares output, and the target output y is represented as a 1-of-K vector, where K is the number of output and L is the number of layers

$$J(\theta) = \sum_{i=1}^n \left(\frac{1}{2} \| h(x_i, \theta) - y \|^2 \right) + \lambda \sum_1^L \text{sum}(\| \theta^{(1)} \|^2)$$

where l indexes the layer number. CNNs have recently become a popular DL method and have achieved great success in large-scale visual recognition, which has become possible due to the large public image repositories, such as ImageNet.

4. Proposed System

4.1 Image Preprocessing

Preprocessing tasks include geometrically correcting imagery to improve the positional accuracy, compressing imagery to save disk space, converting lidar point cloud data to raster models for speed up rendering in GIS systems and correcting for atmospheric effects to improve the spectral qualities of an image.

4.2 Pretrained CNN

The deep convolutional features learned by pretrained CNN are sufficiently discriminative for land use classification. For classification, CNN will be trained on UC Merced land-use data set by Matlab.

CNN network can learn features and get a better performance even with limited data set.

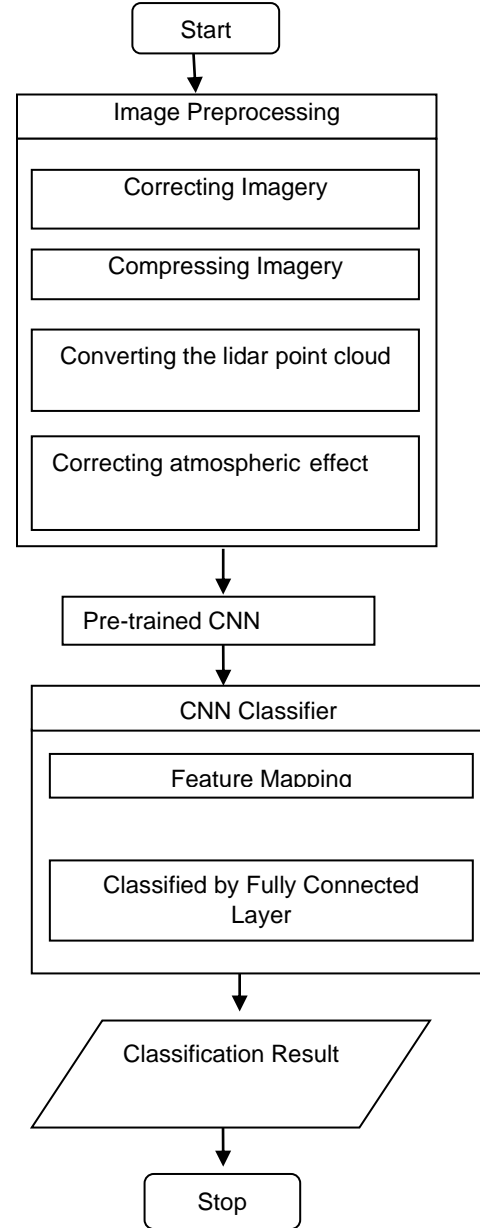


Figure 1. Proposed System

4.3 CNN Classifier

The input image passes to the first convolutional layer. Then, after multiple layers of convolution and padding, the output in the form of a class is needed. The convolution and pooling layers would only be able to extract features and reduce the number of parameters from the original images. The Feature extractor of CNN maps the features. A fully connected layer is needed to generate the final output equal to the number of classes.

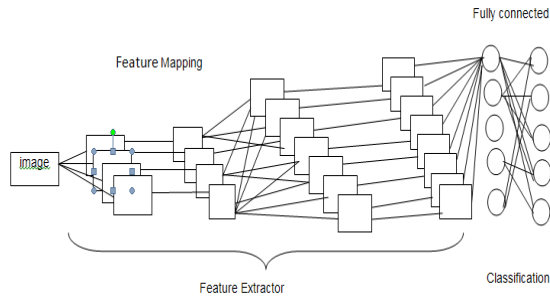


Figure.2 CNN Classifier

4.4 Dataset

The UC Merced land-use data set is investigated, which is a set of aerial orthoimagery with a 0.3048-m pixel resolution extracted from United States Geological Survey national maps. The UCMerced data set has been used as a benchmark for land use classifier evaluation in numerous publications. The data set consists of 21 land-use classes containing a variety of spatial patterns, some with texture and/or color homogeneity and others with heterogeneous presentation. The data set was compiled from a manual selection of 100 images per class, each RGB image being approximately 256×256 pixels. The 21 land-use types include agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis court classes.

5. Conclusion

The Land use and land management practices have a major impact on natural resources including water, soil, nutrients, plants and animals. Land use information can be used to develop solutions for natural resource management issues such as salinity and water quality. In future work, we will try to use the benefit of using convolutional neural network (CCN) to perform land use classification via remote sensing images.

6. References

- [1] Anzi Ding, Xinmin Zhou, "Land-use Classification with Remote Sensing Image Based on Stacked Autoencoder", International Conference on Industrial Informatics, 2016.
- [2] Anqi Wang, Peng Liu, Chao Xie, "Urban Land Use Classification from High-resolution SAR Images Based on Multi-scale Markov Random Field".
- [3] Dino Ienco, Raffaele Gaetano, Claire Dupaquier, and Pierre Maurel, "Land Cover Classification via Multitemporal Spatial Data by Deep Recurrent Neural Networks", IEEE Geoscience and Remote Sensing Letters, 2017.
- [4] Grant J.Scott, Matthew R. England, William A. Strams, Richard A.Marcum and Curt H. Davis, "Training Deep Convolutional Neural Networks for Land-Cover Classification of High Resolution Imagery", IEEE Geoscience and Remote Sensing Letters, 2017.
- [5] Liangpei Zhang, Lefei Zhang, Bo Du, "Deep Learning for Remote Sensing Data", IEEE Geoscience and Remote Sensing Magazine", 2016.

Evaluation of Face Recognition Techniques for Facial Expression Analysis

Hla Myat Maw, K Zin Lin, Myat Thida Mon
University of Information Technology, Yangon Myanmar
hmyatmaw@uit.edu.mm, kzinlin@uit.edu.mm, myattmon@uit.edu.mm

Abstract

Face recognition is an important area in the field of biometrics. It has been an active area of research for several decades, but still remains a challenging problem because of the complexity of the human face. Many recognition methods have been proposed, however, most of them are not able to make use of local salient features to effectively capture the face information. Generally, the performance of face recognition system is determined by extracting feature vector exactly and classifying them into a class accurately. Therefore, it is necessary to pay attention to feature extraction method and classifier. In this paper, we compare and analyze the Principle Component Analysis (PCA), Two Dimensional Principle Component Analysis (2DPCA) and Histogram of Oriented Gradients (HOG) based on the recognition rate and access time from the experimental results. The experiment is done on three sets of databases: the AT&T, Yale and own created face database.

Keywords- Face Recognition, Evaluation, HOG, PCA, 2DPCA

1. Introduction

Face recognition is very important in pattern recognition and image processing which gained much attractive attentions in recent years. It has many applications in a variety of fields, especially in the security systems. Given still or video images of a scene, the recognition system can identify or verify one or more person in the scene using a stored database of faces [1]. The general method of face recognition is shown in Figure 1.

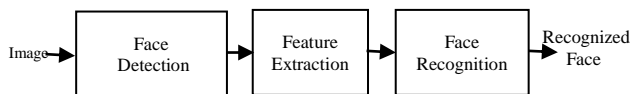


Figure 1. The general method of face recognition system

There are two kinds of algorithms in 2D face recognition field: local feature-based (face components, such as eyes, nose, mouth, etc.) and global feature-based (holistic). Global feature-based methods have been proven to be very successful in face recognition area [2].

These methods are distinctive, robust to occlusion and do not require segment the component parts from face. Many algorithms, for example, Eigenface, Fisherface, etc., are developed and performed well under some limitations, but the variation of faces, such as expression, pose, age, lighting, etc., affects the performance of recognition. HOG is very useful in face and facial expression recognition. It supports irregular shapes and partial occlusions. It is a simple but powerful approach to build robust HOG descriptors. Principal Component Analysis (PCA) [3] is a classical algorithm for face recognition [5]. Two dimensional PCA (2DPCA) [4] is an improvement of PCA in terms of both recognition rate and computation efficiency. Therefore, it's more important to evaluate the HOG algorithm with the other two algorithms using standard methodology.

The rest of this paper is organized as follows: Section 2 gives an overview of the related works. Section 3 presents background theory. Section 4 presents experimental setup. Section 5 presents experiment result and analysis. Section 6 closes with a conclusion and future work.

2. Related works

In [1], Principal Component Analysis is widely used linear subspace image based dimensionality reduction technique. Eigen features calculated here are eigenfaces. Face image, in the form of image vector, is appended column wise. Then the average vector is computed that represents a mean face. Also, a difference vector is computed for each user to qualify the differences to the mean face. Then the covariance matrix of the difference vectors is computed. Finally, principal axes can be obtained by eigen decomposition of covariance matrix. The first N eigenvectors presenting the highest eigen values will be retained and represents the most significant features of faces. Finally, each user model is represented as a linear combination (weighted sum) of coefficients corresponding to each eigenface.

In [2], Two-dimensional Principal Component Analysis (2DPCA) has been proposed and been widely applied in face recognition. Different from the classical PCA, 2DPCA takes a 2D-matrix-based representation model rather than simply the 1D-vector-based one. And image covariance matrix is constructed directly from the 2D image matrices. Since the size of image covariance

matrix is much smaller, 2DPCA can evaluate the matrix accurately and computationally more efficiently than PCA.

In [4], the authors examined two face recognition systems, PCA and 2DPCA algorithms. The feature projection vectors obtained through the PCA and 2DPCA methods and these vectors are applied to test image. The systems used Euclidean Distance based classifier. The results show that recognition accuracy is depended on the number of training sample and number of largest eigenvalues. Additionally, the recognition performance of 2DPCA is higher than the PCA. PCA is the high computational complexity.

In [5], the authors examined PCA and 2DPCA methods was used for face recognition and tested on face image database to evaluate the performance of two algorithms under conditions where the system will recognize faces which are invariant to expression and developing a new feature set to detect mixed emotions- such as happiness and surprise. 2DPCA is much better than PCA. 2DPCA is based on the image matrix, it is simpler and more straightforward to use for image feature extraction and is computationally more efficient than PCA and it can improve the speed of image feature extraction significantly. 2DPCA needs more coefficients for image representation than PCA.

In [6], the authors analyzed the method of Principal Component Analysis (PCA) and its performance when applied to face recognition. This algorithm creates a subspace (face space) where the faces in a database are represented using a reduced number of features called feature vectors. The PCA technique has also been used to identify various facial expressions such as happy, sad, neutral, anger, disgust, fear etc. The results show that PCA based methods provide better face recognition with reasonably low error rates. This is mainly because principal components have proven the capability to provide significant features and reduce the input size of the images.

In [7], the authors presented an overview of different face recognition techniques, studied and analyzed of the face recognition rate, failure rate, training time and recognition time of various face recognition algorithms like PCA, LDA, SVM, ICA and SVD. The results show that SVD recognition rate is highest as well as training time. But SVD consume more recognition time on comparison to PCA, LDA, ICA and SVM.

3. Background theory

3.1. Principle Component Analysis (PCA)

PCA was invented in 1901 by Karl Pearson. It involves a mathematical procedure that transforms the 2D image into 1D feature vector in subspace. This subspace is also called eigenspace in which the covariance matrix is

obtained as a result of facial features. The subspace formed as a result of PCA conversion makes use of facial feature to characterize different reference images or eigenfaces from the sample dataset. PCA, also known as Karhunen-Loeve (KL) transformation or eigenspace is basically a statistical technique used in image recognition and classification. It is also used for image compression. It provides the linear arrangement of template.

