

Provision of U-city Web Services Using Cloud Computing

SEUNG WOO RHO, CHANG HO YUN, YONG WOO LEE (Corresponding Author)

Department of Electrical and Computer Engineering, The University of Seoul, Seoul, Korea

{dairen, touch011, ywlee}@uos.ac.kr

Abstract—U-city is a future city with high-tech and provides citizens many kinds of U-city services. Because necessary computing resources for U-city services are different, cloud computing that assigns and manages such resources dynamically must be supported in the U-city. As well, U-city portal must support one stop services for the execution of automatic operation to enable users to use U-city services on the web easily. To meet these requirements, we made a U-city portal that integrated various kinds of web services for cloud computing in the U-city altogether. We developed the Virtual Machine Job Manager(VMJM) of the U-city portal for one stop services on the cloud platform, which uses the OpenNebula virtual infrastructure and the Haizea as virtual machine resource management scheduler. It enables general users to execute and monitor U-city services on the virtual machines automatically without experts' knowledge. That is, in this paper, we present web services for cloud computing in the U-city.

Keywords—Web Service, U-city, U-city Middleware, U-city Portal, Cloud Computing, OpenNebula, Haizea

I. INTRODUCTION

The U-city is "a city where U-city Services are available without regard to time and location and which is realized by the application of U-city technologies in order to enhance the quality of lives of citizens within and competitiveness of the city itself." [1]

Particularly, the U-city controls complex processing for operation of services and provides users with many services regardless of hardware, operation system, and network existing on the substructure of system with the U-city middleware [2]. Also, the U-city should offer user web interface user-friendly to enable them to conveniently make use of services provided by the middleware at online through the U-city portal [3] which is one of special portals. To meet these requirements, U-city should have a U-city middleware. The followings are typical

characteristics of the smart U-city middleware, which was developed in our U-city project. First, it provides common device interface for various kinds of sensors and ubiquitous sensor networks. Second, it provides context-aware, that is, intelligent information using the sensed data through the common device interface. Third, it supports user-transparent infrastructure that generates and provides intelligent services, which are invisible to users to various applications. That is, it can reduce the period and expense to develop the U-city applications.

According to user request and complexity of U-city services, each computing resource and environment that they need are not equal. In order to meet these various requirements, it takes much time. Hence, we use cloud computing in the U-city to assign and manage such resources dynamically and easily. Cloud computing can conveniently support many U-city service platforms without change of real physical machines. We also can use the external computing resource with cloud computing when computing resource for executing the complex U-city web service is insufficient. However, most of existing U-cities don't use U-city middleware as well as cloud computing because they only provide basic services like transmission of sensed information.

For this purpose, we apply cloud computing [4] to U-city middleware to operate and manage U-city effectively. In order to inter-operate cloud infrastructure with U-city middleware, we utilize OpenNebula [5] which creates and manages virtual machines, and Haizea [6] which allows us to assign virtual machines on the appropriate physical node

dynamically. We present Virtual Machine Job Manager(VMJM) that a part of cloud manager of U-city portal to enable us to simply execute and monitor services on the cloud platform. We can make use of one stop services for execution of automatic operation with VMJM.

The outline of this paper is organized as follows: In section 2, we compare several web services supported by cloud infrastructures. Section 3 explains architecture of U-city cloud computing platform and cloud manager for providing web service for cloud computing. Section 4 introduces how to implement VMJM and principle of its operation. Section 5 describes scenario that user executes U-city web services for cloud computing. Finally, section 6 gives conclusions and explains future works.

II. RELATED WORK

Currently, U-cities don't use cloud computing. They provide user with only simple web services. Also, there does not exist related works that provide U-city web services for cloud computing. In this section, we shortly look around several kinds of web services for cloud infrastructure.

