Integration of Services for Academic and Research on private cloud system

Khine Khine Htwe
University of Computer Studies, Yangon
khinehtwe@gmail.com

Abstract

Service composition and integration are well investigated problems in Service Oriented Architecture (SOA). However, they still remain among the hard SOA challenges for which automated approaches have yet to be developed. These issues are hindering the agile and cost-effective development of service-based business solutions, and the need for addressing them becomes more pressing with the increase in the number of online (cloud) services. The modern design of business and academic IT landscapes is based upon service-oriented architectures. The integration of (external) services raises various challenges to combine the services. In this system workflows are the preferred means for the integration of services on private cloud systems. The proposed workflow model allows academic users to deploy and run the services on (external) private cloud providers. As a Test-bed environment this system uses a Eucalyptus open source system to build up academic institution-based private cloud system.

1. Introduction

The cooperation of departments, enterprises and academic institutes is an important factor for successful work. Such cooperation yields in significant effects since selected services may be outsourced to partners. Modern technologies such as service-oriented architectures (SOA) [1] allow the forming of different IT systems to a coherent IT landscape. Small business functions are encapsulated in loosely-coupled services which are combined to executable workflow. They allow a flexible integration of enterprise IT services. The will increase interoperability between the different domains as well as between the services inside one domain and ease switching between service providers.

Software as a Service (SaaS) is defined in [2] 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.

The problem is to integrate the services on different private cloud system. The solution depends on user groups and requirements. The contributions of this paper are as follows:

- This system intends to provide the services that are existed on the two cloud providers according to the user request based on the workflow of this system.
- It also wants to reduce the complexity of execution of services which services are needed to integrate.
- Finally the goal of the system is to minimize the execution time of the services by using workflow-based integration.

The rest of this paper is organized as follows. Section 2 describes the related work. Academic Cloud Service Architecture is explained detail in section 3. Target Users for Private Cloud System is described in section 4 as well as Services for Private Cloud System is also explained in section 5. Background of Workflow and proposed idea are presented in section 6 and 7. Finally section 8 concludes the paper.

2. Related Work

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 service solutions 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. 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 [1]. 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. Academic Cloud Service Architecture

According to system implementation, there are two private clouds providers involved such as general private cloud and research-oriented private cloud. The following definitions are presented in this paper for ACSA:

AcademicCloud = {general private cloud }U{research-oriented private cloud }

ACSE=<U, S, CI, CM, CD, CP>
where
$U=U^{I} \cup U^{II} \cup U^{III}$
$U^{I}=$ {u | u is a general cloud service users}
$U^{II}=$ {u | u is a research cloud service users}
$U^{III}=$ {u | u is a general and research cloud service users}
where $U^{III} = U^{I} \cup U^{II}$

$S=S^{I} \cup S^{II} \cup S^{III}$
$S^{I} =$ {s | s is a general cloud services}
$S^{II} =$ {s | s is a research cloud services}
$S^{III}=$ {s | s is a general and research cloud service}
where $S^{III} = S^{I} \cup S^{II}$

$S^{I}=$ {{SaaS}, {DaaS}, {PaaS}, {IaaS}}
$S^{II}=$ {{SaaS}, {PaaS}, {IaaS}}

CI={ci | ci is a cloud infrastructure }
where cloud infrastructure is a dynamic IT infrastructure.

CM={cm | cm is a cloud management}
where network and application monitoring, identity Management, policy enforcement, service-level agreement management, and service lifecycle management.

CD={cd | cd is data and metadata}
where data elements which are used for building ACSA style enterprise architecture.

CP={cp |cp is process or workflow}
Each process is composed by multiple services in integration (orchestration and/or choreography)

for completing a whole or partial business process or task.

3.1. Specification of Academic Cloud Service

(1) Cloud Service Interface Type
I_{type} ={User Interaction Interface, Web Service Interface, REST Interface, Web Application Interface, Event Interfaces}

(2) Cloud Service Access Type
$A_{\mathbf{type}}$ ={a | a is a client access protocol method}, such as Web User Interaction (HTTP), Web Service API (SOAP), REST API (HTTP), Web Application API.