The main advantage of this approach is that it is easy to implement, fast and less expensive than any other feature classifier. But it endows invariance information in the presence of varying lighting and scaling condition. The main idea of principal component analysis is to find the vectors which best account for the distribution of the face images within the entire image space. Steps for Feature Extraction:

1. The first step is to obtain a set S with M face images. Each image is transformed into a vector of size N and placed into the set.

$$S = \{ \Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M \} \quad (1)$$

2. Second step is to obtain the mean image Ψ .

$$\text{Mean face: } \Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

3. Then find the difference Φ between the input image and the mean image

$$\Phi_i = \Gamma_i - \Psi \quad (3)$$

4. Next seek a set of M orthonormal vectors, μ_n , which best describes the distribution of the data. The k^{th} vector, μ_k , is chosen such that

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (\mu_k^T \Phi_n)^2 \quad (4)$$

is a maximum, subject to

$$\mu_1^T \mu_k = \delta_{1k} = \begin{cases} 1 & \text{If } 1=k \\ 0 & \text{Otherwise} \end{cases}$$

Where μ_k and λ_k are the eigenvectors and eigenvalues of the covariance matrix C

5. The covariance matrix C has been obtained in the following manner

$$\text{Covariance Matrix: } C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T \quad (5)$$

$$= AA^T$$

$$A = \{ \Phi_1, \Phi_2, \Phi_3, \dots, \Phi_n \}$$

6. To find eigenvectors from the covariance matrix is a huge computational task. Since M is far less than N^2 by N^2 , we can construct the M by M matrix $L = A^T A$

7. Find the M eigenvector, v_l of L .

8. These vectors (v_l) determine linear combinations of the M training set face images to form the eigenfaces μ_l

$$\text{Eigenface: } \mu_l = \sum_{k=1}^M v_{lk} \Phi_k \quad l = 1, 2, \dots, M \quad (6)$$

9. Project each of the original images into eigenspace. This gives a vector of weights representing the contribution of each eigenfaces to the reconstruction of the given image.

$$\omega_k = \mu_k^T (\Gamma - \Psi)$$

$$\Omega^T = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]$$

Where μ_k is the k^{th} eigenvector and ω_k is the k^{th} weight in the vector. $\Omega^T = [\omega_1, \omega_2, \omega_3, \dots, \omega_M]$

3.2. 2D Principal Component Analysis (2DPCA)

A straightforward image projection technique called two-dimensional principal component analysis (2DPCA) is developed for image feature extraction. In contrast to PCA's covariance matrix, the image covariance matrix's size using 2DPCA is much smaller. As a result, 2DPCA has two important advantages over PCA. First, it's easier to evaluate the covariance matrix accurately. Second, less time is required to determine the corresponding eigenvectors they extract the features from the 2DPCA matrix using the optimal projection vector. The vector's size is given by the image's size and the number of coefficients.

Two Dimensional PCA (2DPCA) is an improvement of PCA. 2DPCA does not need to transform 2D face image to 1D vector. However, one disadvantage of 2DPCA (compared to PCA) is that more coefficients are needed to represent an image.

Training Algorithm:

Input: training images

Output: image features, eigenvector matrix, feature matrix

Method

- Apply pre-processing techniques to the M training images
- Obtain the average image A of all training samples:

$$\bar{A} = \frac{1}{M} \sum_{i=1}^M A_i \quad (7)$$

c) Estimate the image covariance (scatter) matrix G :

$$G_t = \frac{1}{M} \sum_{i=1}^M (A_i - \bar{A})^T (A_i - \bar{A}) \quad (8)$$

d) Compute d orthonormal vectors X_1, X_2, \dots, X_d corresponding to the d largest eigenvalues of G . X_1, X_2, \dots, X_d construct a d -dimensional projection subspace.

The optimal projection vectors of 2DPCA, X_1, X_2, \dots, X_d are used for feature extraction.

Project $A_1; \dots; A_M$ on each vector X_1, \dots, X_d to obtain the principal component vectors:

$$Y_i^j = A_j X_i \quad i = 1; \dots; d; j = 1; \dots; M \quad (9)$$

3.3. Histogram of Oriented Gradients (HOG)

The Histogram of Oriented Gradients is a feature based descriptor that was initially proposed for pedestrian detection by Dalal and Triggs. HOG is very useful in facial expression recognition. It supports irregular shapes and partial occlusions. It is a simple but powerful approach to build robust HOG descriptors.

HOG algorithm consists of three steps:

- Gradient Computation.
- Orientation Binning.
- Block Normalization.

Histogram of Oriented Gradients (HOG) is illumination invariant and is found by using magnitude/pixel orientation. Firstly, X and Y gradients of the image is calculated using gradient filter ($G_x = [-1, 0, 1]$, $G_y = [-1, 0, 1]$). Where G_x is the horizontal kernel mask and G_y

Is the vertical kernel mask.

- Centered $f'(x) = \lim_{h \rightarrow 0} \left(\frac{f(x+h) - f(x-h)}{2h} \right)$ (10)

- Gradient

- o Magnitude $s = \sqrt{S_x^2 + S_y^2}$ (11)

- o Orientation $\theta = \arctan\left(\frac{S_y}{S_x}\right)$ (12)

Then using these gradients, corresponding magnitude and angle orientations [ranges 0° - 180° (unsigned) and 0° - 360° (signed)] are calculated. The angular orientations are divided into fragments/parts which are called bins. Secondly, the resulting gradient image is divided into smaller non overlapping spatial regions called cells.

These cells can be rectangular or circular. Each pixel in the cell casts a vote that is weighted by its gradient magnitude and contributes to an orientation aligned with the closest bin in the range 0° - 180° (unsigned) or 0° - 360° (signed). The orientation of bins are evenly spaced and generate an orientation histogram. This step is known as Orientation Binning. After Orientation Binning, the cell histograms are normalized for better invariance to illumination and contrast. This requires grouping of cells into larger and spatially connected blocks. The Histogram of Oriented Gradients descriptor is obtained by concatenating the components of the cell histograms which are normalized from all the block regions. These blocks overlap typically, means that every cell contributes to the final descriptors at least more than once. There are two kinds of block geometries: Rectangular HOG and Circular HOG blocks. R-HOG blocks are rectangular or square grids, which are characterized by three parameters: cells per each block, pixels per each cell and channels per each histogram. In the human face detection experiment conducted by Dalal et.al., the most favorable parameters were observed to be four number of 8×8 pixel cells per each block (16×16 pixels per block) with 9 histogram channels. The R-HOG blocks are quite similar to the SIFT descriptors.

3.4. Classification

Testing image can be classified with training images by calculating the distance or similarity measures between their corresponding feature vectors X and Y ; the smaller the distance between the feature vectors, the more similar are the faces. This paper define a simple similarity score to measure the extent to which the face is recognized and it is calculated as

Euclidean
Distance
$$d(X, Y) = \sqrt{\sum_{i=1}^n (X_i - Y_i)^2} \quad (13)$$

Where X, Y are in the data set X and X_i, Y_i are the i^{th} coordinates of X and Y . This measures the dissimilarity on X .

4. Experimental setup

Face recognition home page [8] supplies over 10 face databases to test the performance about different aspects of faces. In order to test the effect of the number of training images per person, a large range of images per person are needed. Therefore, the two popular and classical face database, AT&T and Yale, are selected. This paper analyzed and classified them according to the main characteristics of databases, such as pose subsets,

expression subsets, etc. In this paper, experiments are based on AT&T face database, Yale face database and own created database are used.

4.1. AT&T face database

AT&T face database contains 40 distinct persons, each person having ten different face images. There are 400 face images in total, with 256 gray and the resolution of 92×112 . These face images are attained in different situations, such as different time, different angles, different expression (closed eyes/open eyes, smile/surprise/angry/happy etc.) and different face details (glasses/no glasses, beard/no beard, different hair style etc.). Figure 2 shows the pictures of two persons with 10 pictures per person on AT&T database.



Figure 2. Sample face images in AT&T database.

4.2. Yale face database

There are 15 persons with 10 different poses, under 64 different illumination conditions. The size of picture is 480×640 . Figure 3 shows the pictures of two persons with 11 pictures per person on Yale database.



Figure 3. Sample face images in Yale database.

4.3. Own created face database

There are 5 persons with 10 different poses, under different situations, such as different angles, different expression and illumination conditions. The size of picture is 135×112 . Figure 4 shows the pictures of two persons with 10 pictures per person on own face database.



Figure 4. Sample face images in own created face database.

4.4. Performance formulas

Recognition performance has many measurement standards. The most important and popular formula is recognition rate in equation (14).

$$\text{Recognition rate\%} = \frac{\text{the number of recognized images}}{\text{the number of testing images}} * 100 \quad (14)$$

5. Experiment result and analysis

In order to evaluate the performance of all the 3 algorithms on the same face database and the effect for different face database.

The experiments were carried out repeatedly as follows: First, the global feature-based features were calculated in training set and testing set; Second, each testing image were matched with the training set by its distance metrics; Third, the average recognition rate and run time were calculated by testing 10 times independently. The results were analyzed in comparison to the PCA, 2DPCA and HOG.

Table 1. Comparison for the recognition rate and access time of PCA, 2DPCA and HOG on AT&T face database

Method	No. of training	No. of test	Recognition rate	Access time
PCA	200	200	78.0%	6.1 sec
2DPCA	200	200	91.5%	5.8 sec
HOG	200	200	87.5%	68.5 sec

Table 2. Comparison for the recognition rate and access time of PCA, 2DPCA and HOG on Yale face database

Method	No. of training	No. of test	Recognition rate	Access time
PCA	75	75	80%	12.5 sec
2DPCA	75	75	86.7%	10.01sec
HOG	75	75	92%	59.26 sec

Table 3. Comparison for the recognition rate and access time of PCA, 2DPCA and HOG on own created face database.

Method	No. of training	No. of test	Recognition rate	Access time
PCA	25	25	100%	1.14 sec
2DPCA	25	25	100%	0.73 sec
HOG	25	25	100%	3.55 sec

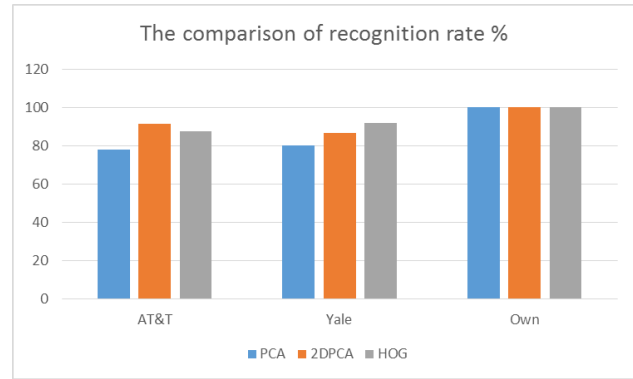


Figure 5. Recognition rate

Figure 5 illustrates the recognition accuracy of PCA, 2DPCA and HOG from each test. This figure indicates that the performance of 2DPCA and HOG are much better than PCA under conditions recognize faces which are invariant to expression and developing a new feature set to detect mixed emotions – such as happiness and surprise.

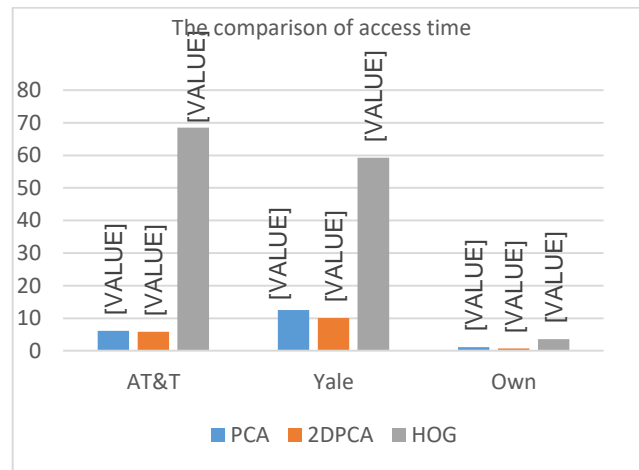


Figure 6. Access time

Figure 6 illustrates the access time of PCA, 2DPCA and HOG from each test. This figure indicates that 2DPCA is computationally more efficient than PCA and HOG. And then it can improve the speed of image feature extraction significantly.

