International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:6, No:9, 2012

Design of Multiple Clouds based Global Performance Evaluation Service Broker System

Dong-Jae Kang, Nam-Woo Kim, Duk-Joo Son, Sung-In Jung

***Abstract*—**According to dramatic growth of internet services, an easy and prompt service deployment has been important for internet service providers to successfully maintain time-to-market. Before global service deployment, they have to pay the big cost for service evaluation to make a decision of the proper system location, system scale, service delay and so on. But, intra-Lab evaluation tends to have big gaps in the measured data compared with the realistic situation, because it is very difficult to accurately expect the local service environment, network congestion, service delay, network bandwidth and other factors. Therefore, to resolve or ease the upper problems, we propose multiple cloud based GPES Broker system and use case that helps internet service providers to alleviate the above problems in beta release phase and to make a prompt decision for their service launching. By supporting more realistic and reliable evaluation information, the proposed GPES Broker system saves the service release cost and enables internet service provider to make a prompt decision about their service launching to various remote regions.

***Keywords*—**GPES Broker system, Cloud Service Broker, Multiple Cloud, Global performance evaluation service (GPES), Service provisioning

I. Introduction

CLOUD computing allows cloud consumer to conveniently rent access to fully featured applications, to software development and deployment environment, and to computing infrastructure assets such as network-accessible data storage and processing. In IaaS case, a high level of compatibility can be maintained between legacy application and workloads in an IaaS cloud, because IaaS clouds allow cloud consumers to install and run operating systems of their choosing [1].

As the issues of multiple or federated cloud usefulness evolves, the integration of cloud services can be too complex for cloud consumers to manage. To ease this difficulty, a cloud consumer may request cloud service from a cloud service broker, instead of contacting a cloud provider directly.

Dong-Jae Kang is with the Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Korea and with Department of Computer Software Engineering, University of Science & Technology, Daejeon, Korea(phone: 82-42-860-1561; fax: 82-42-860-6699; e-mail: djkang@ etri.re.kr).

Nam-Woo Kim is with the Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Korea (e-mail: hellowkorea@etri.re.kr).

Duk-Joo Son is with the Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Korea (e-mail: djson@etri.re.kr).

Sung-In Jung is with the Electronics and Telecommunications Research Institute, 218 Gajeong-ro, Yuseong-gu, Daejeon, 305-700, Korea (e-mail: sijung@etri.re.kr).

A cloud broker is an entity that manages the use, performance and delivery of cloud services and negotiates relationship between cloud service providers and cloud service consumers [2].

Cloud service broker is the role of indirect interconnection between two or more cloud service providers achieved through an interconnecting cloud service provider. Brokering service functions generally include, but are not limited to, service intermediation, service aggregation and service arbitrage [3].

Recently, internet services, such as game and SNS, tend to dramatically grow. With this trend, an easy and prompt service deployment has been important for internet service providers to successfully maintain time-to-market.

In case of global service deployment, they have to pay the cost for repetitive SW installation, system configuration and visiting to service target area to make a decision of the proper system location, system architecture, system scale and so on.

Generally, before launching a internet service, the company performs several service evaluation in Lab environment that is simulation tests including functional test, speed and performance test, usability test for the service. But it tends to have big gaps in the evaluated data between the real and simulated environment, because it is very difficult to accurately expect the local service environment, local network situation, service delay, network bandwidth and other SLA related factors [4].

This situation affects to next service release phase, open and close beta testing that will be performed by limited real user groups, and it makes internet service providers to pay big cost for re-configuration, modification and tuning of the service systems. Consequently, service launching time will be delayed, and it also causes time-to-market to be delayed.

Therefore, to resolve or ease the upper problems, we propose the cloud based GPES Broker system and its use case that helps internet service providers to alleviate the above problems. GPES Broker system supports fast provisioning of resource infrastructures needed in service evaluation, system and computational resources, over the multiple or federated clouds and provides automated and integrated evaluation environment including monitoring of various performance factors and process about the evaluated data for the GPES consumers.

In detail, the proposed GPES Broker system uses multiple or federated clouds and provides several useful functionalities, prompt deployment to target location against multiple cloud providers, service and VM lifecycle management, automated service evaluation methods.

International Journal of Business, Human and Social Sciences
ISSN: 2517-9411
Vol:6, No:9, 2012

By providing more realistic and reliable evaluation information, the proposed GPES Broker system saves the service release cost and enables internet service provider to make a prompt decision about their service launching who want to launch their service to another regions.

