WORKFLOW-BASED INTEGRATION OF SERVICES ON A PRIVATE CLOUD SYSTEM

Khine Khine Htwe, Thinn Thu Naing

University of Computer Studies, Yangon, Myanmar
khinehtwe@gmail.com, ucsy21@most.gov.mm

Abstract

Integration processes gain in importance today. Service-oriented Architecture plays a vital role in the modern design of business and academic IT landscapes. The integration of services still remains among the hard SOA challenges. The need for addressing them becomes more pressing with the increase in the number of online (cloud) services. This system also defines the possible scenarios to focus the service integration. Integration processes in the academic private cloud intend to interoperate between different department units and different user groups according to their roles and responsibility. The main contribution of this paper is that Petri Nets Workflow Model is applied for possible scenarios to do the service integration and it is going to test on Eucalyptus open source private cloud architecture. As a result, the development effort can be reduced and the portability. And the performance can also be improved significantly.

Keywords: Service-oriented Architecture; Service Integration; Workflow; Petri Nets; Eucalyptus

1 Introduction

Software as a Service (SaaS) is defined in as a software application delivery model, where a software vendor deploys and hosts software applications in a multi tenant (cloud) platform for its customers to operate the application over the Internet as services. In recent years, SaaS has emerged as a new paradigm for software delivery in software cloud, attracting more and more interest from both industry and academia. SaaS applications can be deployed in a cloud computing environment and accessed through Internet by web browsers [1].

Technically cloud computing can be described as virtual centralization of distributed system. In academic cloud computing can be instrumental. It can give the academic institution a new way to provide students and researchers the computational facility. Nowadays, it will incur a huge expense for individual to have enough computational facility needed for their academic and research purpose.

Small academic functions are encapsulated in loosely-coupled services which are combined to executable workflow. They allow a flexible integration of academic or enterprise IT services. This will increase interoperability between the different domains. It also increases between the services inside one domain and ease switching between service providers.

A cloud application influences cloud computing in software architecture, often eliminating the need to install and run the application on the customer's own computer. In turn, it minimizes the burden of software maintenance, ongoing operation, and support. It comprises web mail, face book, security as a service, software as a service, storage, etc.

The specification of a workflow can be done by a Petri nets. Petri nets have been studied intensively in computer science. The description of workflows by Petri nets offers the following advantages:

1) Petri nets are well suited to represent discrete dynamic models. A workflow combined with its execution is such a model because the points in time of the execution of the activities form a discrete set in comparison to the time axis which is continuous.
2) A Petri net has a solid mathematical definition, its syntax and semantics are precisely defined.
3) If a workflow is mapped onto a Petri net, certain properties such as the reachability of an end marking is mathematically provable.
4) There are a large number of analysis methods which deal with the verification of Petri nets.

The rest of this paper is organized as follows. Section 2 describes the related works. Proposed system overview is explained thoroughly in Section 3. Components of academic private cloud service architecture are described in Section 4 and Integration session is also explained in Section 5. Possible scenarios for workflow-based integration and case study are presented in Section 6 and 7. Finally Section 8 concludes the paper.

978-1-61284-159-5/11/$26.00 ©2011IEEE

2 Related works

Orchestrating Secure Workflows for Cloud and Grid Services describes the requirements, design, implementation, and evaluation of a new orchestration architecture that facilitates the integration of stateful and stateless services across organizational boundaries [2]. Control Plane Integration for Cloud Services is discussed in [3]. This paper addresses the problem of control plane integration for management and control of cloud services. Solution Marketplace for Service Composition and Integration is expressed in [4]. The solution reuse at a large scale can be exploited to address challenges of service composition and integration by harnessing the collective intelligence and labour of various businesses and people present on the Internet. SaaS and Integration Best Practices are surveyed in [5]. It highlights the emerging integration technologies that can help ease the burden of integration SaaS applications. Integration as a service solution is beginning to simplify integrations, especially in the cloud to cloud space. Cloud-based Enterprise Mashup Integration Services for B2B Scenarios is in [6]. Enterprise Mashup and Lightweight Composition approaches and tools are promising solutions to unleash the huge potential of integrating the mass of end users into development. A Taxonomy of Workflow Management Systems for Grid Computing is explained in [7]. In this paper scientists and engineers are building more and more complex applications to manage and process large data sets, and execute scientific experiments on distributed resources. Enterprise Cloud Service Architecture explains Cloud computing, a new paradigm of distributed computing, introduces many new ideas, concepts, principals, technologies and architectural styles into enterprise service-oriented computing [8].