6. Conclusion and Future Work

The paper presents three feature extraction methods: PCA, 2DPCA and HOG. The feature extracted by each of them are used to classify the faces. AT&T face database, Yale and own created datasets are used. Euclidean distance is used to recognize face. The result is compared with different datasets. Experiments demonstrated the recognition rate of HOG is nearly equal with 2DPCA but HOG needs more coefficient and more take times to

access. Future work is to get less time for computation, reduce dimension, consume less memory and increase rate for recognition

7. References

- [1] Zhao W., Chellappa R, Phillips and Rosenfeld, “Face Recognition: A Literature Survey”, ACM Computing Surveys, Vol. 35, No. 4, December 2003, pp. 399–458
- [2] Jian, Yang and A.Z, “Two-dimensional PCA: A new approach to appearance-based face representation and recognition,” IEEE Transaction on Pattern analysis and Machine Intelligence, 26(1), 131–137 (2004).
- [3] Dalal N, Triggs B. “Histograms of oriented gradients for human detection.” In: Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society Conference on IEEE; 2005; 1:886-93.
- [4] Dhiraj K. Das, “Comparative Analysis of PCA and 2DPCA in Face Recognition” *International Journal of Emerging Technology and Advanced Engineering*, ISSN 2250-2459, Volume 2, Issue 1, January 2012.
- [5] Priti Subramaniam, Sapana A Fegade, “Performance Improvement of 2DPCA Algorithm for Face and Facial Expression Recognition” *International Journal of Emerging Trends & Technology in Computer Science (IJETTCs)*, Volume 2, Issue 3, May-June 2013.
- [6] Sukanya Sagarika Meher, Pallavi Maben, “Face Recognition and Facial Expression Identification using PCA”, 2014 IEEE International Advance Computing Conference (IACC).
- [7] Rashmi Ravat, Namrata Dhanda, “Performance Comparison of Face Recognition Algorithm Based on Accuracy Rate”, *International journal of Advanced Research in Computer and Communication Engineering*, Vol. 4, Issue 5, May 2015.
- [8] Mislav.Grgic and Kresimir.Delac, “Face recognition homepage”, <http://www.face-rec.org/databases/> (2008).
- [9] ORL face database, <http://www.uk.research.att.com/facedatabase.html>.
- [10] Yale University face database, <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.

RFSgIndex: Frequent Subgraph Index for Subgraph Matching in RDF Data

Khin Myat Kyu, Kay Thi Yar, Kyawt Kyawt San

University of Information Technology, Yangon, Myanmar

khinmyatkyu@uit.edu.mm, kaythiyar@uit.edu.mm, kyawtkyawtsan@uit.edu.mm

Abstract

Graphs are used to represent the relationships between objects in various domains such as bioinformatics, social networks, and Web exploration. With the rapid increase of RDF data, efficient graph data management technique is needed to store and retrieve these data. Indexing is an effective technique to reduce data searching space and retrieve them as fast as possible. In this paper, RDF frequent subgraph based approach (RFSgIndex) is proposed to find all the matches of a query graph (a SPARQL query) in a given single large graph (RDF data graph). Firstly, frequent subgraphs are extracted from the RDF data graph using Frequent Subgraph Mining algorithm. Given the query graph, the system finds all the possible occurrences of the query graph by using RFSgIndex. The proposed approach will reduce data searching space and speed up the query response time.

Keywords- RDF data, SPARQL query, graph indexing, frequent subgraph mining, subgraph isomorphism

1. Introduction

The Resource Description Framework (RDF) is a standard data model of the Semantic Web, and SPARQL was recommended by W3C as the standard query language to access RDF data. RDF is a flexible, schema-free and graph-structural data model. Today, the size of RDF data is very large. In order to process large-scale RDF data, many RDF systems have been proposed [3], i.e., Jena, Sesame, SW-store, RDF-3X, etc. They store RDF data in relational tables and process SPARQL queries using relational operators, such as scan and join operators. We call them as relation-based RDF stores, because they use the relational model.

The main problem of relation-based RDF stores is that they need too many join operations for processing SPARQL queries, especially for complex graph patterns. Thus, to address this problem, many techniques have been proposed, i.e., the clustered property table [1], vertical partitioning [1], multiple indexing [2]. These techniques have been focused on the optimized storage layout, indexing methods and efficient join processing. However, the graph-structural data model of the RDF and the graph pattern matching nature of SPARQL queries have

significant challenges for efficient processing of SPARQL queries over large-scale RDF data.

In real life applications, we need not only to deal with large database graphs, but also to find all the matches of the query graph. For example, web data networks (social networks, and citation networks, etc.) are often much larger than the graphs used in previous indexing methods. A single RDF data graph may contain millions of vertices. A user may want to find all the occurrences of a particular pattern (subgraph), e.g., name of professors and lectures got their PhD/Master's degrees from universities which are located in London. In different occurrences of the pattern, the name of professors/ lecturers involved may be different since many universities may share the same location. As a result, all occurrences of a particular pattern need to be retrieved. Therefore, we want to solve the following problem for the necessity of real life research—how to find all the matches of a query graph in a large database graph, i.e., finding all subgraphs in the database graph that are isomorphic to the query graph.

Subgraph isomorphism test is believed to be an NP-complete problem and indexing the large graph is practically infeasible. In fact, indexing the large graph is already very difficult. According to our knowledge, many previous methods only apply to graphs of tens of vertices. In addition, most of biological networks, e.g., protein interaction networks, are of thousands of vertices. Unlike to the characteristics of the underlying datasets and the problem definition, algorithms developed for the database of multiple graphs setting may not be used to solve this single graph database problem. Thus, it is necessary to develop a novel solution. When the database is composed of a number of graphs, many of the predominant methods are frequent-substructure based. Researchers preprocess the database and adopt a filter-and-verification process to speed up the subgraph search. False positives are removed by a given pruning strategy. Then, a subgraph isomorphism algorithm is performed on each of the remaining candidates to obtain the final results. These methods have proven to be effective and efficient. However, since we have only a single database graph, the old definition of "frequency" cannot apply. The difficulty of defining "frequency" over a single database graph has been addressed by [8, 9]. Furthermore, since the database graph is much larger than the database graphs for which the frequent subgraph mining tools designed, those tools

may not work at all. The change of the problem setting requires us to define frequency from a new perspective, and at the same time segment the database graph in a meaningful way.

The main difficulty of indexing a single graph with thousands vertices and edges lies in the fact that the subgraph isomorphism is an NP-complete problem. In the traditional database indexing research, the data set size is very large. Thus the goal is to optimize the disk access time. However, in the graph indexing problem setting, the raw graph is not very large, e.g., in the range of megabytes. The computation time to find all occurrences of a subgraph in a graph database is very long. There exist two extreme solutions: (1) Store and index all possible subgraphs of a graph. This is not practically feasible due to the exponential number of possible subgraphs for a graph of thousands edges and nodes, which may require terabyte of storage. (2) Only store the raw database graph. Since the size of the raw database graph is small, it can be easily fit in the main memory. However, the query (matching) time will be very long due to the NP-hard complexity. As a result, we need to identify a solution which lies somewhere between these two extremes, that only utilizes an index structure of a reasonable size and can provide efficient query time.

The remainder of this paper is organized as follows: related work is presented in Section 2. Section 3 defines the preliminary concepts. Section 4 present the proposed approach and describe the algorithm used to build the RFSgIndex. Expected results are presented in Section 5, and the final conclusions are drawn in Section 6.

2. Related Work

Graphs are used to model complex data objects in the real world, e.g., chemical compounds, biological networks, images, social networks and semantic web. Due to its wide usage, it is important to organize, access, and analyze graph data efficiently. As a result, graph database research has attracted a large amount of attention from the database and data mining communities, such as subgraph search in a database of multiple graphs [11, 12, 16, 17, 21], approximate subgraph matching [10, 13, 18], frequent subgraph mining [14, 15, 19, 22], and correlation subgraph query [20].

Among many graph-based applications, it is quite important to retrieve those database graphs containing the query graph efficiently. This is called a subgraph search problem and is closely related to work in this paper. To speed up the subgraph search, researchers preprocess the database and adopt a filter-and-verification framework. First, false positives are removed by a given pruning strategy. Then, a subgraph isomorphism algorithm is performed on each of the remaining candidates to obtain the final results.

Many pruning strategies have been proposed, which can be divided into two sub-categories. The first sub-category is the frequent discriminate substructure based filtering. The approaches in this sub-category apply data mining techniques to extract some discriminating substructures, then build inverted index for each feature.

Query graph q is denoted as a set of features, the pruning power of which is always dependent on the set of selected features. With the inverted indexes, we can find the complete set of candidates.

Many algorithms have been proposed to improve the effectiveness of the selected features, such as gIndex [16], TreePi [21], FG-Index [17] and Tree+ δ [11]. In gIndex, the authors propose a discriminative ratio for features. Only frequent and discriminative subgraphs are chosen as indexed features.

In TreePi, due to the manipulation efficiency of trees, frequent and discriminate subtrees are chosen as feature set. The frequent subgraphs and edges are used as indexed features in FG-Index. In Tree+ δ , the authors use frequent trees and a small number of discriminative subgraphs as indexed features.

The second sub-category is the path, vertex, and neighborhood substructures based filtering, in which no data mining based feature selection is necessary. There are several representative algorithms.

In GraphGrep [12], the authors propose to use all paths up to maxL length as index features. Similarly, GraphGrep also builds inverted index for each path. In Closure-Tree [10], a pseudo subgraph isomorphism test is performed by checking the existence of a semi-perfect match from vertices in the query graph to vertices in a data graph (or graph closure).

In TALE [18], an approximate matching method was proposed for complex query graphs based on neighborhood units. In [13], the authors introduced a pattern matching method based on a combination of techniques: use of neighborhood subgraphs and profiles, joint reduction of the search space, and optimization of the search order.

Since paths, vertices and neighborhood units are less discriminative than the frequent substructures, these algorithms may have less pruning power but better manipulation efficiency.

Most of these techniques only apply to a database of multiple small or medium sized graphs. These databases are large in the sense that they contain many graphs. Many of these methods care more about whether any database graph contains the query graph or not, instead of finding all the matches of the query graph in a given database graph.

3. Preliminaries

In this section, we present the formal data model of the RDF and SPARQL. We assume the existence of three

pairwise disjoint sets: a set of uniform resource identifiers (URIs) U , a set of literals L , and a set of variables VAR . We assume that blank nodes have their local URIs and treat them same as the resources. Variable symbols start with “?” to distinguish them from URIs and literals. An RDF data set is a collection of statements in the form of subject (s), predicate (p), object (o). A statement $t \in U \times U \times (U \cup L)$ (without variables) is called an RDF triple, and a statement $tp \in (U \cup VAR) \times U \times (U \cup L \cup VAR)$ (triple with variables) is called a triple pattern. It should be noted that in our model the joins that have predicate variables are not considered, because this join type is rarely used.

Definition 1. (RDF graph). We define an RDF graph for the RDF database D as $G_D = (V_D, E_D, L_D)$, where V_D is a set of vertices corresponding to the subjects and objects of all triples in D ($V_D \subseteq (U \cup L)$), E_D is a set of directed edges corresponding to all triples that are from the subjects to the objects, and L_D is an edge-label mapping, $L_D : E_D \rightarrow P_D$, such that $t(s, p, o) \in D, L_D(s, o) = p$.

Definition 2. (Query graph) A query graph for a SPARQL query Q is defined as $G_Q = (V_Q, E_Q, L_Q)$, where V_Q is a set of vertices corresponding to the subjects and objects of all triple patterns in Q ($V_Q \subseteq (U \cup L \cup VAR)$), E_Q is a set of directed edges corresponding to all triples that are from the subjects to the objects, and L_Q is an edge-label mapping, $L_Q : E_Q \rightarrow P_D$, such that $tp(s, p, o) \in Q, L_Q(s, o) = p$.

Definition 3. (Frequent graph) A graph is assumed as a frequent graph if its support is larger than the minimum threshold defined by the user.