Eucalyptus(Elastic Utility Computing Architecture for Linking Your Programs to Useful Systems)[7] is one of popular open source packages used to build cloud computing infrastructure. Distinctive thing of this package is to be compatible with Amazon EC2(Amazon Elastic Compute Cloud)[8] which is cloud computing interface of Amazon. However, web services of EC2 are focused on cloud infra service and do not offer web service executable on the virtual machine. In addition, it is complex to create virtual machine and has an intricate setting process. Elastic Map-Reduce[9] commercially supported by Amazon just provides Map-Reduce based web services.

Nimbus[10] provides basic web services that can manage cloud infra and one-click service that can build cluster automatically on the Nimbus web. However, although Nimbus offers users web service which can manage job on the virtual machine with globus[11], they should install parts of globus and additional client program on the client PC.

OpenNebula[12] is another interesting open source application developed by Universidad Complutense de Madrid(Under the Apache License). OpenNebula supports idea called Hybrid cloud as well as private cloud structure. Hybrid cloud could more enhance extension level using combination of both private cloud and public cloud. Also, OpenNebula supports Xen[13], KVM/Linux[14] and VMware[15] and performs management and inspection work with some factor like libvirt.

OpenNebula Management Console[16] is a console providing web services that can manage OpenNebula infrastructure. It was recently developed directly by SARA Computing & Services Center and is continuously being in progress as open-source by various participants. Although it provides web services for virtual machine management easier than other cloud infrastructures, OpenNebula Management Console does not provide web services based on the virtual machine as well.

Web services mentioned above are focused on cloud infra service and need related knowledge to use these. On the contrary to this, we offer web services that general users can create virtual machines automatically without knowledge concerning cloud infra, so they execute and monitor U-city services on the virtual machines. In the next section, we explain our system architecture in detail.

III. ARCHITECTURE

A. Cloud Computing Platform in the U-city middleware

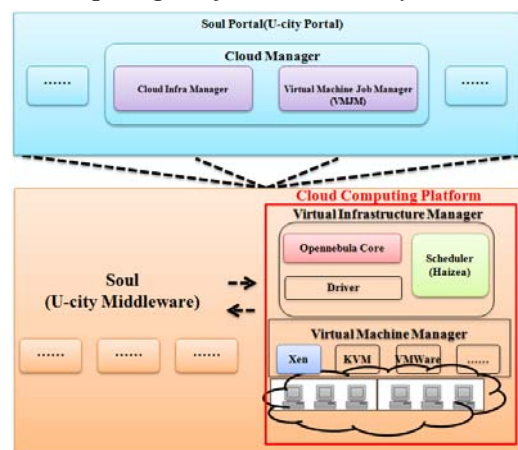


Figure 1. Cloud computing platform in the U-city middleware

Figure 1 shows the cloud computing platform in the U-city middleware. Cloud computing platform is largely composed of four parts: Physical node, Virtual Machine Manager, Virtual Infrastructure Manager, Cloud User Interface.

Virtual machine manager such as Xen, KVM, VMware, etc. supplies virtual environment where they create virtual machine image in that virtual infrastructure needs actually. We used combination of OpenNebula and Haizea to organize virtual infrastructure. This combination allows users to assign virtual machines on the desirable physical node when they create VMs, to lease VMs, and to reserve VMs in advance. OpenNebula supports virtualization, network, storage, and external cloud-related driver. Besides, it integrates and monitors information of many physical nodes and virtual nodes periodically.

Because the image created by virtual machine manager has pure operation system, it must be set-upped suitably for services in the U-city middleware. There are two kinds of method to achieve it. First is to set necessary programs on the image created in the virtual machine manager after connecting directly through the console. Second is that requisite programs are installed on the virtual machines automatically after administrator adds context line in the settings file of the OpenNebula.

The first method takes more time than the second for setting at initial. However, once the first method is set-upped, because it copies pre-installed image when virtual machines are created, creation time of virtual machine is more shorten. Accordingly, we used images appropriately set-upped for each U-city service in advance.

U-city portal includes cloud manager that provides web services executable on the cloud infra. Details of this explain in the following section.