(3) Cloud Service Provisioning Type
$P_{\mathbf{type}}$ ={Applications, Resources, Data, Platform}

(4) Cloud Service Control/Ownership Type
$O_{\mathbf{type}} = \{ O_{general}, O_{research} \}$

ACS= $\langle I_{ACS}, A_{ACS}, P_{ACS}, O_{ACS}, SLA_{ACS} \rangle$
where $I_{ACS} \subset I_{type}$, $A_{ACS} \subset \mathbf{A}_{type}$, $P_{ACS} \subset P_{type}$, $O_{ACS} \subset \mathbf{O}_{type}$, SLA_{ACS} is the service level agreement between providers and users.

3.2 EUCALYPTUS Design

Eucalyptus is open source and is developed for academic purpose, allowing researchers to build their own cloud infrastructure through compatible Amazon EC2 interfaces. The architecture of the EUCALYPTUS system is simple, flexible and modular with a hierarchical design reflecting common resource environments found in many academic settings.

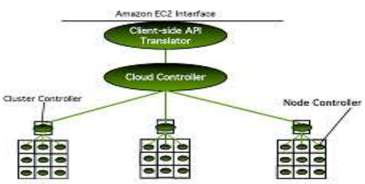

Figure 1: Eucalyptus Design

The terms of the eucalyptus components are as follows:

Node Controller controls the execution, inspection, and terminating of VM instances on the host where it runs.

Cluster Controller gathers information about and schedules VM execution on specific node controllers, as well as manages virtual instance network.

Storage Controller (Walrus) put/get storage service that implements Amazon's S3 interface, providing a mechanism for storing and accessing virtual machine images and user data.

Cloud Controller is the entry-point into the cloud for users and administrators. It queries node managers for information about resources, makes high-level scheduling decisions, and implements them by making requests to cluster controllers.

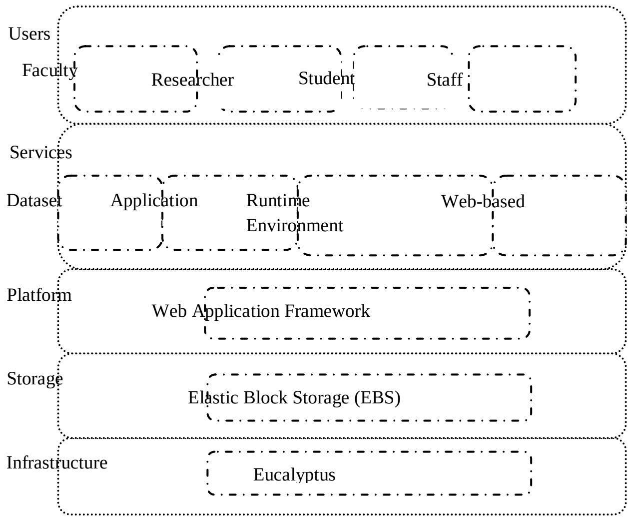

Figure 2: Eucalyptus and Academic Cloud Service Architecture

4. Target Users for Private Cloud System

Let G is the set of user groups.
G= {{g1}, {g2}, …………, {gn}}
G_{user}={{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.

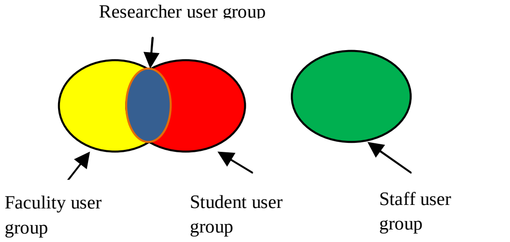

Figure 3: Academic institution-based private cloud user group architecture

5. Services on Private Cloud System

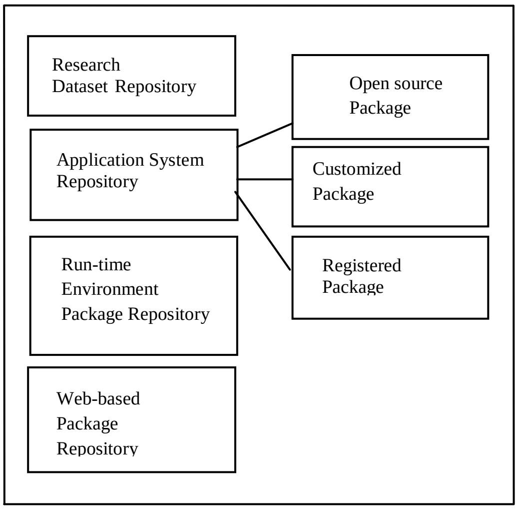

Figure 4: Services on Private Clouds

5.1 Services and Service Level Agreement

In this system, some services are shared among user groups but most of services are dedicated for specific user group. Even in the same group, some of the services are shared and some services are private service. Service usage is defined on each register user.