Definition 4. (Subgraph Isomorphism) Given two graphs, $g = (V, E, L)$ and $g' = (V', E', L')$, a subgraph isomorphism from g to g' is an injective function $f: V \rightarrow V'$, such that $\forall (u, v) \in E, (f(u), f(v)) \in E', L(u) = L'(f(u)), L(v) = L'(f(v))$, and $L(u, v) = L'(f(u), f(v))$.

A graph g is called a subgraph of another graph g' (or g' is a supergraph of g), denoted as $g \subseteq g'$ (or $g' \supseteq g$), if there exists a subgraph isomorphism from g to g' .

4. Proposed Approach

In our proposed approach, there are two main phases: RFSgIndex construction and subgraph (SPARQL) query processing.

4.1. RFSgIndex Construction

We build RFSgIndex using the frequent subgraph mining algorithm, GRAMI, which was originally proposed for use in a single large graph. In this section, we present algorithms **FrequentSubgraphMining** and

SubgraphExtension which are used to generate frequent subgraphs from the given RDF data graph.

Algorithm 1 FrequentSubgraphMining

Input: An RDF database D and $minSup$

Output: All subgraphs S such that $sup(S) \geq minSup$

1. RFSgIndex $\leftarrow \emptyset$
 2. Let frequentEdges be the set of all frequent edges of D
 3. for each $e \in frequentEdges$ do
 4. RFSgIndex $\leftarrow RFSgIndex \cup SubgraphExtension(e, D, minSup, frequentEdges)$
 5. remove e from frequentEdges
 6. return RFSgIndex
-

Algorithm 2 SubgraphExtension (S, D, minSup, frequentEdges)

1. result $\leftarrow S$, candidateSet $\leftarrow \emptyset$
 2. for each e in frequentEdges and nodes of S
 3. if e can be used to extend then
 4. Let ext be the extension of S with e
 5. if ext is not already generated then
 candidateSet \leftarrow candidateSet \cup ext
 6. for each $c \in candidateSet$ do
 7. if $sup(c) \geq minSup$ then
 8. result \leftarrow result \cup c
 9. return result
-

FrequentSubgraphMining firstly identifies set of frequentEdges that contain all frequent edges (i.e., with support greater or equal to $minSup$). For each frequent edge, **SubgraphExtension** is executed. This algorithm takes as input a subgraph S and tries to extend it with the e of frequentEdges (Lines 2-5). All possible extensions are stored in candidateSet. The **SubgraphExtension** (Line 5) algorithm checks all extensions which are already generated or not, which are extended from e . Then, Line 6-8 eliminates the members of candidateSet that less than $minSup$. We assume that their extensions are infrequent. Finally, **SubgraphExtension** is recursively executed to further extend the frequent subgraphs. In this way, all frequent subgraphs are extracted from the RDF data. After all the frequent subgraphs are obtained, the system store them into RDF-3X – open source relation-based RDF store.

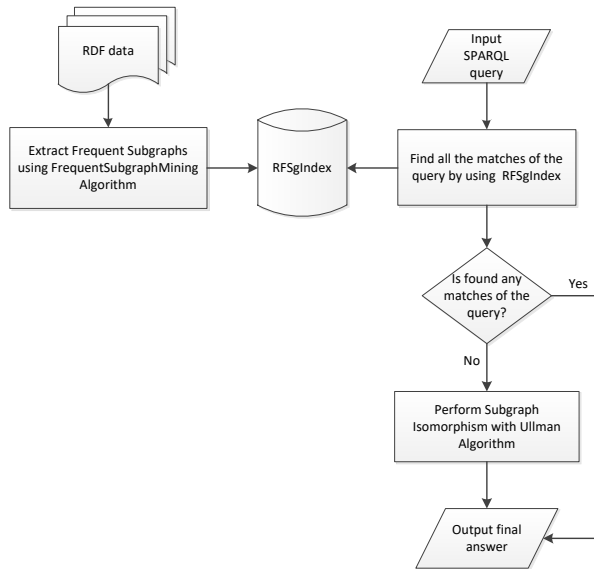


Figure 1. System Flow of Proposed System

4.2. SPARQL query processing

When a SPARQL query is given to the query processor, the query processor finds all the possible matches of the query by using RFSgIndex. If any matches of the query are found in the RFSgIndex, the exact answers of the query are returned. If not found, the query processor have to perform subgraph isomorphism to obtain the final results. We use Ullman algorithm, state-of-the-art subgraph isomorphism algorithm, to find the subgraphs that are isomorphic to the query graph. Since an infrequent graph means the graph occurs in only a small number of graphs in the database, the time of subgraph isomorphism tests will be small.

5. Expected Result and Evaluation

Finding all occurrences of a subgraph in a graph database is a time consuming as the size of the database is very large. At that case, we need to reduce data search space to get efficient query processing. So, this paper proposes the RDF frequent subgraph index (RFSgIndex) for subgraph matching in the RDF data. If the query that is input to the system is found in the RFSgIndex, the proposed approach will reduce data searching space and query execution time. But, even if the query is not found in the RFSgIndex, the time for processing subgraph isomorphism of the query graph is too small because infrequent graphs are rare in the RDF data and will not be queried frequently. So, it does not degrade the overall performance for most queries.

A comprehensive performance study will be conducted using three datasets: Lehigh University Benchmark (LUBM) [23], Yet Another Great Ontology 2 (YAGO2)

[24], and SPARQL Performance Benchmark (SP2B) [25]. LUBM is a benchmark dataset whose domain is the university, YAGO2 is a knowledgebase derived from Wikipedia, WordNet, and GeoNames, and SP2B is a benchmark that simulates the DBLP scenario.

The proposed approach could improve the query processing time by indexing only frequent subgraphs in the RDF data. In our ongoing research, the proposed approach will be verified in terms of index construction time, size of indexes, and query execution time over the three datasets.

6. Conclusion and Future Work

There are various challenges and issues in graph data management. Many different approaches have advantages and disadvantages. Some are focused on storage layout, indexing methods and efficient join processing. In this paper, frequent subgraph indexing (RFSgIndex) is proposed for efficient subgraph matching in RDF data. Indexing only frequent subgraph can get better query processing than indexing all possible subgraph patterns. The proposed method will provide smaller search space and faster query processing.

We will adopt the DFScode canonical form as in gSpan [15] to check extensions which are extended from e are already generated or not (Line 5 of SubgraphExtension algorithm). This will be considered in our future work.

7. References

- [1] M.T. Ozsu, "A survey of RDF data management systems", *Frontiers of Computer Science*, Vol. 10, No. 3, June 2016, pp. 418-432.
- [2] S. Sakr, G. AI-Naymat. "Graph indexing and querying: a review", *International Journal of Web Information Systems*, Vol. 6, No. 2, June 2010, pp. 101-120.
- [3] Y. Luo, F. Picalausa, G.H. Fletcher, J. Hidders, and S. Vansummeren, "Storing and indexing massive RDF datasets", In *Semantic Search over the Web*, Springer Berlin Heidelberg, 2012, pp. 31-60.
- [4] X. Wang, S. Wang, P. Du, and Z. Feng, "CHex: An Efficient RDF Storage and Indexing Scheme for Column-Oriented Databases", *International Journal of Modern Education and Computer Science*, Vol. 3, No. 3, June 2011, p. 55.
- [5] K. Lakshmi, T. Meyyappan, "A comparative study of frequent subgraph mining algorithms", *International*

Journal of Information Technology Convergence and Services, Vol. 2, No. 2, April 2012, p. 23.

[6] F. Abiri, M. Kahani, and F. Zarinkalam, "An Entity Based RDF Indexing Schema using Hadoop and HBase", 4th International Conference on Computer and Knowledge Engineering (ICCKE), Oct 2014, pp. 68-73.

[7] K. Kim, B. Moon, and H. J. Kim, "RG-index: An RDF graph index for efficient SPARQL query processing", Expert Systems with Applications, Vol. 41, August 2014, pp. 4596 – 4607.

[8] B. Bringmann, S. Nijssen, "What Is Frequent in a Single Graph?", Advances in Knowledge Discovery and Data Mining, 2008, pp. 858-863.

[9] M. Fiedler, C. Borgelt, "Subgraph Support in a Single Large Graph", IEEE 7th International Conference on Data Mining (ICDM) Workshops, Oct 2007, pp. 399-404.

[10] H. He, A.K. Singh, "Closure-Tree: an index structure for graph queries", Proceedings of the 22nd International Conference on Data Engineering (ICDE), April 2006, p. 38.

[11] P. Zhao, J.X. Yu, and P.S. Yu, "Graph indexing: tree + delta \leq graph", Proceedings of the 33rd International Conference on Very Large Data Bases (VLDB), Sep 2007, pp. 938-949.

[12] R. Giugno, D. Shasha, "GraphGrep: A Fast and Universal Method for Querying Graphs", Proceedings of 16th International Conference on Pattern Recognition (ICPR), Vol. 2, 2002, pp. 112-115.

[13] H. He, A.K. Singh, "Graphs-at-a-time: Query Language and Access Methods for Graph Databases", Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, June 2008, pp. 405-418.

[14] M. Kuramochi, G. Karypis, "Frequent subgraph discovery", Proceedings IEEE International Conference on Data Mining (ICDM 2001), 2001, pp. 313-320.

[15] X. Yan, J. Han, "gSpan: graph-based substructure pattern mining", Proceedings IEEE International Conference on Data Mining (ICDM 2003), 2003, pp. 721-724.

[16] X. Yan, P.S. Yu, and J. Han, "Graph indexing, a frequent structure-based approach", Proceedings of the

2004 ACM SIGMOD International Conference on Management of Data, June 2004, pp. 335-346.

[17] J. Cheng, Y. Ke, W. Ng, and A. Lu, "FG-Index: Towards verification-free query processing on graph databases", Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data, 11 June 2007, pp. 857-872.

[18] Y. Tian, J.M. Patel, "TALE: A Tool for Approximate Large Graph Matching", IEEE 24th International Conference on Data Engineering, 7 April 2008, pp. 963-972.

[19] M. Koyuturk, A. Grama, and W. Szpankowski. "An efficient algorithm for detecting frequent subgraphs in biological networks", Bioinformatics, Vol. 20, 4 August 2004, pp. i200-i207.

[20] Y. Ke, J. Cheng, and W. Ng, "Correlation search in graph databases", Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 12 August 2007, pp. 390-399.

[21] S. Zhang, M. Hu, and J. Yang. "Treepi: A novel graph indexing method", IEEE 23rd International Conference on Data Engineering (ICDE 2007), 15 April 2007, pp. 966-975.

[22] A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data", Principles of Data Mining and Knowledge Discovery, 2000, pp. 13-23.

[23] Y. Guo, Z. Pan, and J. Heflin, "LUBM: A benchmark for OWL knowledge base systems", Web Semantics: Science, Services and Agents on the World Wide Web, Vol. 3, 31 Oct 2005, pp. 158-182.

[24] J. Hoffart, F.M. Suchanek, K. Berberich, and G. Weikum, "YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia", Artificial Intelligence, 2012, pp. 28-61.

[25] M. Schmidt, T. Hornung, G. Lausen, and C. Pinkel, "SP2Bench: A SPARQL performance benchmark", Proceedings of the 25th International Conference on Data Engineering (ICDE), 29 March 2009, pp. 222-233.

[26] M. Elseidy, E. Abdelhamid, S. Skiadopoulou, and P. Kalnis, "Grami: Frequent subgraph and pattern mining in a single large graph", Proceedings of the VLDB Endowment, Vol. 7, No. 7, 1 March 2014, pp.517-528.

A Functional Resonance Analysis Method to risk analysis of functional flood defenses in Yangon

Kyi Pyar Hlaing, Nyein Thwet Thwet Aung, Swe Zin Hlaing
University of Information Technology, Yangon, Myanmar
kyipyarhlaing@uit.edu.mm, nyeinthwet@uit.edu.mm, swezin@uit.edu.mm

Abstract

Multifunctional use of flood defences is seen as a promising solution for improving the synergy between flood protection and urban development combining the functions can, however, create unintended dependencies, which can influence the desired performance of the system in unexpected ways. Recognizing the risks associated with these dependencies early during the conceptual design phase can help to improve the system capability to mitigate the resulting threats and to take advantage of the opportunities created. The proposed systems use the Functional Resonance Analysis Method (FRAM) for qualitative risk analysis of multifunctional flood defences. The analysis results are used to identify the threats and opportunities that need attention during the design of a multifunctional flood defence and to propose recommendations for how to address them.