B. Cloud Manager

Cloud Manager consists of Cloud Infra Manager and Virtual Machine Job Manager. Cloud Infra Manager supports web services that administrator can create and manage virtual machine on the U-city cloud platform. Virtual Machine Job Manager supports web services that general users can execute and monitor U-city services on the virtual machine automatically.

1) Cloud Infra Manager: Cloud Infra Manager basically supports web services that manage virtual machines, physical node, image template, virtual machine network and virtual machine users. OpenNebula provides users with command user interface fundamentally, but also XML-RPC API[17] for communication with web interface. OpenNebula Management Console was implemented with XML-RPC API and PHP which is web application language. We used it to offer U-city cloud infra management web services for administrator.

2) Virtual Machine Job Manager: Virtual Machine Job Manager (VMJM) supports U-city service submission web service that users can execute U-city services for cloud computing on the web and U-city service monitoring web service that they can monitor status of submitted jobs. Users can construct the cluster using one stop service on the VMJM. This automatic process is composed of selection of virtual machine template, creation of virtual machine, network setting, etc. Also, they can execute U-city service on the cluster. This one stop service is combination of the creation of cluster and the service execution.

IV. IMPLEMENTATION : VIRTUAL MACHINE JOB MANAGER

We used PHP 5.3.3 and Apache Httpd 2.2.17 on the Linux, ubuntu 9.04 to implement Virtual Machine Job Manager easily and quickly and CodeIgnitor[18] that is the fastest among existing PHP MVC framework. The VMJM has two controllers (U-city service submission and monitoring), which are a U-city model and a U-city view. In the U-city model and view, there are two service model and view. As using MVC framework, maintenance and extension of code were easier, and reusability increased. The Flow of overall system is as follows:

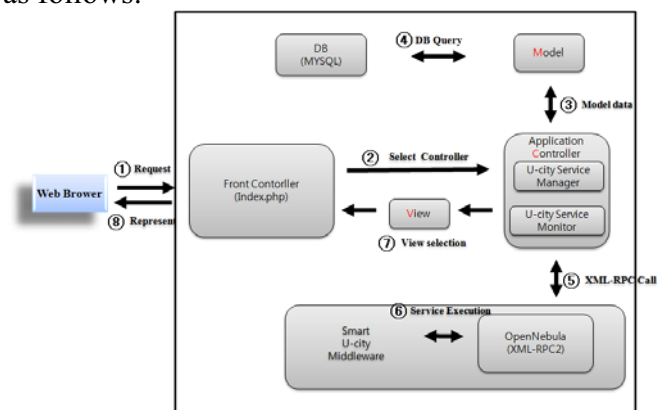


Figure 2. MVC flow chart of Virtual Machine Job Manager

The Figure 2 shows each sequential flow of VMJM. First of all, after front controller receives user request, it transfers the request to application controller. Application controller calls

corresponding web service controller by the request and loads appropriate U-city model and other resources by controller logic. Finally, view module of matching web service is rendering, and then it is sent to web browser.

The figure 3 shows hardware configuration of the U-city cloud computing platform. We used 15 nodes, which consist of a master node and fourteen sub-nodes.

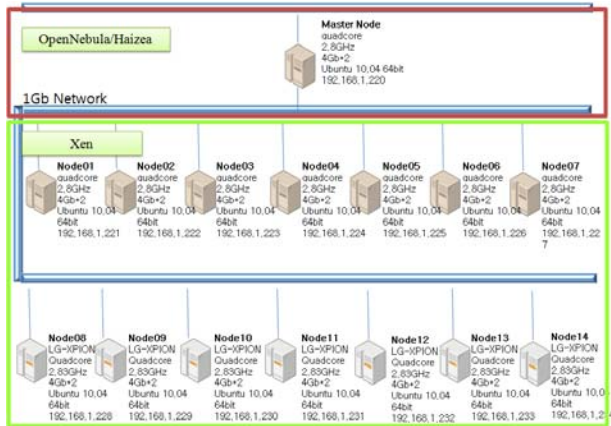


Figure 3. Hardware Configuration of U-city cloud computing platform

VMJM consists of two web services. One is the U-city service submission, and another is the U-city service monitoring.