3 Proposed system overview

3.1 Eucalyptus and its components

Eucalyptus architecture is deployed with some components: Cluster Controller (CLC) as front-end interface component, the several Cluster Controllers (CCs) in which Storage Controllers (SCs) are attached to provide the ESB block storage. Then the several Node Controllers (NCs) are working as back-end nodes. According to networking architecture point of view, the front-end node is configured with two network interfaces: one is connected to public campus network and another one is connected to private VM networks into Node Controllers (back-end node). Eucalyptus components are as follows:

- ❖ Cloud Controller (CLC)
- ❖ Walrus Storage Controller (WS3)
- ❖ Storage Controller (SC)
- ❖ Cluster Controller (CC)
- ❖ Node Controller (NC)

3.2 Proposed system framework

This system implements an academic private cloud, with its own private services. There are a lot of service providers in this private cloud according to their duties. They are service providers such as Administrator, Faculty, and Staff. Service Integration process will be performed based on the following: Intra-cloud wise (service integration inside only private cloud) and Inter-cloud wise (service integration between different clouds; public cloud and private cloud).

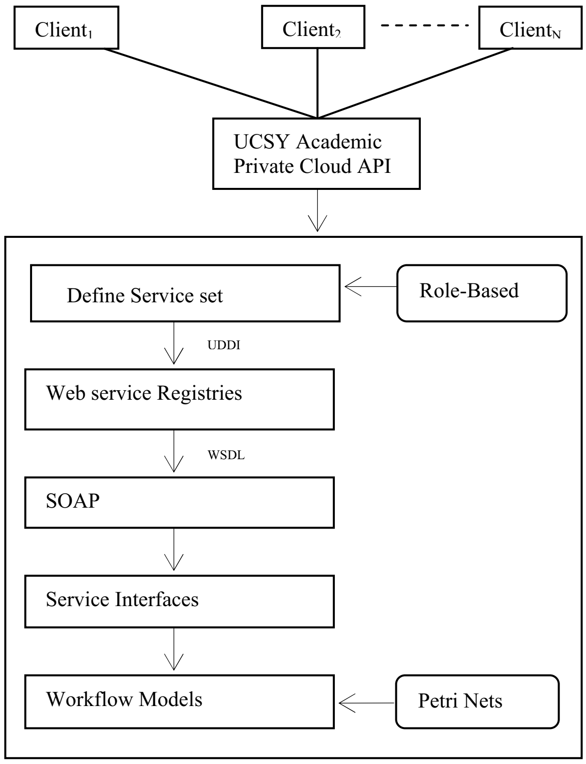

Figure 1 Proposed system architecture

Intra-cloud wise

1) Firstly, Classification of service according to the roles of users in user groups according to the Role-Based Access Control and Set Theory allows the rapid modeling, development and implementation of service integration scenarios.
2) Secondly, both Web Services Description Language (WSDL) and Service-Oriented Access Protocol (SOAP) may be presented in a single bundling scenario. There are additional problems that need to be addressed such as modeling and classifying of services according to roles of users in user groups and reliable integration of services that are required to integrate for possible scenarios. Service logic within the workflow can be tedious to keep the integration.
3) Thirdly, Petri Nets Model is used to get the performance and reliable of the service

integration.

Inter-cloud wise

The private cloud service will also be integrated with the public cloud service such as Google cloud (Google Mail, Google Talk, Google Calendar, Google Docs, and so on) and Amazon cloud (Amazon Web services – Elastic Compute Cloud (EC2) and Simple Storage Service (S3)). Google cloud service will be implemented using Google Application Engine.

4 Components of academic private cloud service architecture

4.1 User group

Let G is the set of user groups.
$G= \{\{ U_g \}, \{ U_{res} \}, \{ U_{gres} \}\}$ (OR)
G={{faculty}, {staff}, {student}, {researcher}}
In faculties user group, Professors, Associate Professors, Lecturers, Assistant Lecturers, Tutors.