Keywords: Multifunctional flood defences, Risk analysis, Socio-technical system, Flexibility, Uncertainty, Functional modeling Dependence.

1. Introduction

Multifunctional use of flood defence is proposed as a promising solution for dealing with the conflicts of flood protection and urban development as well as enhancing the cost effectiveness of reinforcement interventions [1].

Both the negative and positive impacts arising from these changes have to be taken into account to plan not only for minimizing the unwanted negative outcomes, but also to take advantage of the opportunities for improving the system performance [2, 3]. Conducting such a risk analysis early during the conceptual design phase can help the designers to proactively identify and handle these potential risks.

Multifunctionality can induce dependencies between the system components, which leads to complexities in risk analysis of such a system [4]. Once the functions are combined, they become part of a broader socio-technical context in which the well-/mal-functioning of the system depends not only on its technical performance, but also on the role of humans as operators, inspectors, and users of the system [5].

This research investigates the application of the Functional Resonance Analysis Method (FRAM) [6] for qualitative risk analysis of multifunctional flood defences. The term 'risk' is used in this research to denote the uncertain outcomes that could be either positive or negative. The objective is to identify how the dependencies caused by the multifunctional use of flood defences can strengthen or weaken the desired performance of the system when there is a change in its working environment. FRAM is selected because it enables modeling both negative and positive events resulting from (intended and unintended) dependencies between the functional components of a multifunctional flood defence. The premise of FRAM is based on the generic steps of the system analysis to analyze the system functions by breaking apart the system into the functional components that are relatively well known, identifying the dependencies between the components, and investigating the impacts of the dependencies on system performance.

2. Background Theory

2.1. System Definition

From a structural point of view, [7] state that A multifunctional flood defence often consists of at least two objects: a water retaining structure for flood protection and a secondary structure placed in close vicinity of the flood defence which is not intended for flood protection. a multifunctional flood defence as a combination of functions such as transport, housing, agriculture, nature and recreation with the primary function of flood protection.

In this proposed system, the working definition of multifunctional flood defence refers to:

A zone that is primarily used for flood protection, but serves other non-water retaining functions (e.g. transportation, housing).

In principle, there is no limit to the number and type of functions that can be combined with the flood protection function. The combination of the function(s)

is considered as multifunctional only if the structure of the secondary function (secondary object) is located partly or fully in one of the standard flood protection zones around the flood defences.

2.2. System dependencies

Multifunctionality does not only refer to a high concentration of several activities in a relatively small space, but also implies that it induces various types of relationships between the combined functions. If these created relationships are such that the state of one function of the system becomes reliant on or is influenced by the state of another one, then there is a dependency between them [8].

Ref. [9] classifies the intended relationships among the infrastructure components based on the mechanisms that connect them. Ref. [10] selects and indicates the physical (or functional) and geographical relationships as the most relevant types of dependency to be considered for water related infrastructures. These two types can also reflect the intended relationships between the components of a multifunctional flood defence, which are caused by combining and relating the functions and/or co-locating and connecting the associated structures.

Physical dependency refers to the situation in which the state of one function is intentionally designed to be dependent on the other functions. Geographical dependency occurs where the structural elements of a system are co-located in such a way that a local environmental event can affect all elements.

3. Methodology

The effectiveness of flood defences in reducing the risk of flooding is well-known although ensuring their desired performance involves significant challenges. One challenge is that present methods are not able to fully describe and predict the performance of a single flood defence under controlled conditions[11]. Another challenge is that the operating environment of flood defences changes constantly and is associated with uncertainties. Combining other functions with the primary function of flood protection further complicates the matter. The intended physical and geographical dependencies add new relationships between the system components and their operating environment, which can influence the desired performances of a multifunctional flood defence. Identifying these potential dependencies during the early development phase of multifunctional flood defences can help to improve the system design to handle unexpected outcomes.

3.1. The ‘Functional Resonance Analysis Method’

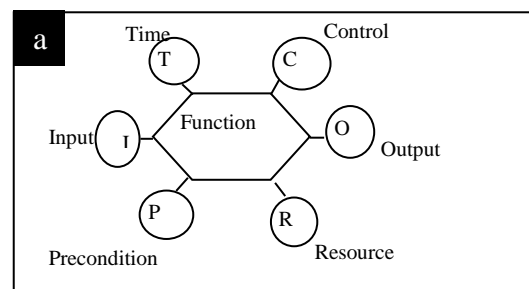
FRAM uses a novel representation of the system performance based on the concept of functional resonance that originates from wave theory in physics. The term ‘stochastic resonance’ is transferred to describe the variability of performance of the functions within a sociotechnical system. It is claimed that the inevitable changes in a system and its working environment can lead to variability in the performance of individual functions. Propagation and aggregation of the performance variability caused by the dependencies between the functions may result in unintended outcomes. The functional model of the system is developed and used to identify the potential dependencies between the functions for specific (retrospective or prospective) scenarios. In short, FRAM is implemented in four steps as follows:

Step 1: Identifying and describing the functions

The premise of FRAM is the decomposition of the system into its functional entities, including the technical, operational, and organizational activities, which are involved in the day to day work of the system to succeed. The functions are characterized by the six aspects of Input (I), Output (O), Precondition (P), Resource (R), Time (T), and Control (C) and are visualized as shown in Fig. 1a. The six functional aspects are linked together to address the dependencies between the human technical activities during the specified scenarios as shown in Fig. 1b.

Step 2: Characterizing the performance variability

The second step of FRAM determines the possible sources and types of variability for the individual functions. The potential sources of variability in the outcome of a function can be related to change in one of the six aspects of the function itself; the aspects of the other functions; and the operating environment. The type of variability indicates how the outcome of a function may change and is characterized qualitatively in terms of a time (too early, on time, too late, not at all) and precision (precise, acceptable, and imprecise).



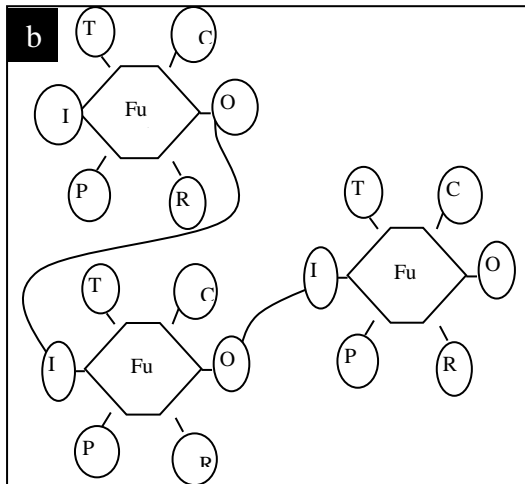


Figure.1. (a) The graphical representation of the six functional aspects; (b) a demonstration of the functional dependencies as represented by the connecting lines.

Step 3: Aggregation of performance variability

This step is aimed at identifying the potential dependencies that propagate the variability and the aggregation of variability leading to unexpected (either positive or negative) outcomes based on the description of a particular scenario. This aggregation is also called ‘functional resonance’. Any detected possible functional resonance, for the specified event (or scenario), is taken as a discernible ‘signal’ of a threat or opportunity.

Step 4: Responding to performance variability

The functional model of the preceding steps is used to identify proper strategies (elimination, prevention, protection and facilitation) to cope with possible occurrences of uncontrolled performance variability. Thus far, FRAM has been predominantly applied to retrospective safety and accident investigations, where the primary focus is on variability of human-centered functions.

4. Proposed System

The proposed system use FRAM is because it is well suited for representing the complex relationships between the functional components of socio-technical systems [12]. This method is used to derive the potential dependencies between the functional components of a multifunctional flood defence in order to provide input for risk analysis [13,14], demonstrating both the threatening and opportunistic outcomes[15,16]. The FRAM method has the limitation because of the lack of sufficient information and expert availability. So this proposed system aim to get more detailed analysis of flood defence.

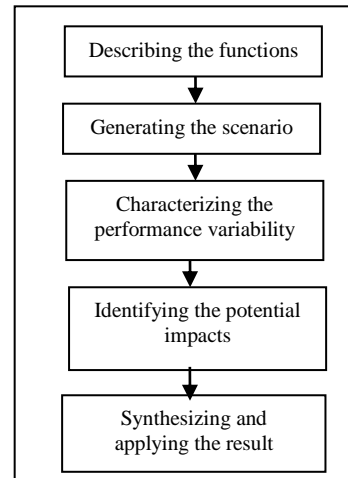


Figure 2. Proposed system design

5. Conclusion

The objective of system is to enhance the risk analysis of multifunctional flood defences by developing a tool for the system designers to explore how the flood protection and secondary function can mutually impact each other positively and negatively. Functional modeling of multifunctional flood defences by developing a tool for the system designers to explore how the flood protection and secondary function can mutually impact each other positively and negatively. This system examines the application of the Functional Resonance Analysis Method’ (FRAM) to the risk analysis of multifunctional flood defences. For the purpose of the, FRAM is customized into five steps for describing, characterizing and visualizing the functions of a multifunctional flood system and their dependencies. The method provides a qualitative tool for a broader view, analysis, and visualization of many imaginable internal and external changes to the system including various types of human, technical, and environment interactions. The proposed system aim to get more detailed analysis of flood defence.

6. Reference

[1] De Groot R. Function-analysis and valuation as a tool to assess land use conflicts in planning for sustainable, multi-functional landscapes. *Landsc Urban Plan*2006;75:175–86.

[2] Hollnagel E, Prologue: the scope of resilience engineering. *Resilience engineering in practice: a guidebook*, 2011. p. xxix–ix.

[3] Rogers JW, Louis GE. Risk and opportunity in upgrading the US drinking water infrastructure system. *J Environ Manag* 2008;87:26–36.

- [4] Johansson J, Hassel H. An approach for modeling interdependent infrastructures in the context of vulnerability analysis. *Reliab Eng Syst Saf* 2010; 95:1335–44.
- [5] Comfort LK. Risk, security, and disaster management. *Annu Rev PolitSci* 2005; 8:335–56.
- [6] Egan MJ. Anticipating future vulnerability: defining characteristics of increasingly critical infrastructure-like systems. *J Contingencies Crisis Manag* 2007; 15:4–17.
- [7] Van Veelen PC, Voorendt MZ, Van Der Zwet C. Design challenges of multifunctional flood Defences. A comparative approach to assess spatial and structural integration. *Res Urban Ser* 2015;3:275-92.
- [8] Caporaso JA. Dependence, dependency, and power in the global system: a structural and behavioral analysis. *Int Organ* 1978;32:13-43.
- [9] Rinaldi SM, Peerenboom JP, Kelly TK. Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Syst IEEE* 2001;21:11-25.
- [10] Zimmerman R. Understanding the implications of critical infrastructure interdependencies for water. *Wiley Handbook of Science and Technology for Homeland Security*; 2009.
- [11] Buijs FA, Simm J, Wallis M, Sayers P. Performance and Reliability of Flood and Coastal Defences; 2007.
- [12] Clay-Williams R, Hounsgaard J, Hollnagel E. Where the rubber meets the road: using FRAM to align work-as-imagined with work-as-done when implementing clinical guidelines. *Implement Sci* 2015;10:125.
- [13] Woltjer R, Hollnagel E. Functional modeling for risk assessment of automation in a changing air traffic management environment. In: *Proceedings of the 4th international conference working on safety*. 2008.
- [14] Frost B, MO JP. System hazard analysis of a complex socio-technical system: the functional resonance analysis method in hazard identification. *The Australian System Safety Conference (ASSC 2014)*. Melbourne, Australia; 2014.
- [15] Hollnagel E. *FRAM: the functional resonance analysis method: modelling complex socio-technical systems*. Ashgate Publishing, Ltd; 2012.
- [16] Lundblad K, Speziali J, Woltjer R. FRAM as a risk assessment method for nuclear fuel transportation. In: *Proceedings of the 4th international conference working on safety*. Crete, Greece; 2008.