5.2 Defining Essential Data objects in Private Cloud System

There are many data objects in academic institution-based private cloud system. Among them essential objects are as follows:

- Subjects and Lessons
- Departments Information
- Exam and Tutorials dates
- Student information
- Academic records
- Research Program and Master Program and so on.

5.3 Defining General Services and Integration Services

Service Integration process will be performed based on following:

- Intra-cloud wise (service integration inside one cloud)
- Inter-cloud wise (service integration between different clouds)

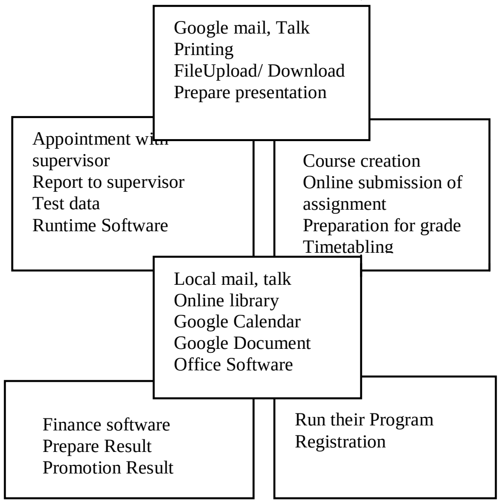

Figure 5: Block Diagram for Services group for Users

For example, in the case of integration service where a researcher gets simulation data from data service and uses it in his / her research program in the another cloud as data service, it is the inter-cloud wise service integration. 'Staff' user group computes grading and prepare result sheets from the academic record of data center from same private cloud. Such case is an example of intra-cloud wise service integration. And also Mail service (general service) is available to all members. Internet mailing is granted based on member's role and his / her access right permission.

6. Background of Workflow

A workflow is composed by connecting multiple tasks according to their dependencies. The workflow structure, also referred as workflow pattern [9][10], indicates the temporal relationship between these tasks. In general, a workflow can be represented as a Directed Acyclic Graph (DAG) [11] or a non-DAG.

In DAG-based workflow, workflow structure can be classified as sequence, parallelism, and choice. Sequence is defined as an ordered series of tasks, with one task starting after a previous task has completed. Parallelism represents tasks which are performed concurrently. In choice control pattern, a task is selected to execute at run-time when its associated conditions are true.

In addition to all patterns contained in a DAG-based workflow, a non-DAG workflow also includes the iteration structure in which sections of workflow tasks in an iteration block are allowed to be repeated. Iteration is also known as loop or cycle. These four types of workflow structure, namely sequence, parallelism, choice and iteration, can be used to construct many complex workflows. Moreover, sub-workflows can also use these types of workflow structure as building blocks to form a large-scale workflow Workflow Model (also called workflow specification) defines a workflow including its task definition and structure definition. Two types of workflow models are namely abstract and concrete.

In an abstract model, a workflow is described in an abstract. An abstract model provides a flexible way for users to define workflows without being concerned about low-level implementation details. Tasks in an abstract model are portable and can be mapped onto any suitable services at run-time by using suitable discovery and mapping mechanisms.

In contrast, a concrete model binds workflow tasks to specific resources. In some cases, a concrete model may include tasks acting as data movement to transfer data in and out of the computation . In other situations, tasks in a concrete model may also include necessary application movement to transfer computational code to a data site for large scale data analysis.

7. Proposed System

This section describes the proposed system to offer workflow of integration services. A user requests a certain given cloud service, we apply a workflow to describe and manage the cloud service. And then finally the result is returned to the users.