In staff user group, Registration staff, Admin staff, Finance staff, Library staff. In student user group, under graduate student, post graduate student and research student. Since researcher can be members of both faculty and student groups, members in researchers may have access rights of both faculty and student groups.

4.2 Service set for user group

S_{pub} ={s | s is a public cloud services}
S_{priv} ={s | s is a private cloud services}
S={s | s is a public and private cloud service}

Services$_{\mathbf{Faculty}}$= {Report, Mark Entry, Define Exam Dates, Prepare Exam Question, Announce Conferences, Review Conference Papers, Internet Mail, Upload/ Download Service, Define Lesson Plans, Post Exercises, Post Assignments, Review Exercise, Review Assignments, Accept/ Deny Appointment }

Services$_{\mathbf{Student}}$= {File Management, Submit OwnFile, Post Exercises, Submit Assignments, Submit Conference Paper, Review Results, Get Academic Records, Run Exercise Program, Define Research Area, Define Requirements, Run Program, Appointment withSupervisor}

Services$_{\mathbf{Researcher}}$= {Define ResearchArea, Define Requirements, Run Research Program}

Services$_{\mathbf{Staff}}$ = {Manage Attendance (Students/ Staff), GiveWarning, Manage Promotion, Student Enrollment, Draw Time Table, Declare Exam dates, Enroll Books, Borrow Books, Timely Report, Calculate Student Fee, Compute Salary, Accounting Reports}

4.3 Eucalyptus and academic private cloud service architecture

This system will implement a Eucalyptus open source system with academic private cloud service architecture and provide the requested service by using the proposed workflows. The proposed academic private cloud architecture with Eucalyptus is shown in Figure 2.

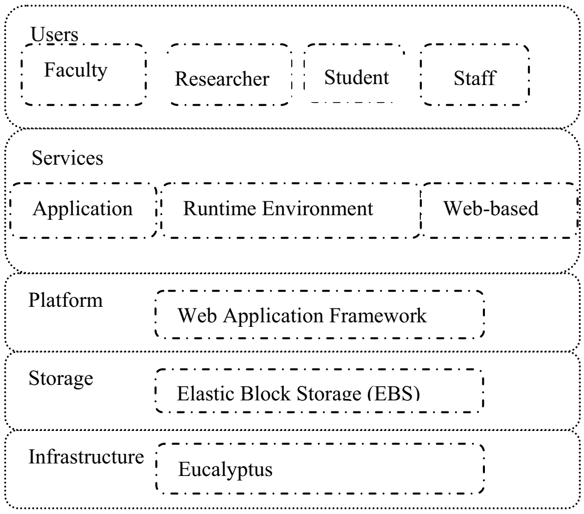

Figure 2 Eucalyptus and academic private cloud service architecture

5 Integration session

This section describes workflow, Petri Nets Model and workflow-based integration with Petri Nets.

5.1 workflow

A workflow is an executable business process. Workflow models support the administration, modeling, and execution of workflows. Before a workflow can be executed it has to be described in a description. This is called workflow specification. Many instances of the workflow are created according to the workflow specification during the run time of the system.

The main elements of a workflow specification are:

- activities/ services
- subjects/users
- data items
- data flow
- control flow

The basic building blocks of a workflow are the activities whose temporal and logical order is given by the control flow. To describe an activity, it has to be specified which users are allowed to execute an activity and which data items are needed for and created during the execution. Subjects can be associated with users, but also groups of users. In practice, the concept of role is very popular. The execution of an activity is bound to a specific role.

Examples of data items are database entries and files that are stored in Elastic Block Storage. A subject executes an activity by creating new and/or using already existing data items. The data flow states how the data items move between the different activities. The control flow will be given through a Petri net, the data flow and the assignment of subjects to activities through attributes of activities.