Feature Extraction Method for Aspect-Based Sentiment Analysis

Win Lei Kay Khine, Nyein Thwet Thwet Aung, Thet Thet Zin

University of Information Technology, Yangon, Myanmar

winleikkhine@uit.edu.mm, nyeinthwet@uit.edu.mm, thetthetzin@uit.edu.mm

Abstract

In our daily life, we take opinions of our friends and we are influenced in decision making process. Opinion is the view or the judgment about something. Opinion Mining (OM) or Sentiment Analysis (SA) is the computational analysis of public's opinion, emotion, sentiments, and attitude toward entities and their attributes expressed in written text. These entities may be products, services, organizations, individuals, events, issues, or topics. In sentiment analysis, formal and informal opinion text like product reviews, news articles, tweets, forum discussions, blogs, and Facebook posts are also applicable to all domains. The main purpose of sentiment analysis is to extract the main opinions, on which the decision can be made very right. Paper intends to classify sentiment polarity on product review datasets by using Mutual Information as a feature selection method. Because product reviews are highly focused and they are opinion rich. After the feature selection, we aim to classify the extracted features with Naïve Bayes, SVM and Maximum Entropy to get the accurate sentiment polarity.

Key Words- Sentiment Analysis, Opinion Mining, Feature Selection, Feature Extraction, Aspect-Based Sentiment Analysis

1. Introduction

Sentiment analysis has been an active research field of natural language processing (NLP) for finding sentiments of people for a specific product, services, films, news, organizations and so on. It aims to mine this information to find out the popular sentiment about any product and its associated features. The main goal is to determine whether the expressed opinion in the text is positive, negative or neutral. Neutral opinion usually means "no opinion". The Sentiment analysis can be done by classifying the text on document level, sentence level and feature or aspect level. Document level only gives the whole document polarity; it considers the opinion of only one opinion holder. Sentence level deals with multiple sentences, but can distinguish only subjectivity or objectivity of sentences which is not equivalent to derive sentiment of particular feature. It

can only determine sentence's polarity that is positive, negative and neutral. If there are more than one object and feature with complex sentences then the above levels failed to get accurate sentiment analysis. Neither document-level nor sentence-level analyses discover what people like and dislike exactly. For it, Feature level sentiment analysis is adopted not only dealing with the polarity classification of particular features but also handling many problems as coordinating conjunctive sentences and comparative sentences which are quite tough to extract the opinion. This level of analysis was earlier called feature-based, which is now called aspect-based sentiment analysis. The goal of this level is to discover sentiment on entities and/or their aspects. In this paper, we intend to use aspect-based sentiment analysis because sentiment classification on both document and sentence levels are not sufficient: they do not tell what people like or dislike. The main objectives are to extract people's opinion about a particular product and produce a feature-based opinion summary of multiple reviews.

2. Related Works

In the literature [4] proposed a novel feature reduction method using standard deviation based on more variation or dispersion of features in feature space. They used three popular classifiers, namely: Naïve Bayes, Maximum Entropy and Support Vector Machine for sentiment classification and ensemble of these classifiers. They then compared their proposed method with other feature reduction methods used on book and music reviews.

According to Tuba Parlar, Selma Ayse Özel [5], they proposed a new feature selection method called "Query Expansion Ranking" that is based on query expansion term weighting methods. They compared their Query Expansion Ranking with Chi Square method and Document Frequency Difference (DFD). Experiments are conducted on four Turkish product review datasets that are Book, DVDs, electronics and kitchen appliances reviews by using a supervised machine learning classification method, namely Naïve Bayes Multinomial classifier. Finally, their new feature selector improves classification accuracy better than Chi Square and Document Frequency Difference.

Bagheri et al. [7] proposed a sentiment classification model for the document level of cell phone reviews. They presented a new feature selection approach MMI (Modified Mutual Information) based on the Mutual Information method and applied three feature selection approaches, MI, Term Frequency Variance (TFV) and MMI with the Naïve Bayes learning algorithm. To test their method, they compiled a dataset of 829 online customer reviews in Persian language from different brands of cell phone products including Nokia, Apple, Samsung, Sony, LG, Motorola, Huawei and HTC. Their proposed approach, MMI in overall can reach to 85% of F-score classification correctly and it overcomes other techniques. (MI-58% and TFV-84%)

J. Ashok Kumar and S. Abirami [8] implemented the OMSA approach and analysed the results by using a single dataset for different feature extraction or selection techniques namely single word, Multiword, Document Level, Phrase Level, Tf-idf single word and Tf-idf Multiword. Experimental procedure has been carried out with an extension of the OMSA approach. In this approach, the Polarity Classification Algorithm (PCA) and evaluation procedure is applied to verify the accuracy. The evaluation procedure is tested with four different datasets.

According to my review, many researchers did sentiment analysis on many fields using various methods. This paper intends to do sentiment analysis with high accuracy. Accuracy will increase with excellent features for relative domain and classification methods. Therefore, this paper aims feature selection and classification for sentiment analysis.

3. Theory Background

Sentiment Analysis has been a good research area from a long time. It is a very challenging task in these days, which is a great requirement in every field as in Political field, Marketing field and in social field mainly. Many techniques can be found for sentiment analysis. It can be done based on Document level, Sentence level and Aspect or Feature level. But the first two levels didn't consider object features that have been commented in a sentence. So the aspect/feature level sentiment analysis is more appropriate compared to both levels. In this paper, feature extraction method for product review dataset will be proposed. There are so many feature selection techniques for Feature Extraction. These are document frequency, Information Gain, Chi-Square, TF-IDF, Information Mutual, Standard deviation, etc... . Here are some of the feature selection methods:

3.1. Feature Extraction

There are so many feature selection methods like uni-gram, bigrams, Information Gain (IG), Chi-square (CHI), Document Frequency (DF), Information Mutual (IM) and so on. The main task of feature selection and feature reduction/extraction is reduction dimension in feature space. It causes the removal of irrelevant features and results in the following outcomes: more efficient categories; easier analysis of sentiment after reduction; visualization of results; and there may be a better perception of low dimension.

3.1.1. Document Frequency. In the Document Frequency (DF) method, features are ordered by document frequency for each feature in a whole document. This method is the simplest measure for feature reduction and has a linear time complexity capable of scaling a large dataset.

3.1.2. Information gain (IG). Information gain is the most commonly used feature selection method in the field of machine learning. It calculates the relevance of a feature for prediction of sentiment of review by analyzing the presence or absence of a feature in a document.

3.1.3. Chi-square (CHI). Chi square measures the lack of independence between a feature and a class. If a feature f in the related class c has a low score, it can be less informative, so it can be removed. [5]

The challenges in feature extraction in sentiment analysis are facing different issues like large feature space problems, redundancy, domain dependency, difficulty in implicit feature identification. After the feature extraction, we will use classifiers to evaluate the results. For the classification, the most commonly used algorithms are Naïve Bayes (NB), Support Vector Machine (SVM), Decision Trees and Maximum Entropy (ME) classifiers are used to classify the polarity of a given text of the feature.

3.2. Classification Techniques

3.2.1. Naïve Bayes. The Bayesian classification is a statistical method underlying a probabilistic model and supervised learning algorithms. Naive Bayes (NB) uses a features vector matrix to determine a document depending upon polarity classes (i.e. positive and negative classes) by probability. It attaches a document to the relevant class with the highest probability.

3.2.2. Support Vector Machine. Support Vector machine (SVM) is a most popular algorithm that can classify data as either linear or nonlinear. It can also

map input data to high dimensional feature spaces, in addition to classifiers' SVM support regression, binary and multiclass classification respectively. The support vector can be either linear or nonlinear.

3.2.3. Decision Tree. It is a tree in which internal nodes are represented by features, edges represent tests to be done at feature weights and leaf nodes represent categories which results from above tests. It categorizes a document by starting at the tree root and moving successfully downward via the branches (whose conditions are satisfied by the document) until a leaf node is reached. The document is then classified in the category that labels the leaf node. Decision Trees have been used in many applications in speech and language processing [6].

3.2.4. Maximum Entropy. Maximum Entropy (ME) classifier is one of the machine learning methods used for natural language processing applications, as it is implemented using a multinomial logit model as the classifier rule. ME is a kind of statistical inference that can be used to estimate any probability distributions on the partial knowledge.

4. Proposed System

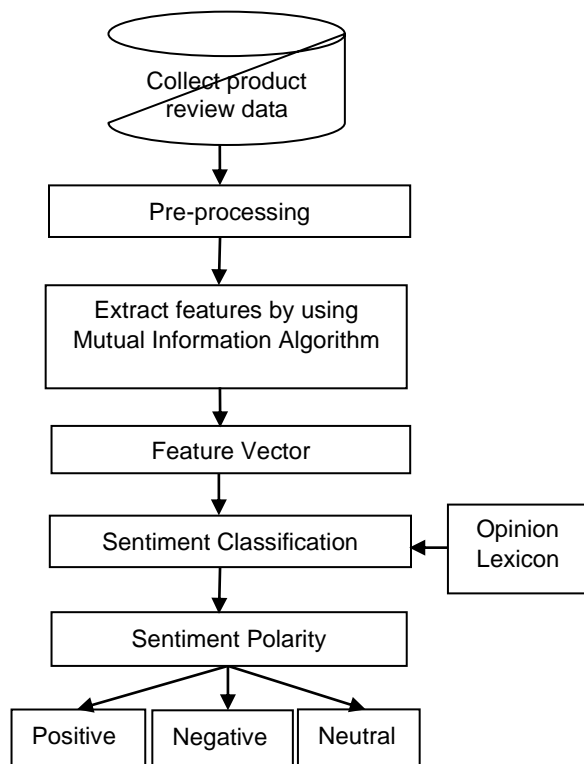


Figure 1. Proposed system architecture

Figure 1 gives overall system architecture for sentiment analysis. Firstly, we collect the products' review data from Amazon.com. This data may be noisy and hence we need to be pre-processed. Secondly, the pre-processing step is used for the removal of stop words and special characters. After the pre-processing phase, we extract the relevance features by using the "Mutual Information" feature selection method. This paper intends to modify the Mutual Information Algorithm to get the right features for classification phase. Feature selection tries to find a subset of features that are important and valuable for the classification task. Finally, the classifier is applied to test whether it is able to detect the right features or give the right classification. This paper intends to get the higher accuracy for selected features that are extracted by Mutual Information than by using other methods.

4.1. Datasets

The data collection process is the first stage in this step. In this stage, a freely available dataset will be used for preprocessing the data. The collected dataset serves as input to the pre-processing stage and further the feature selection or extraction method has been applied over it to classify the polarity into positive, negative, and neutral. In this step, the datasets were downloaded from [9]. It contains 28000 reviews for book information, for example. The attributes in this book dataset include RowID, ProductID, Publisher, ReleaseDate, ProductDimensions, ShippingWeight, Language, NumPages, Type Edition and FullDesc. Other datasets are from [10].

4.2. Preprocessing phase

The preprocessing phase involves the following steps:

4.2.1. Tokenizing. In the pre-processing phase, reviews are scanned to extract tokens consisting of words and numbers.

4.2.2. Removal of Stop Words. Stop words do not have any sentiment information, so we need to remove stop words in the pre-processing step i.e. words such as she, he, at, about, of, in, on, the, etc...

4.2.3. POS Tagging. POS tagging assigns a tag to each word in a text and classifies a word to a specific morphological category such as noun, verb, adjective, adverb, etc.

4.2.4. Stemming. The stemming process converts all the inflected words present in the text into a root form

called a stem. For example, 'automatic,' 'automate,' and 'automation' are each converted into the stem 'automat.'

4.3. Feature Selection and Extraction

The aim of feature selection is to remove the irrelevant and redundant features, so it can produce better prediction accuracy and better efficiency. In this paper, we will use the Mutual Information (MI) as a feature extraction. MI is one of the most effective approaches for optimal feature extraction. It measures the mutual dependence of two or more variables. In this context, the feature extraction processing a feature vector from the data which have the largest dependency. MI selects features that are not uniformly distributed among the sentiment classes because they are informative of their classes. And we can see that MI giving more importance to the rare term.