II. Related Works

As the issues of cloud federation and multiple cloud usefulness evolves, the integration of cloud services and cloud federation can be too complex and difficult for cloud consumers to manage and use. To alleviate this situation, many research groups have been studied about related issues, such as reference architecture[2], definition and role of cloud broke [2], [3], [5], [6], business model [7], [8] and related standard [3], [9], [10], [11]. The several platforms for cloud federation or cloud brokering functionality have been researched. In early 2008, OpenNebula [12], enhanced in the RESERVOIR [13] European Commission-funded project, became the first open-source software for deploying private and hybrid clouds for the federation of clouds. In 2010, SpotCloud [14] launched a Cloud capacity marketplace which can be viewed as a the first example of commercial Cloud Federation and SlapOS [15] demonstrated a new approach to Cloud Federation: non cooperative federation. By registering virtual machines and bare metal servers into a common resource pool called SlapOS Master, capacity from different Cloud vendors can be aggregated as it were a single vendor. SlapOS Master thus acts as a gateway between different Cloud providers without requiring any cooperation between them. Accords, the OCCI broker developed as part of the CompatibleOne project [16], was demonstrated in 2011 and showed the possibility to aggregate cloud capacity from different cooperating Cloud vendors by matching their APIs into OCCI. It relies on a description of virtual machine images and virtual hosts based on the OCCI standard. In enterprise section, several considerable implementations have been developed to serve as intermediaries between end users and cloud providers. CloudKick launched in March 2009 as part of the winter class at start-up incubator Y Combinator, and it provides management tools for Amazon and Rackspace. Users monitor their clouds through a dashboard, which also allows for tagging and color coding of nodes for easier identification. RightScale offers a cloud management platform that enables organizations to deploy and manage applications across multiple clouds. Kaavo offers enterprise management of cloud services from Amazon, Flexiscale and GoGrid. EnStratus offers a cloud management platform for enterprise applications running on Amazon and Rackspace clouds. CloudSwitch claims to move data center applications to clouds without modification, allowing customer to manage their apps from within the datacenter using existing tools and processes. DeltaCloud, an open source project aimed to develop an ecosystem of tools, is scripts and applications for the cloud. The project also aims to write a common, REST-based API to enable developers to write once and manage across multiple clouds [17].

As another application of multiple cloud usage, the BonFIRE Project [18] is providing a state-of-the art multi-site cloud facility for applications, services and systems research in the Internet of Services community.

The facility will give researchers access to large-scale virtualized compute, storage and networking resources with the necessary control and monitoring services for detailed experimentation of their systems and applications.

III. Use Case of Federated Cloud based GPES

A. Applicable Coverage of GPES Broker system

A software release life cycle is the sum of the phases of development and maturity for a piece of computer software. It consists of two periods, testing and development period and release period. And test and development period includes six different phases, such as pre-alpha, alpha, beta, open and closed beta, release candidate and release [19], [20].

In these phases, beta step generally begins when the software is feature complete. Software in the beta phase will generally have many more bugs in it than completed software in aspect of speed and performance issues.

So, the focus of beta testing is to reduce impacts to users, often incorporating usability testing. The process of delivering a beta version to the users is called beta release and this is typically the first time that the software is available outside of the organization that developed it.

The applicable coverage of proposed GPES Broker system is focused on the beta phase before open and closed beta phase that the service is released to restricted group of individuals for a user test.

B. Use Case

GPES Broker system abstracts incompatible capabilities and interfaces on behalf of GPES consumers to provide common, open and standardized ways for access to heterogeneous cloud service providers to be needed in GPES.

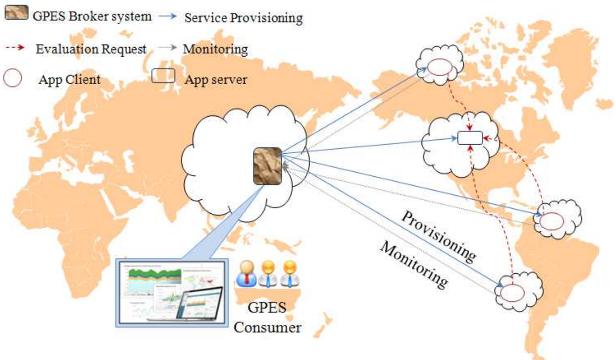

Fig. 1 Conceptual view of Global Performance Evaluation Service

Proposed GPES Broker system intermediates between GPES consumer and cloud service providers to allow GPES consumer to easily perform service performance evaluation work to know something, e.g. proper system scale, service delay or workload according to the number of service requesters in the target area.

