# A Broker for Cloud Resource Management and Its Experimental Performance Analysis

Ye Ren, Seonghwan Kim, Dongki Kang, Chan-Hyun Youn
Dept. of Electrical Engineering, Korea Advanced Institute of Science and Technology
e-mail : {catherine, s.h_kim, dkkang, chyoun}@kaist.ac.kr

## ABSTRACT

When users access to use the computing resources in the cloud, they expect specific quality of service (QoS) which should be guaranteed by the service provider. Meanwhile, the service provider should adopt proper schemes to enhance the resource utilization. In this thesis, we propose the MapChem-Broker which aims to satisfy users' QoS requirements as well as enhance the resource utilization by controlling the provision of VM resources in the cloud. On the experimental cloud testbed, we compare the proposed scheme with an existing one for VM resource provisioning. Results show that the proposed scheme outperforms the existing one.

## 1. Introduction

Nowadays people's lives have been more and more interactive with and beneficial from a multitude of applications which assist us to live and work in a smarter and healthier [1] way than ever before. In the field of chemistry/pharmacy, MapChem [2] is an integrated web service for collaborative pharmaceutical research. Running the MapChem application can bring about heavy computation workload which is usually time-consuming if solely processed by the local device. Cloud computing enables users to submit job processing requests and run jobs in the cloud environment. Users submit their job requests along with some Quality of Service (QoS) constraints and service providers need to guarantee the required QoS and reach a Service Level Agreement (SLA). Meanwhile, service providers need to effectively manage the dynamic and cost-driven cloud environment. Therefore, a solution is desired by which the service provider can satisfy use's QoS requirement as well as best utilize the cloud resources.

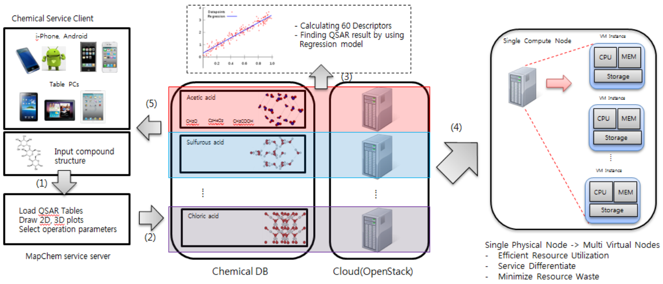


Figure 1 MapChem Service Overall Process

In this paper, a MapChem-Broker has been proposed and implemented which can guarantee users' QoS as well as enhance the resource utilization. As for the experimental evaluation, we compare the proposed resource provisioning scheme with another existed scheme under the cloud testbed.

## 2. Framework & Scheme of the MapChem-Broker

The broker is the middleware between the user and the service provider by controlling the resource allocation that aims to satisfy users' QoS requirements as well as best utilize the cloud resources. As shown in Figure 2, each job request submitted by the user is with its application policy such as time constraint or budget constraint and such policy are declared through specific QoS parameters in the $SLA_1$ between the user and the broker. There can be many QoS requirements specified in the $SLA_1$ by the user. In our framework, the QoS parameters only include the time constraint and the budget constraint for executing the job request. This $SLA_1$ is mapped by the broker to $SLA_2$ and the $SLA_2$ will include specific expected resource parameters of the needed computing resource.

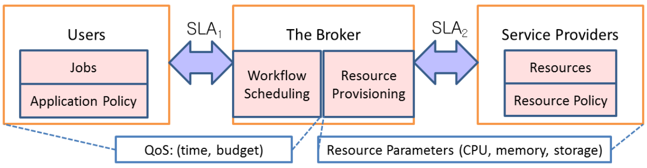


Figure 2 Bipartite SLAs among users, the broker, and service providers

Each resource type offered by the service provider can be modeled with two elements: the job queue and the resource capability. The job queue of a resource type keeps an ordered set of jobs scheduled to this resource type but not yet executed. The capability is the computational speed of the resource configuration, which can be closely related to three parameters – CPU type, memory size, and storage/hard disk size. Therefore, in this thesis, we define the resource parameters in $SLA_2$ consist of CPU type (number of unit cores) and the memory size, since processing the MapChem services does not require storage. Accordingly, we define the available resource policy as shown in Table 1. A more powerful resource type with higher computing capability is more expensive in terms of monetary cost. The MapChem-Broker will obtain such resource policy from the service provider beforehand.

In addition, due to resource capability limits, each instance of a certain resource type has an upper limit for the number of job requests that it can concurrently process without severe performance degradation. In order to enhance the resource utilization, the broker needs to maintain that each resource instance process the maximum but under-upper-limit number of job requests. The specific upper limits for each kind of job request processed on each type of provided resources need to be tested and confirmed through a great number of experiments. In this thesis, out of simplification, we roughly estimate the upper limits and manually set these values in the experimentation.

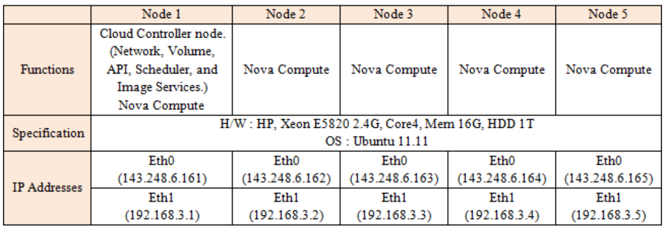
Table 1 Definition of VM Resource Types

| **Definition of VM Resource Types** |
|---|
| TINY (2.8GHz 1 Core, 1GB RAM) |
| SMALL (2.8GHz 2 Cores, 2GB RAM) |
| MEDIUM (2.8GHz 4 Cores, 4GB RAM) |

Combining all these above information, the MapChem-Broker will make a resource provisioning schedule for the current job request. First, the MapChem-Broker decides the resource type for processing the current job request using the algorithm of similarity degree [3] and then the job request is dispatched to the job queue of that decided resource type. Second, for each to-be-processed job request in the queue, among all current instances of the decided type, the MapChem-Broker will choose the one that is processing the most job requests and yet not exceeding its upper limit. Finally, the processing job will be scheduled to that specific instance. Through such a scheme, the resource provisioning will meet users' QoS requirements and also best utilize the resources in the cloud.