4.4. Sentiment Classification

In this paper, Naïve Bayes, Support Vector Machine and Maximum Entropy will be used as a classifier to get the accurate the sentiment polarity with high accuracy. The experimental will be evaluated by precision, recall and F-Score methods.

5. Conclusion

Sentiment analysis is one of the widest areas of research and improvement with techniques and classification approaches. In this paper, feature extraction in sentiment analysis will be proposed. Feature extraction in sentiment analysis is now becoming an active area of research. Feature selection methods can help to improve the classification performance of sentiment analysis in terms of both accuracy and run time. For the future work, we will try to use the Mutual Information (MI) as a feature selection. After that, Naïve Bayes, Support Vector Machine and Maximum Entropy are used to compare the results.

6. References

- [1] Bing Liu, "Sentiment Analysis and Opinion Mining", Morgan & Claypool Publishers, May 2012.
- [2] R.Rajput, A.K.Solanki, "Review of Sentiment Analysis Methods using Lexicon Based Approach", International Journal of Computer Science and Mobile Computing, India, 2016, pp. 159-166.
- [3] T.Shaikh, Dr.D.Deshpande, "Feature Selection Methods in Sentiment Classification of Amazon Product Reviews", International Journal of Computer Trends and Technology (IJCTT)-Volume 36 Number 4, ISSN:2331-2803, India, June 2016, pp 225-230.
- [4] A.Yousefpour, R.Ibrahim, H.Nuzly, A.Hamed, "A Novel Feature Reduction Method in Sentiment Analysis", International Journal of Innovative Computing 4, 2014, pp 34-40.
- [5] T.Parlar, S.A.Ozel, "A New Feature Selection Method for Sentiment Analysis of Turkish Reviews", IEEE, 2016.
- [6] N.S.Joshi, S.A.Itkat, "A Survey on Feature Level Sentiment Analysis", International Journal of Computer Science and Information Technologies, Vol.5(4) India, 2014, pp 5422-5425.
- [7] A.Bagheri, M.Sarace, F.de Jong, "Sentiment Classification in Persian: Introducing a Mutual Information-based Method for Feature Selection", IEEE, 2013.
- [8] J. A. Kumar, S. Abirami, "An Experimental Study of Feature Extraction Techniques in Opinion Mining", International Journal on Soft Computing, Artificial Intelligence and Applications (IJSCAI), Vol.4, No.1, India, February 2015, pp 15-21.
- [9] <http://liu.cs.uic.edu/download/data/>.
- [10] <http://jmcauley.ucsd.edu/data/amazon/links.html>.

A Personalized Recommendation System Using Collaborative Filtering With Feature Based Sentiment Analysis

Nyein Ei Ei Kyaw, Thinn Thinn Wai, Thiri Haymar Kyaw

University of Information Technology, Yangon, Myanmar

nyeineieikyaw@uit.edu.mm, thinnthinnwai@uit.edu.mm, thirihaymarkyaw@uit.edu.mm

Abstract

Recommendation systems help users to deal with the information overload problem by producing personalized content according to their interests. For presenting the personalized recommendation according to the new user's demand is big challenge. Beyond the traditional recommender strategies, there is a growing effort to incorporate users' reviews into the recommendation process, since they provide a rich set of information regarding both items' features and users' preferences. This proposal proposes a recommender system that uses users' reviews and preference of new users to meet the individual interest.

Keywords- Recommender system, Preferences, Review

1. Introduction

Today, people currently live in an era of information. They are surrounded by a excess of data in the form of reviews, blogs, papers and comments on various websites. With the success of the Web 2.0, more and more companies capture large-scale information about their customers, providers, and operations. In recent years Recommender systems are software applications that attempt to reduce information overload by recommending items of interest to end users based on their preferences. With the growing number of alternative services, effectively recommending services that users preferred have become an important research issue. Service recommender systems have been shown as valuable tools to help users deal with services overload and provide appropriate recommendations to them. Examples of such practical applications include CDs, books, web pages and various other products now use recommender systems. Over the last decade, there has been much research done both in industry and academe on developing new approaches for service recommender systems.

2. Related Works

Several papers have addressed the problems to meet the personalized requirement of user in various ways.

Pallavi R. Desai, B. A. Tidke proposed a system that present personalized recommendation list and recommend the most appropriate items to the user by using weights of keywords are used to indicate user' preferences and a user-based collaborative filtering algorithm is adopted with OpenNlp to generate appropriate recommendations [1]. Khushboo R. Shrote, Prof. A.V. Deorankar proposed a system in which feedback analysis is done using sentiment analysis to recommend services. Keywords are used to indicate what the users prefer [2]. Susan Thomas, Jayalekshmi S proposed a system in which sentimental analysis on the reviews is done using Naïve Bayes, a machine learning technique to distinguish between the positive and negative reviews. It also use MonoDB database to store the review detail [3].Shakhy.P.S1, Swapna.H2 proposed a recommendation system which considers not only user reviews but also the temporal information about location of the services. It use Apache Mahout learning library and MongoDB to store reviews [4]. Dr. Kogilavani Shanmugavadivel and their colleagues proposed a system deals with the implementation of personalized rating to the services for hotel reservation system and booking of cars. This system performed opinion mining on the review at the sentence level using Bayes theorem and negation rule algorithm [5].

All the above papers based on the previous users' reviews and new user preferences are considered as keyword. Sentiment analysis is performed on the reviews by using machine learning algorithms and then similarly of previous users' keyword set and active user keyword set are computed and finally recommend the top k services to the new user. The candidate service set that system provide and similar terms associated with candidate service are manually specified.

Therefore, this proposal intends to construct the domain ontology concerned with terms and related terms of the domain and perform sentiment analysis on the review and recommend the top k services to the user in order to meet the need of user more accurately. This proposal consists of the following parts: (1) A user-based collaborative filtering recommendation system, (2) domain ontology is constructed for features identification of the domain (3) Sentiment analysis on the reviews is done using Lexicon based approach.

3. Background Theory

Personalized recommender systems help users to find a wide variety of products online, and assist users in making decisions. At such highly rated Internet sites as Amazon.com, YouTube, Netflix, Yahoo, TripAdvisor, Last.fm, and IMDb, recommender systems play a very important role. Users could take advantage of these recommender systems to find a variety of products, videos, books, and news that they like from the massive available item set [8]

3.1 Recommendation System

Recommendation System, a sub-class of information filtering system, helps in predicting top-N preferred items for a user. Recommendation system can be classified into content based approach, collaborative filtering approach and hybrid approach. Content based recommendation systems will recommend items based on the description of the items and profile of the user. Collaborative filtering recommendation system will recommend items based on the similarity between the users who have rated the same item before. This methods build a model using information about past purchases or ratings provided by users. This model may be used for prediction of preference rating for a given item. Hybrid approach is a combination of content based and collaborative filtering approaches.

3.2 Collaborative Filtering (CF)

Collaborative filtering methods analyze large amount of information about preferences of users and predict preferences of similar users for recommending items. Collaborative filtering (CF) methods are of two types: item based and user based collaborative filtering. Item based CF, recommend items based on the similarity between the items rated by the same user in the past. User based CF, recommend items based on the similarity between the users who have rated the same items.

3.3 Similarity Computation

Similarity computation between items or users is a critical step in collaborative filtering algorithms. There are many different methods to compute similarity or weight between users or items such as Pearson correlation, Vector Cosine-Based Similarity, Euclidean distance, Minkowski distance, Cosine similarity.

3.4 Sentiment Analysis

Opinion mining (Sentiment Analysis) is a Natural Language Processing (NLP) and Information Extraction (IE) task that aims to obtain the feelings of the writer expressed as positive, negative or neutral opinions by analyzing large number of documents.

Sentiment Classification techniques can be divided into machine learning approach, lexicon based approach and hybrid approach. Machine learning approach is based on machine learning algorithms. Lexicon based approach is based on sentiment lexicon and hybrid approach combines both approaches. Three level of sentiment analysis are:

1. Document Level: The whole file contains group opinion. The file verifies whether it conveys the positive or negative sentiment.
2. Sentence Level: Verifies that whether a sentence conveys positive, negative or neutral meaning.
3. Entity and Aspect Level: Product features are defined as product attributes or components. Analysis of such features for identifying sentiment of the document is called as feature based sentiment analysis. In this approach positive or negative opinion is identified from the already extracted features.

3.5 Ontology

Ontology is a formal representation of the knowledge by a set of concepts within a domain and the relationships between those concepts. It is used to represent the properties of that domain and may be used to describe the domain. Domain ontology is one kind of ontology which is used to represent the knowledge for a particular type of application domain (e.g., a product domain).

Domain-Ontology structure mainly consists of several attributes of ontology concept. Concept is the abstract of concrete objects. The attributes of concept are used to describe the characteristics of various aspects of concept. For example, film is a domain concept, director, actor, and released age are attributes of the concept [6].

4. Overview of the system

In the proposed system, the reviews of specific domain are firstly collected. Then preprocessing step is carried out on the review. In this stage stop words and HTML tags are removed from the reviews. Then the words are converted to the root forms of the word. That is words such as “computing”, “computation” and ”computes” have the root form as “compute”. This process of getting the root form of the words is called stemming. Porter Stemmer algorithm is used for stemming.

After that, domain ontology is constructed to collect the terms and related similar terms of the domain. It contains the candidate services and reference work of the candidate service set which contains terms with similar meanings of the specific domain. The system then performs the opinion mining step to assign the polarity of each feature from the domain ontology based on SentiWordnet. Similarity computation of previous user and active user preference are performed on the keyword set of both users using Cosine similarity measures. Finally, personalized rating is calculated and top k recommendation lists are presented to the user.

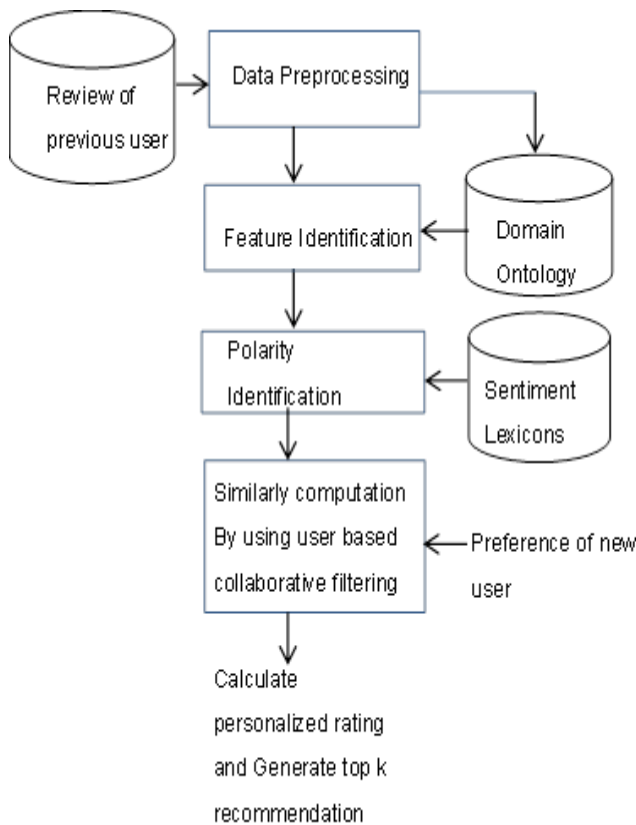


Figure1. Proposed system architecture

5. Conclusion

Most existing service recommender systems are only based on a single numerical rating to represent a service's utility as a whole. In fact, evaluating a service through

multiple criteria and taking into account of user review can help to make more effective recommendations for the users. Therefore, the proposed system will be implemented the personalized recommendation system by constructing domain ontology and sentiment analysis on the review to meet the individual user interest.