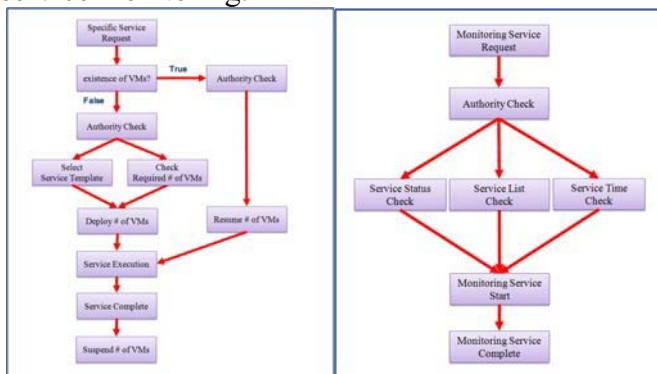


Figure 4. Sequence diagram of U-city service submission and monitoring

A. U-city Service Submission Web Service

The U-city service submission web service is a service that users who have execute permissions or administrator can execute some U-city services. The left of the figure 4 shows sequence diagram of U-city service submission.

First, if user selects U-city service on the cloud manager, system checks pre-created VMs, and then if they exist, after checking authority, VMs is resumed using one.vm.action API and 'resume'

parameter. Otherwise, after checking authority, system calls Deploy_VM() function that is a internal function of system, and deploys necessary virtual machines. Here, Deploy_VM function uses one.vm.allocate API supported by the OpenNebula. The one.vm.allocate API is used by rpc_request function, and this function requires template name and auth-session parameter. Finally, it executes the service. After finishing the service, the VMs are suspended by the one.vm.action API and 'suspend' parameter. The all course are processed automatically. At this time, requirements for selecting appropriate physical node to create virtual machine are pre-defined in the corresponding service template.

B. U-city Service Monitoring Web Service

The U-city service monitoring web service is a service that users can check status and time of running services. The right of the figure 4 shows the sequence diagram of service monitoring web service. This is very simple. If user requests service inquiry, system calls Monitor_Service() function by the request, and it represents information concerning list, status and time of VM-based services on the screen.

V. USE-CASE OF U-CITY WEB SERVICE : 3D GIS VISUALIZATION OF ENVIRONMENT INFORMATION

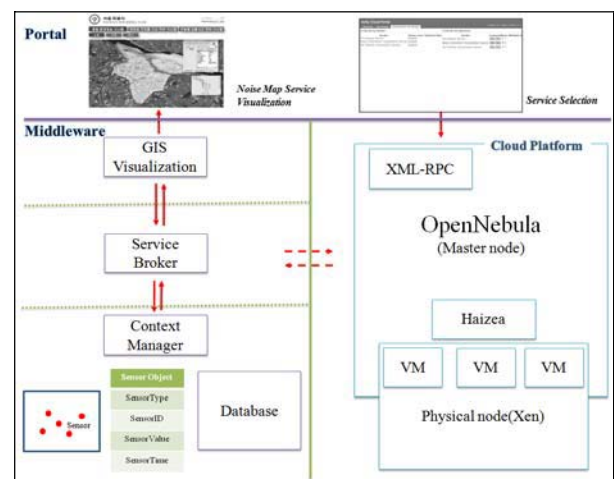


Figure 5. Scenario example of 3D GIS visualization U-city service

3D GIS visualization of environment information with cloud computing is one of many web services provided in the U-city. The figure 5 shows the use-case of U-city web service, which is 3D GIS visualization of environment information. This use-

case process is as follows. A citizen of the U-city requests 3D GIS visualization service of noise information of "A" zone on the web. In here, we assume that the person has no any knowledge about cloud infra or the U-city middleware, and only has authority that can request or stop services. Many ubiquitous sensor networks are connected to collect environment information such as air, noise, etc in the corresponding "A" zone. Information collected from sensors is processed by the U-city middleware. Noise information has three values(X, Y, Z) to display 3D and it is processed with distributed parallel computing to offer him result in a real time. The U-city middleware checks available resources in the U-city cloud system and optimal quantity of virtual machines suitable to visualize noise information, and then it constructs the cluster. If available resources are insufficient in the U-city cloud system, the U-city middleware leases external cloud resources. Lastly, after this service is executed on the cluster, user can see result of this service on the web.