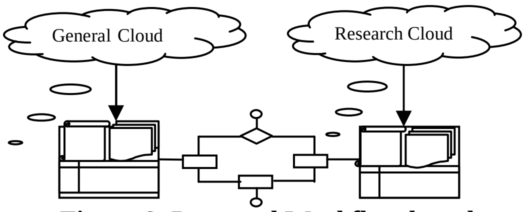

Figure 6: Proposed Workflow-based Integration Design

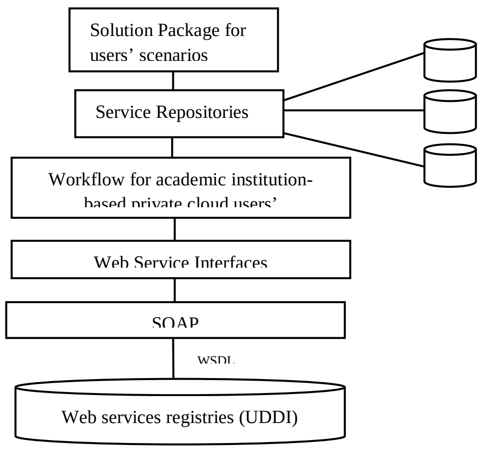

Figure 7: Overview of the Proposed System

Proposed system is considered on these following scenarios:

Scenario 1: Researchers need to integrate services such as report for project that are in the research-oriented private cloud and office software that are in the general private cloud and email services that are in the research-oriented private cloud.

Scenario 2: Student needs the enrollment (registration) for the new academic year.

Scenario 3: Researchers run their research program that are in the research-oriented private cloud and also needed input test data that are in the general private cloud.

We denote an application workflow as a **Directed Acyclic Graph** (DAG) represented by =(V,E), where V ={T1, ..., Tn} is the set of tasks, and E represents the data dependencies between these tasks, that is, fj,k =(Tj ,Tk) ∈ E is the data produced by Tj and consumed by Tk.

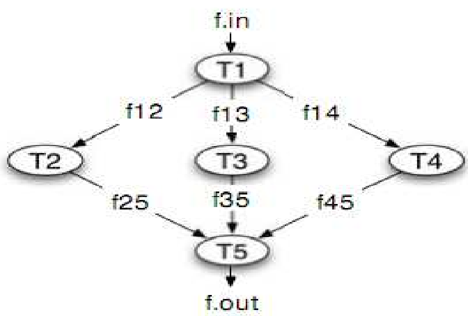

Figure 8: Example Workflow for Scenario 2

An example workflow **for Scenario 2** we consider the process of the enrolment of prospective students at our university. This example has been chosen, because a number of university groups (i.e admission office, academic departments, and financial office) must collaborate during this process, and the part of the process that each group is responsible for can be thought of and modeled as a sub-process.

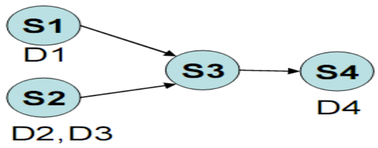

Figure 9: Example Workflow for Scenario 3

An example workflow **for Scenario 3** we consider the process of running the program for respective research user. This example has been chosen, because research user program and their test data need to integrate to produce their result.

8. Conclusion

This system **presents workflows** that allow for **integration of services** from different private cloud provider. Researchers are able to share, extend and even execute workflows and share data sets with fellow researchers and developers from across domains of interest. This system implements the services integration system to support the availability and scalability of SOA of the academic institution-based private cloud system as well as provides the services by doing the workflow-based integration of application services.

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 November 29 -December 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.

[9] W.M.P. van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A. P. Barros. "Workflow Patterns". Technical Report, Eindhoven University of Technology, 2000.

[10] W.M.P. van der Aalst, A.H.M ter Hofstede, B. Kiepuszewski, and A. P. Barros. "Advanced Workflow Patterns". In CoopIS 2000, Lecture Notes in Computer Science (LNCS) 1901, Springer-Verlag, Heidelberg, Germany, 2000; 18-29.

[11] R. Sakellariou and H. Zhao. "A Low-Cost Rescheduling Policy for Efficient Mapping of Workflows on Grid Systems". Scientific Programming, 12(4):253-262, IOS Press, Netherlands, December 2004.