And it will require several functionalities, such as prompt deployment to target location against multiple cloud providers, service and VM lifecycle management, automated monitoring and process about the evaluation data and so on. Fig.1 shows conceptual view and Fig.2 represents operation flow of the GPES. A multiple cloud based GPES allows Internet-based service providers to secure reliable performance evaluation information without repetitive service installation and visiting in service target area. In Fig.1, App Client and App Server means the target service (or application) to be evaluated by GPES. In the scenario, GPES portal is graphical user interface wrapping the GPES Broker system, so it supports integrated or advanced GUI based functionality using GPES Broker system's support. We assume target service consists of server and client part. In Fig.1, monitoring data includes resource usage, response time, the number of user connection, operation log and SLA related information.

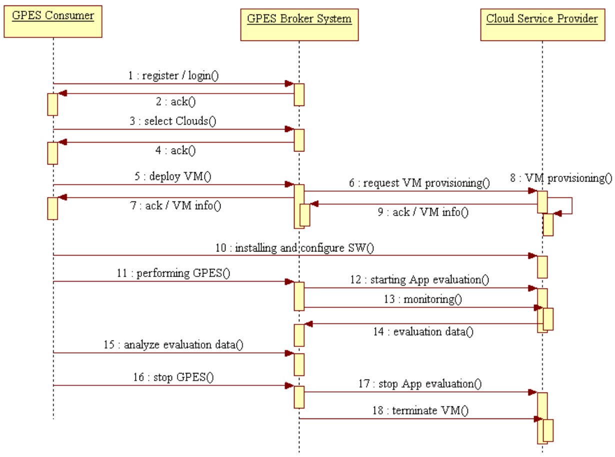

Fig. 2 Operational flow of Global Performance Evaluation Service

The related scenario of GPES is as below.

- A GPES consumer registers (for new users) or logins (for existing users) in the GPES portal.
- The GPES consumer selects the several target clouds to deploy the service to be evaluated, the selection will be based on SLA, security, price, geographical location and so on. And it includes locations for service server and clients and the system specifications
- The GPES Broker system deploys the related VMs for service server and clients to the selected clouds. In this step, GPES Agent is installed to the deployed VMs to monitor or control the service, server and clients.
- The GPES Consumer connects and configures the VMs deployed on the selected cloud. This step may include installing service related software.
- The GPES Consumer performs the planned performance evaluation on the clouds, such as service response time, resource usage based on workload variation and so on. In this step, he can control the lifecycle of related VMs and service on multiple clouds.
- GPES Broker system monitors the service and gathers evaluation data from related GPES Agent on VMs running on selected clouds and saves it. In this step, GPES consumer can access the evaluation data and analyze it.
- When the evaluation is finished, GPES Consumer secures realistic and reliable evaluation information based on the selected location. And it enables the service provider to make a prompt decision.

In step 5, GPES consumer can move service server to another cloud to find the proper location for better service performance. And he shall also transfer service clients between clouds to deploy and increase more virtual service users to another location. At this point, GPES consumer doesn't need to install or configure the related software again, because he can use existing VM instance that is already completely setup.

C. Primary Requirements

In this section, we describe primary requirements of GPES Broker system derived from use case and the main requirements are as below.

1. Multiple or federated cloud usage

GPES consumer shall select and use multiple or federated clouds in various regions according to their requirements, price, performance, geographical location and so on.

2. Automated service evaluation

GPES Broker system should support automated service evaluation environment to reduce or prevent repetitive SW installation and visiting to service target areas.

3. Management of VM and service running on multiple cloud

GPES Broker system should support integrated single management portal for VM and service running on multiple or federated clouds

4. Prompt VM provisioning and transmission between clouds

GPES Broker system should support prompt VM provisioning over multiple clouds and high speed VM transmission between heterogeneous clouds.

IV. Design of GPES Broker System

A. GPES Broker System Architecture

GPES Broker system intermediates global performance evaluation service between GPES consumer and cloud service providers to support automated evaluation environment.

The role of a GPES Broker system is to take in the functional and non-functional requirements from a GPES consumer and to perform a match with the proper resources available on the various cloud infrastructures it is linked with.

And it can combine IaaS from different cloud infrastructures to fulfill the GPES user's demand. After preparing the evaluation infrastructures, e.g. VM, GPES Broker system deploys target service into the infrastructure and performs required pre-planned service evaluation based on monitoring and analyzing the status of service performance.