## 3. Experiments and Discussion

We use the MapChem web service as the application model, where a job request means analyzing an input SDF file and computing the QSAR table. The cloud testbed is configured with OpenStack as shown in Figure 3. On such a testbed we compare the proposed resource allocation scheme with the existing Dynamic Scaling Scheme (DSS) [4], both are for the VM management in the cloud environment.



| | Node 1 | Node 2 | Node 3 | Node 4 | Node 5 |
|---|---|---|---|---|---|
| Functions | Cloud Controller node.<br>(Network, Volume, API, Scheduler, and Image Services.)<br>Nova Compute | Nova Compute | Nova Compute | Nova Compute | Nova Compute |
| Specification | H/W : HP, Xeon E5820 2.4G, Core4, Mem 16G, HDD 1T<br>OS : Ubuntu 11.11 | | | | |
| IP Addresses | Eth0<br>(143.248.6.161) | Eth0<br>(143.248.6.162) | Eth0<br>(143.248.6.163) | Eth0<br>(143.248.6.164) | Eth0<br>(143.248.6.165) |
| | Eth1<br>(192.168.3.1) | Eth1<br>(192.168.3.2) | Eth1<br>(192.168.3.3) | Eth1<br>(192.168.3.4) | Eth1<br>(192.168.3.5) |

Figure 3 Experiment Testbed Configuration

We generate job requests at different interval time for a consecutive 10 mins and the job requests' type are randomly selected from the three kinds of SDF input files, sdf_50, sdf_200, sdf_400. We measure the total time span that is from the first job request generated time to the time that the system finishes processing all the job requests; and name this total time span as the total completion time whose inverse value is the numeric value of "performance". The total cost is the sum of every individual cost of each VM that has been used in the whole processing time. Cost-performance is an integrated metric to evaluate the cost-efficiency of the resource utilization. In this thesis, the cost-performance is defined as the Performance divided by the Total Cost. From results shown in Figure 4, we see that the proposed scheme outperforms the existing DSS in terms of a higher cost-performance. This advantage owes to the similarity degree algorithm in resource type selection and also that the proposed scheme allows for multiple job requests concurrently processed within one VM instance.

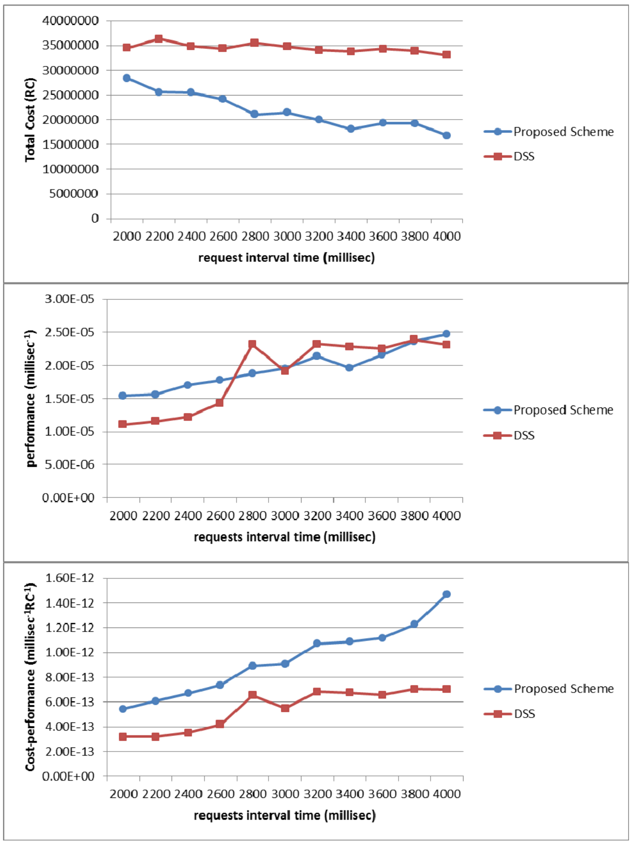


Figure 4 Comparisons between the proposed scheme and DSS

## 4. Conclusion

In this paper, we proposed the MapChem-Broker which implements a VM resource provisioning scheme in order to guarantee user's QoS requirements as well as enhance resource utilization. MapChem web services are deployed as the job requests model. Through the experimental evaluation, we show the proposed VM resource provisioning scheme outperforms the DSS in terms of higher cost-performance.

## Reference

[1] SangJin Jeong, Chan-Hyun Youn, *et al.* "An Integrated Healthcare System for Personalized Chronic Disease Care in Home-Hospital Environments". IEEE Transaction on Information Technology in Biomedicine, 2012.

[2] PharosChemTM and PharosMapTM User Manual Version 2.1.X, distributed by EQUIS & ZAROO and Advanced Network and Computing Lab at KAIST, Korea. 2012.

[3] Shengzhong Deng. "A Study on Policy Adjuster Integrated Grid Workflow Management System". M.S. Thesis, Information and Communication University, Daejeon, Republic of Korea, 73 pages. 2008.

[4] Trieu C. Chieu, Ajay Mohindra, *et al.* "Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment". IEEE International Conference on e-Business Engineering, 2009.