VI. CONCLUSIONS

We explained provision of U-city web services using cloud computing in this paper. At first, we established cloud computing platform to work together with U-city middleware and examined several web services for cloud infra management. Also, we introduced one stop services for automatic process that user can simply execute and monitor U-city services without knowledge of cloud infra on the web. In the future work, we will continue to support more and more U-city web service to all users.

ACKNOWLEDGMENT

This study was supported by the Seoul Research and Business Development Program(10561), Smart(Ubiquitous) City Consortium and Seoul Grid Center. We would like to give thanks to Miss Hae Sun Jung, Mr. Jong-Won Park, Mr. Hyun-Kyu Park, Mr. Cheol Sang Yoon, Mr. Chang-Won Lee and the staffs of Seoul Grid Center and the members of Smart(Ubiquitous) City Consortium for their contribution to this research.

REFERENCES

- [1] Ministry of Land, Transport and Maritime Affairs, Decree about building a ubiquitous city, This Decree enter into force on June 9, 2009, 9770.
- [2] H.S. Jung, et al., "An Intelligent Ubiquitous Middleware for U-city: SmartUM," *Journal of Information Science and Engineering*, vol. 25, no. 2, 2009, pp.375-388.
- [3] Seung Woo RHO, Yong Woo LEE, "U-city Portal For Smart Ubiquitous Middleware", 2010 The 12th International Conference Advanced Communication Technology(ICAICT), 2010. 2, pp.609-613
- [4] A white paper produced by the Cloud Computing Use Case Discussion Group, "Cloud Computing Use Cases", Version 4.0 2 July 2010
- [5] (2010) OpenNebula website.[Online]. Available: <http://opennebula.org/>
- [6] (2010) Haizea, An Open Source VM-based Lease Manager website.[Online]. Available: <http://haizea.cs.uchicago.edu/>
- [7] Daniel Nurmi, et al, "The Eucalyptus Open-source Cloud-computing System", 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, 2009, pp.124-131
- [8] (2010) Amazon EC2 website.[Online]. Available:<http://aws.amazon.com/ec2/>
- [9] (2010)Elastic MapReduce website.[Online]. Available:<http://aws.amazon.com/elasticmapreduce/>
- [10] (2010)Nimbus website.[Online]. Available:<http://www.nimbusproject.org/>
- [11] (2010) globus website.[Online]. Available:<http://www.globus.org/>
- [12] Sotomayor B, Montero R.S, Lioriente I.M, Foster I, "Virtual infrastructure management in private and hybrid clouds", *IEEE Internet Computing*, Vomlume 13, Issue 5, 2009, pp.14-22
- [13] (2010)Xen hypervisor website.[Online]. Available: <http://www.xen.org>
- [14] (2010) Linux/KVM website.[Online]. Available: http://www.linux-kvm.org/page/Main_Page
- [15] (2010) VMware website.[Online]. Available: <http://www.vmware.com/>
- [16] (2010) OpenNebula Management Console website.[Online]. Available: <http://dev.opennebula.org/projects/management-console>
- [17] (2010) OpenNebula XML-RPC API website.[Online]. Available: http://www.opennebula.org/documentation:archives:rel1.2:api_examples
- [18] David upton, "CodeIgniter for Rapid PHP Application Development: Improve your PHP coding productivity with the free compact open-source MVC CodeIgniter framework", 2009
- [19] (2010) UML website.[Online]. Available: http://en.wikipedia.org/wiki/Unified_Modeling_Language