In this section, we describe the architecture of GPES Broker system. GPES Broker system consists of three major components, GPES Portal, GPES Manager and Cloud Service Broker. The detail architecture is shown in Fig. 3.

GPES Portal is responsible for abstracting complex functionality and architecture of GPES Manager and Cloud Service Broker. So, it supports GUI based integrated and advanced functionality wrapping the interface of other two components. GPES Manager is responsible for monitoring and control of the services to be evaluated and Cloud Service Broker manages provisioning, monitoring and control of VM to be needed in GPES. Each VM will be deployed against multiple or federated clouds in various regions and target service will be run in the prepared VMs.

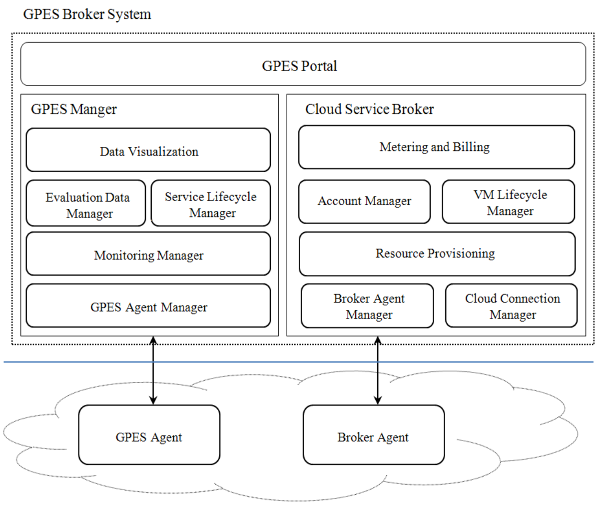

Fig. 3 Architecture of GPES Broker system

B. GPES Broker System Components

1. Cloud Service Broker and Agent

Cloud Service Broker manages provisioning, monitoring and control of VMs to be needed in GPES. And, as like GPES Manager, it co-works with Broker Agent to provide additional useful functionalities, such as high speed VM transmission between heterogeneous clouds.

The role of Metering and Billing component is the management of the platform price list comprising the prices of the different categories and provisioning operations.

And Metering part gathers the information related with billing, such as resource usage, uptime, log data by running VMs.

Account Manager is responsible for GPES consumer's access control and security. In this system, we use two layered account management method that Cloud Service Broker secures representative account about each cloud instead of GPES consumers. And Cloud Service Broker manages each consumer's access account based on it. So, in aspect of GPES Broker system, it has one account per each cloud. This way provides simple and comfortable method to aggregate multiple clouds.

VM Lifecycle Manager supports the functionalities to monitor and control VM status. According to the GPES steps, VM should be changed in its status, e.g. start, resume, restart, stop and terminate.

Resource Provisioning component is to promptly deploy VMs used by GPES to multiple clouds. And it performs this operation based on GPES consumer's requirements, such as geographical location, price, security level, performance and resource matching. This component also supports high speed VM transmission between heterogeneous clouds through Broker Agent in each cloud. Broker Agent provides a complete abstraction of the portability problems, especially about different APIs, for the GPES Broker system. It also enables to move from one cloud to another by VM image conversion to run on various cloud infrastructures.

Cloud Connection Manager is to aggregate multiple cloud capacity from different cooperating cloud service providers by proxy their APIs. So, it is responsible for the management of generic provisioning contracts as requested by the GPES consumer during the creation of the VM instance.

2. GPES Manager and Agent

GPES Manager is responsible for monitoring and control of the services to be evaluated and deals with the performance evaluation data.

The end point of monitoring data collection is provided by the GPES Agent, it resides on the provisioned VM and data collected in this way will be returned to the Monitoring Manager in GPES Manager. GPES Agent gathers service related information including resource usage, response time, the number of connection success or failure, operation log and SLA related factors.

Service Lifecycle Manger controls the service status running on VMs in multiple clouds, According to the GPES processing steps, service should be configured, start and stop to secure performance evaluation data. Evaluation Data Manager support a data access functionalities for GPES consumers, it stores evaluated data to predefined storage and allows consumers to securely access to the data and analyze it. GPES Agent Manager handles a lot of GPES Agents dispersed on multiple clouds. GPES Agent Manager embeds GPES Agent into VMs when it is deployed and manages agent related information including connection, location, status and so on.

Finally, Data Visualization component is to analyze gathered data and show it though graphical ways, such as graph and chart.