5.2 Petri nets model

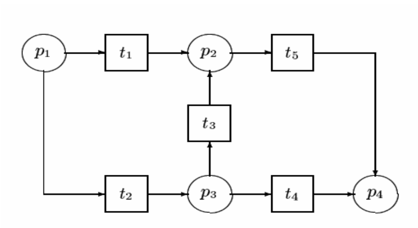

Figure 3 Example of a Petri nets

Definition 1(Petri Nets)

A Petri net N is a triple N = (P, T, F). P is the finite set of places, T is the finite set of transitions with $P \cap T = \varphi$. The flow relation F is defined by

$$F \subseteq (P \times T) \cup (T \times P)$$

Let y ∈ PUT. •y is called the preset of y and is defined by

$$\bullet y := \{x \in P \cup T : (x, y) \in F\}$$

y• is called the postset of y and is defined by

$$y \bullet := \{x \in P \cup T : (y, x) \in F\}$$

Figure 3 shows the example of Petri net. It consists of the places p1, ….., p4 and the transitions t1, …., t5. The set of P, T, and F are defined as follows:

P= {p1, ….., p4}

T={ t1, …., t5}

F={(p1,t1), (p1,t2), (t1,p2), (t2,p3), (p3,t3), (t3,p2), (p2,t5), (t5,p4), (p3,t4), (t4,p4)}.

The graphical interpretation of a Petri net is a bipartite graph. Places can only be connected to transitions; transitions can only be connected to places. Places are represented graphically as circles, transitions as rectangles. The graphical representation of an element (x, y) F is an arrow from x to y: the example (t1, p2) ∈ F. Therefore an arrow is connecting transition t1 with place p2. The preset and postset of a transition is a set of places. This set can be empty. The preset and postset of a place is a set of transitions. This set can be empty, too. Presets and postsets from the example: •t1= {p1}, t1•= {p2}, p3•= {t3, t4}, and •p1= φ.

Definition2 (Behavior of Petri Nets)

A non empty set M with $M \subseteq P$ is called a marking of a Petri net. A transition t is called activated under marking M, if

1. $\bullet t \subseteq M$ and
2. $M \cap t\bullet = \varphi$.

An activated transition can fire. If a transition t fires, M changes: $M \xrightarrow{t} M'$. The new marking M’ is defined by

$$M' := (M \setminus \bullet t) \cup t\bullet$$

The first marking of a Petri net is called start marking. Graphically a marking is represented by filled circles in all its places. These filled circles are called tokens.

The behavior of the Petri nets shall be illustrated the example in Figure 3. Let the start marking be {p1}. A token is put on place p1. The transition t1 and t2 are activated. If t2 fires the token will move from p1 to p3. The two transitions t3 and t4 are activated now. If t4 fires the token will move to p4. Now there are no more activated transitions. Formalized: $\{p1\} \xrightarrow{t2} \{p3\} \xrightarrow{t4} \{p4\}$.

5.3 Workflow-based integration with Petri Nets

This section describes the connection between workflows according to Petri nets described in Definition 1 and 2. A workflow specified by a Petri net will be called Petri net workflow. A Petri net workflow has the following characteristics:

1) Activities in workflows correspond to transitions in Petri nets. Executing an activity corresponds to the firing of a transition.
2) The marking of a Petri net represents the current state of a workflow.
3) The tokens are also called control tokens. The control tokens represent the state of the workflow, i.e. which activities are activated.
4) The flow relation says how the tokens can move in the net. The control flow of a workflow is determined by the flow relation and the start marking.

6 Possible scenarios for workflow-based integration

This system considers the following possible scenarios for service integration in academic institution based private cloud environment.

Scenario 1: Researchers have to get the **conferences announcement** for conferences and submit their research papers inside their academic private cloud.

Scenario 2: Students or researchers have to get the **virtual machines** with their needed software and applications anytime and anywhere.

Scenario 3: Student needs the **enrollment** (registration) for the new academic year.

7 Case study

This section expresses the workflow for enrollment of students in academic private cloud environment.

This scenario 3 has been chosen because a number of university departments such as registration department and finance department are needed to integrate their services. Figure 4 represents a workflow model of the activities involved the university departments such as registration department and finance department need to integrate their services. Figure 5 represents a Petri net workflow model of the activities involved in the enrollment.

7.1 Workflow for enrollment

1) If the student is a new student it has two conditions
   - ❖ First condition requests the entrance id and information of the student.
   - ❖ Second condition checks the student has been transferred from another university, checks the student's academic background.
2) If the student is an old student it will get into two conditions
   - ❖ First condition the student does the enrollment for the new class.
   - ❖ Second condition the student does the enrollment for the old class.
3) And then correct student associated with the request is officially enrolled so that fees have been paid.