6. References

- [1] Pallavi R. Desai, B. A. Tidke , “A Survey on Smart Service Recommendation System by Applying Map Reduce Techniques”, International Journal of Science and Research (IJSR) 2014
- [2] Khushboo R. Shrote, Prof. A.V. Deorankar, ”Sentiment Analysis Based Feedback Analysed Service Recommendation method For Big Data Applications”, International Journal of Scientific & Engineering Research 2016
- [3] Susan Thomas, Jayalekshmi S., “Recommendation System with Sentimental Analysis using Keyword Search”, international journal for advance research in engineering and technology 2015
- [4] Shakhy.P.S1, Swapna.H2, ”Improved Keyword Aware Service Recommendation System for Big Data Applications”, International Journal of Innovative Research in Computer and Communication Engineering 2015
- [5] Dr. Kogilavani Shanmugavadeivel, Dr. Thangarajan Ramasamy , Dr. Malliga Subramanian, ” Semantic Ranking Based Service Recommendation System using MapReduce on Big Datasets” , International Journal of Advances in Computer and Electronics Engineering 2017
- [6] Cheng Xiao, Dequan Zheng, Yuhang Yang, Automatic Domain-Ontology Structure and Example Acquisition from Semi-Structured Texts Sixth International Conference on Fuzzy Systems and Knowledge Discovery 2009
- [7] Saikat Bagchi, “Performance and Quality Assessment of Similarity Measures in Collaborative Filtering Using Mahout”, 2nd International Symposium on Big Data and Cloud Computing
- [8] F. Ricci, L. Rokach, B. Shapira, P.B. Kantor (Eds.), Recommender Systems Handbook, Springer, 2011, <http://www.springerlink.com/content/978-0-387-85819-7>.

Efficient Classification of Concept Drift in Data Stream

Ei Thwe Khaing
University of Information Technology
eithwekhaing@uit.edu.mm

Abstract

The classification in data streams is widely studied in the literature over the last decade. In recent literature, many research contributions use incremental or progressive learning strategies to classify the data streams. Stream classification is a variant of incremental learning of classifiers that has to satisfy requirements specific for massive streams of data. There are many methods such as single classifiers, windowing techniques, drift detectors and ensemble methods. The classifier ensembles provide a way of adapting to changes by modifying ensemble components or their aggregation method. Adaptive Classifier Ensemble (ACE) method use to provide a natural way of adapting to change by modifying ensemble members. This method improves more accuracy and adaptable than other ensemble methods.

Keywords – Data Stream Mining, Classification, Concept drift

1. Introduction

A data stream is a massive, infinite, temporally ordered, continuous and rapid sequence of data elements. Research on data stream is motivated by emerging applications involving massive data sets such as customer click streams, supermarket, telephone records, stock market, meteorological research, multimedia data, scientific and engineering experiments and sensor data. A new generation of mining algorithms are needed for real time analysis and query response for these applications since most conventional data mining algorithms can only be applied to static data sets that may be updated periodically in large chunks, but not in continuous streams of data [1].

While data mining has become a fairly well established field now, the data stream problem poses a number of unique challenges which are not easily solved by traditional data mining methods. Some of issues of data stream like dynamic nature, infinite size and high speed, unbounded memory requirements, Lack of global view, handling the continuous flow of data impose a great challenge for the researchers dealing with streaming data. Unlike traditional data sets, it is impossible to store an entire data stream or to scan through it multiple times due

to its tremendous volume. New concepts may keep on evolving in data streams at different times, to deal with this any data stream processing algorithm have to continuously update their models to adapt to the changes.

When data streams are large, a new computational requirement needs a limited amount of memory and a short processing time in processing the continuous incoming data stream. Data stream situations pose several further requirements on learning algorithms as compared with learning from static data. It is usually impossible to store all the data from streams and only a small part of the data can be stored and used for computations within a limited time span. In most cases, the arrival speed of the incoming instances from data streams enforce their processing in the real time. The data stream mining should have the following properties: high accuracy, fast adaptation to change and low computation cost in both space and time dimensions.

Concept drift is the distribution generating the items of a data stream that changes over time. The concept drift is assumed to be unpredictable, periodic seasonality is usually not considered as a concept drift problem. If seasonality is not known certainty, it might be regarded as a concept drift problem. The core assumption, when dealing with the concept drift problem, is uncertainty about the future that the source of the target instance is not known with certainty. So, the Adaptive Classifier Ensemble (ACE) method in classifier ensembles provide concepts as adapting to changes by modifying ensemble.

The challenging issues of learning ensembles concept drift from data streams. Many classifier ensembles are typically the most often applied approaches in data stream analysis. Traditional learning methods (neural networks, Naive Bayes, nearest neighbor methods, and decision rules) are only able to process data sequentially, but do not adapt, can be easily modified to react to change.

In the further section of this article, Section 2 provides the related works to classify streaming data. Then, Section 3 provides a brief description of the data stream mining technologies. Next, Section 4 describes proposed system in streaming data. Finally, Section 5 summarizes this paper.

2. Related Works

In case of data streams, the amount of distinct options or things that exist would be massive so this

makes even the more number of cache memory or system memory are not appropriate for storing the whole stream data. The major drawbacks of data streams are the speed. Speed of information stream arrival is relatively higher than the speed of information store and process.

The paper [2] mentioned mining data streams with concept drift. The goal of the paper is to propose and validate a new approach to mining data streams with concept-drift using the ensemble classifier constructed from the one-class base classifiers. The base classifiers of the proposed ensemble are induced from incoming chunks of the data stream. Each chunk consists of prototypes and can be updated using instance selection technique when a new data have arrived. When a new data chunk is formed, ensemble model is also updated on the basis of weights assigned to each one-class classifier. The prototype selection is a promising research direction when looking for effective stream mining tools. Other research will also focus on studying influence of the size of both - ensemble model and data chunk on accuracy of the ensemble classifier.

The paper [3] described non-stationary characteristics of streaming data, prediction models are often also required to adapt to concept drifts. It surveys research on ensembles for data stream classification as well as regression tasks. Besides presenting a comprehensive spectrum of ensemble approaches for data streams, we also discuss advanced learning concepts such as imbalanced data streams, novelty detection, active and semi- supervised learning, complex data representations and structured outputs.

The paper [4] proposed a new approach in weighting ensemble components applied to stream data classification. Two theorems were presented that give the foundations for this approach. The user confidence define in these theorems. If the additional components are large, we will obtain an increase of accuracy of the whole ensemble for the entire infinite data stream. This approach is based on the observation that probability of the correct tree outcome is different in various tree sections. It achieves increasing the accuracy of the whole ensemble.

3. Data Stream Mining

Data Stream mining is the process of extracting knowledge structures from such continuous, rapid data records. Mining data streams raises new problems for the data mining community about how to mine continuous high-speed data items that you can only have one look at. Due to this reason, traditional data mining approach needs to be changed and to discover knowledge or patterns from data streams , it is necessary to develop single-scan, on-line , multilevel , multi-dimensional stream processing and analysis methods. Various procedures for extraction of information from data streams were proposed in concern with data mining.

A. Clustering

Envision an enormous measure of dynamic stream information. Numerous applications require the computerized clustering of such information into segments depending on their likeness. In spite of the fact that there are numerous effective grouping algorithms for static information sets, grouping or dividing data streams puts extra imperatives on such calculations, as any information stream model obliges algorithms to make a single pass over the information, with limited memory and constrained calculation time.

B. Classification

There are many strategies to classify static information. This is a two stage process comprising of model development from preparing data and arrangement where the model is utilized to foresee the class names of tuples from new information sets. In a conventional setting, the training information dwell in a generally static database so scanning can be carried out many times, yet in stream information, the information stream is fast to the point that capacity to store them and scanning it several times is infeasible. Another characteristic is time varying in data streams, instead of conventional database frameworks, where just the present state is stored. This change in the nature of the data takes the form of changes in the objective classification model after some time and is alluded to as concept drift. It is a vital thought when managing stream data.

C. Association

There are two stages in algorithms for the association rule. The initial step is to find continuous item sets. All continuous item sets meet the threshold value are found and the second step is to infer association rules. In this progression, in light of the continuous item sets found in the initial step, the rules that meet the certainty basis are inferred. Nevertheless, customary association standard mining calculations are produced to take a shot at static information and, along these lines, can't be connected straight forwardly to mine association rules in stream information. New researches are directed on the most proficient method to get frequently occurring elements, association rules and various patterns in the environment of stream of data.

In these methods, this paper proposes data stream mining in classification. Data Stream Classification is a traditional supervised machine learning task. Both tasks are concerned with the problem of predicting a nominal value of an unlabeled instance represented by a vector of characteristics. The main difference between these tasks is that, in streaming

scenarios, instances are not readily available to the classifier as being part of a large static dataset, and, instead, instances are provided sequentially and rapidly over time as a continuous data stream. Therefore, a data stream classifier must be ready to deal with a great number of instances, such that each instance can only be inspected once or stored for only a short period of time.

4. Proposed System

Concept drift is the distribution generating the items of a data stream that changes over time. The concept drift is assumed to be unpredictable, periodic seasonality is usually not considered as a concept drift problem. If seasonality is not known certainty, it might be regarded as a concept drift problem. The core assumption, when dealing with the concept drift problem, is uncertainty about the future that the source of the target instance is not known with certainty. It can be assumed, estimated, or predicted, but there is no certainty.

The challenging issues of learning ensembles concept drift from data streams. Many classifier ensembles are typically the most often applied approaches in data stream analysis. Traditional learning methods (neural networks, Naive Bayes, nearest neighbor methods, and decision rules) are only able to process data sequentially, but do not adapt, can be easily modified to react to change.

A computational effective algorithm needs to be adapting of concept drift in non-stationary data stream. This paper presents a new algorithm, which adapts very quickly to concept drifts, and has been specifically designed to deal with concepts. We compare our new algorithm with various well-known learning algorithms, taking into data streaming datasets from UCI Machine Learning Repository.

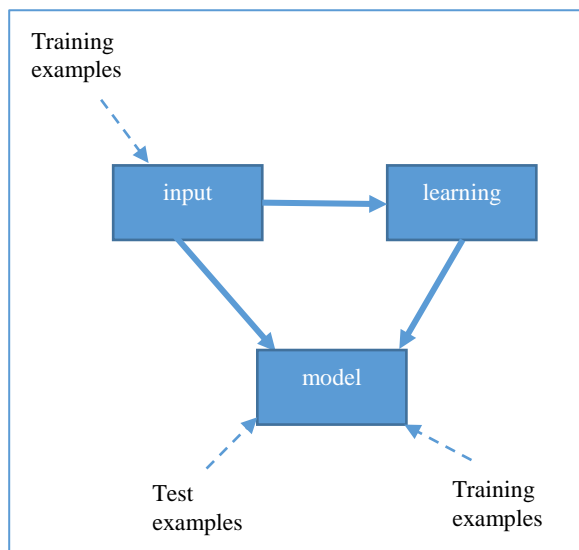


Figure 1. Overview of Data Stream Classification

5. Conclusion

The concept drifts are often unstable and change over time. The underlying data distribution may change as well. Often these changes make the model built on old data inconsistent with the new data and an updating of the model is necessary.

References

[1] R. Kalaivani, Dr. S.Vijayarani. Data Stream Mining – A Survey. IJIRCCE, Vol. 5, Issue 4, April 2017, DOI:10.15680/IJIRCCE.2017.0504172.

[2] Ireneusz Czarnowski, Piotr Jedrzejowicz, Ensemble classifier for mining data streams, *Procedia Computer Science* 35 (2014) 397 – 406.

[3] Bartosz Krawczyk, Leandro L. Minku, Joao Gama, Jerzy Stefanowski, MichalWozniak. Ensemble learning for data stream analysis: A survey. *Information Fusion* 37 (2017) 132–156.

[4] Lena Pietruczuk, Leszek Rutkowski, Maciej Jaworski, Piotr Duda, How to Adjust an Ensemble Size in Stream Data Mining?, *Information Sciences* (2016), doi:10.1016/j.ins.2016.10.028.



SHWETAUNG

ACE
Data Systems Ltd.



flymya.com

ISBN 978-99971-0-381-9



9 789997 103819 >