3. GPES Portal

GPES Portal is responsible for abstracting complex functionality and architecture of GPES Manager and Cloud Service Broker. So, it supports GUI based integrated and advanced functionality wrapping the interface of other two components, GPES Manager and Cloud Service Broker. *Primary interfaces supported by GPES Portal is as below*

- GPES consumer registration / delete
- Cloud registration / delete

- Consumer's requirement input for selection proper cloud
- VM provisioning
- VM lifecycle management including VM transmission
- Service lifecycle management
- Integrated monitoring view for VM and service
- Monitoring factors for GPES
- Visualization of evaluated information

V. Conclusion

In this paper, we proposed the use case and architecture of GPES Broker system. Currently, this system has been researched as a centralized model for global performance evaluation of internet service, but we will also consider decentralized model in future for high availability and flexibility in aspect of system architecture and portability

In case of decentralized model, each cloud service provider has its own Cloud Service Broker. GPES Manager can be deployed and placed in another location far from Cloud Service Broker. So, the relationship of Cloud Service Broker and GPES Manager can be loosed compared with centralized model.

This project is ongoing now. So, it is needed for proposed GPES Broker system to evaluate and verify the functionalities, performance and usefulness to be clear in next steps.

Acknowledgment

This work has been funded by the Korea Institute for Advancement of Technology (KIAT) within the ITEA2 project, Extendable Architecture and Service Infrastructure for Cloud-Aware Software (EASI-CLOUDS, 10014).

References

[1] Lee Badger, Tim Grance, Robert Patt-Comer, Jeff Voas, "DRAFT Cloud Computing Synopsis and Recommendation", NIST Special Publication 800-146, May. 2011.

[2] Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger, Dawn Leaf, "NIST Cloud Computing Reference Architecture", NIST Special Publication 500-292, Sept. 2011.

[3] Draft deliverable on Introduction to the cloud ecosystem: definitions, taxonomies, use cases, high level requirement and capabilities, Focus Group On Cloud Computing, ITU-T, 2010

[4] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. Rose, Rjkumar Buyya, "CloudSim: toolkit for modeling and simulation of cloud computing environment and evaluation of resource provisioning algorithms", Software: Practice and Experience, Vol. 41, Issue 1, pp. 23-50, 2011.

[5] Daryl C. Plummer, Benoit J. Lheureux, Michele Cantara, Tiffani Bova, "Predicts 2012 : Cloud Service Brokerage Will Bring New Benefits and Planning Chanllenges", Gartner, 2011.

[6] Shao Weixiang, Hu Jie, Bhumip Khasnabish, "Cloud Service Broker", IETF, March. 2012.

[7] Rajkumar Buyya, Rajiv Ranjan, Rodrigo N. Calheiros, "InterCloud : Utility-Oriented Federation of Cloud Computing Environment for Scaling of Application Services", University of Melbourne, 2010.

[8] Stefan Ried, "Cloud Broker-A New Business Model Paradigm", Forrester, 2011.

[9] Standard for Intercloud Interoperability and Federation, IEEE, P2302, http://standards.ieee.org/develop/project/2302.html

[10] Unified Cloud Interface, Cloud Computing Interoperability Forum(CCIF), 2012.

[11] Cloud Infrastructure Management Interfaces(CIMI), DMTF, xxx

[12] Distributed Systems Architecture Research Group, "Opennebula project," Universidad Complutense de Madrid, Tech. Rep., 2009, http://www.opennebula.org

[13] B. Rochwerger et al., "The Reservoir Model and Architecture for Open Federated Cloud Computing", *IBM Systems Journal*, Oct. 2008.

[14] SpotCloud, http://spotcloud.com/Technology.5.0.html

[15] J. Smets Solanes, C. Cerin, "SlapOS: A Multi-Purpose Distributed Cloud Operating System Based on an ERP Billing Model", Service Computing(SCC), pp. 765-766, 2011.

[16] CompatibleOne, "CompatibleOne Open Source Cloud Broker Architecture Overview", 2012.

[17] Linda Leung, "Cloud Computing Brokers: A Resource Guide", http://www.datacenterknowledge.com/archives/2010/01/22/cloud-computing-brokers-a-resource-guide/, 2010.

[18] BonFIRE Staff, "BonFIRE User Documentation Release 2.0", 2012.

[19] Jez Humble, David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, ISBN 0-321-60191-2.

[20] Software release life cycle, http://en.wikipedia.org/wiki/Software_release_life_cycle