7.2 Workflow with Petri Nets model

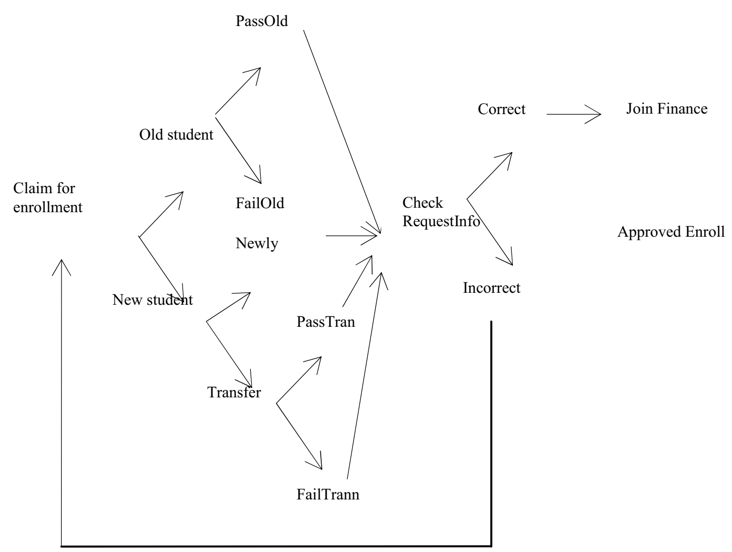

Figure 4 Workflow for Enrollment

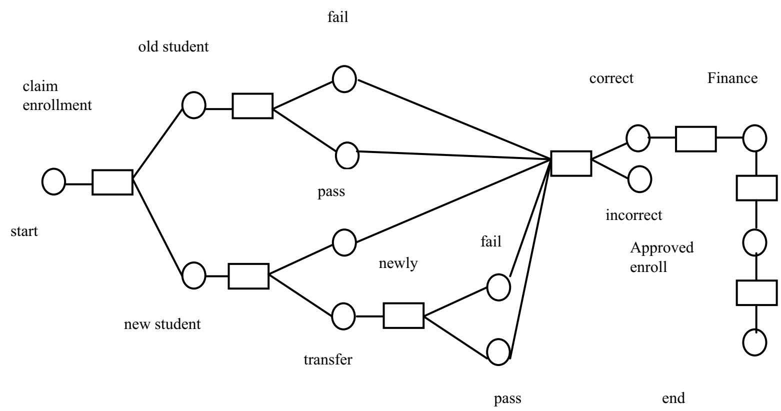

Figure 5 Workflow with Petri Nets for enrollment

8 Conclusions

In this paper, A Petri nets model has been proposed for workflow of academic private cloud environment. The main focus concerned is the mapping from workflow to Petri net based workflow model, which incorporates the control flow and data flow coordination, and an integrated workflow model can be obtained by integrating of services from private cloud and public cloud. In this approach, the workflow models in private cloud need not to change when integrating with the workflow models in other public cloud which is a promising way in workflow modeling for private cloud environment.

References

[1]C. Matthiew, Ken Laskey, Francis McCabe, Peter F Brown, and Rebekah Matz. “Reference Model for Service Oriented Architecture” 1.0. Technical report, OASIS, October 2006.

[2] V. vorgelegt. “Orchestrating Secure Workflows for Cloud and Grid Services”. 07.10.2010.

[3] S. Basu, S. Graupner, J. Pruyne and S. Singhal “Control Plane Integration for Cloud Services” in 11th International Middleware Conference, Bangalore, India Nov 29 –Dec 3, 2010.

[4] H. R. Motahari-Nezhad, J. Li, B. Stephenson, S.Graupner and S. Singhal “Solution Marketplace for Service Composition and Integration” in 3rd International Workshop on Web Service Composition and Adaptation, Los Angeles, USA July 2009.

[5] H. Hai and S.Sakoda “SaaS and Integration Best Practices in January 23, 2009.

[6] R. G. Siebeck, T. Janner and C. Schroth. “Cloud-based Enterprise Mashup Integration Services for B2B Scenarios” in 2009.

[7] Jia Yu and Rajkumar Buyya. “A Taxonomy of Workflow Management Systems for Grid Computing”, http://www.gridbus.org

[8] **T.** Longji, D. Jing, Z. Yajing and Z. Liang-Jie “Enterprise Cloud Service Architecture” in 2010.