http://dx.doi.org/10.5392/IJoC.2012.8.4.007

Adaptive Scheduling for QoS-based Virtual Machine Management in Cloud Computing

Yang Cao*, Cheul Woo Ro(Corresponding author)**
*Dep. of Information & Technology. Eastern Liaoning University, Dandong, CHINA
**Dep. of Computer Eng. Silla University, Busan, KOREA

ABSTRACT

Cloud Computing can be viewed as a dynamically-scalable pool of resources. Virtualization is one of the key technologies enabling Cloud Computing functionalities. Virtual machines (VMs) scheduling and allocation is essential in Cloud Computing environment. In this paper, two dynamic VMs scheduling and allocating schemes are presented and compared. One dynamically on-demand allocates VMs while the other deploys optimal threshold to control the scheduling and allocating of VMs. The aim is to dynamically allocate the virtual resources among the Cloud Computing applications based on their load changes to improve resource utilization and reduce the user usage cost. The schemes are implemented by using SimPy, and the simulation results show that the proposed adaptive scheme with one threshold can be effectively applied in a Cloud Computing environment both performance-wise and cost-wise.

Keywords: Virtual Machine Scheduling, Dynamic Resource Allocation, Cloud Computing, SimPy

1. INTRODUCTION

Cloud Computing, the new computing paradigm, can be viewed as a dynamically-scalable pool of resources (including computing power, storage, platform, and service) which are accessible by the users via the Internet [1]. Companies can rent a large IT infrastructure on a short-term pay-per-usage basis which is only a fraction of the cost of maintaining a supercomputer center.

Virtualization is one of the key technologies enabling Cloud Computing functionalities. Cloud Computing has taken that degree of efficiency and agility realized from virtualization and magnified it further. Through pooled resources with geographic diversity and universal connectivity, Cloud Computing has facilitated delivering hosted software, platforms and infrastructure as a service. One of the virtualization types is server virtualization (also called Virtual Machines), which is employed as computing resources for high performance computing. Amazon EC2 [2], one of the application examples, permit their customers to allocate, access and control a set of VMs running inside their data centers, and adopt pay-as-you-go mode which charges customers for the period of time the VMs are allocated (also called the Leased Time).

In general, VMs have four key characteristics that benefit the user [3]:

-Compatibility: VMs are compatible with all standard x86 computers.
-Isolation: VMs look like physically isolating from each other.
-Encapsulation: VMs encapsulate a complete computing environment.
-Hardware independence: VMs run independently of underlying hardware.

From the security point of view, VMs run isolated ensuring that applications and services running within a VM cannot interfere with the host OS or other VMs. This provides an additional protection against malicious or faulty codes. VMs can achieve optimal hardware resource utilization since it can be easily moved, copied, and reallocated between host servers.

Virtualization results in significant savings through consolidation of infrastructure, but also brings new problems with VM expansion and over-allocation of resources that often affect the efficacy of Cloud Computing. In addition, when a VM is requested, the IT administrators have to perform several manual operations (in this paper, we call it booting time), which increase delays and administrative overhead.

In this paper, we study the resource allocation and scheduling at the application level, instead of mapping the physical resources to virtual resources. Since the Cloud is cost-associative, the only thing to pay is for the computing time of running each VM and for data transfers in and out of the Cloud [4]. Two dynamic VMs scheduling and allocating schemes are presented and compared. One dynamically on-demand allocates VMs while the other one deploying optimal threshold to control the scheduling and allocating of VMs to achieve both of better performance and cost-effectiveness.

The rest of this paper is organized as follows. Corresponding literature review is elaborated in Section II. Section III

* *Corresponding author, Email: cwro@silla.ac.kr*
Manuscript received Nov. 08, 2012; revised Dec 07, 2012; accepted Dec 17, 2012

describes the system modeling and schemes applied. Section IV presents the measure of interests as well as the performance evaluation results. The paper is summarized in Section V.

2. LITERATURE REVIEW

Lots of researchers have paid attention to virtual resources management in Cloud Computing, especially the VMs scheduling and allocating in order to get high system performance while providing satisfied QoS level.

In [5], the issues of job re-shaping and VM resizing are investigated, and the effects of the jobs' efficiency/scalability curve are studied. It looks separately into the benefits gained from size changes under constant efficiency (work-conserving adaptation) and different levels of efficiency changes (non-work-conserving adaptation). It gets conclusion from the results that the change in efficiency and subsequently work load constitutes the dominating performance factor.

Niyato et al. [1] consider different scenario under which the issues of virtual machine management and cooperation formation are presented and analyzed. They propose the cooperative virtual machine management consists of virtual machine allocation and cost management. Based on this, they proceed to analyze the cooperation formation among organizations to minimize their individual costs by using network game approach.

Kyi and Naing [6] in their paper present an efficient scheduling algorithm named Efficient Virtual Machines Scheduling Algorithm (EVMSA). In this algorithm, they employ multiple queues following the FCFS principle, and use three effective policies for resource allocation. Interacting stochastic Markov models is adopted to analyze the performance of efficient scheduling and allocation on Eucalyptus private cloud system. Their proposed approach enables capturing many realistic features of a large sized cloud, with reduced complexity of analysis.

Lin et al. [7] in their paper present a dynamic resource allocation scheme for Cloud Computing using threshold. The main idea of their allocation scheme is to monitor and predict the resource needs of the Cloud applications and adjust the number of VMs based on applications actual requires. They compare this scheme with normal static resource allocation scheme and prove that the dynamic allocation scheme is effective.

Kumar and Palaniswami [8] present a novel Turnaround time utility scheduling approach and focus on how to improve the efficacy of the algorithm. The algorithm designates high priority for task of early completion and less priority for abortions/deadlines issues of real time tasks. The algorithm has been implemented on both preemptive and Non-premptive methods. The experimental results shows that the proposed preemptive scheduling algorithm is more effective compared to the existing utility based non-preemptive scheduling algorithms.

Moschakis and Karatza [9] improve their previous model [4] which applies gang scheduling policy into the virtual machine scheduling and allocation in Cloud Computing. They integrate job migrations and starvation handling scheme into that model. Cost metrics are also the key parameters for evaluation. The experimental results show that those schemes have a significant effect on the previous model.

3. SYSTEM MODELING

The system architecture is shown in Fig.1. The simulation model consists of Datacenter and Broker, as well as a single cluster of VMs connected them together. The Datacenter manages a VM pool (with total of 300 distributed VMs) and is responsible for scheduling VM in/out from the VMs pool; The Broker acts on scheduling jobs (user applications) and allocating them to corresponding VMs. The Datacenter waits for requests from brokers, and provides extra virtual resources (VMs) or revoke excessive virtual resources (VMs) based on the requests from brokers. Initially, the system leases no VMs so the cluster is empty. Depending on the workload changes, the system has the ability to lease new VMs up to a total number of V_{max} = 300. However, not every job request will be accepted by the broker since there may be not enough VMs available at the moment. Therefore, if there is no sufficient VM for the request, the broker will place this request job on the appropriate waiting queue. The scheduling and allocating jobs in the waiting queue as well as to the VMs follow the FCFS scheduling policy.

For the sake of simplicity, we suppose that the processing ability of each VM is the same (C_{VM}=10 jobs), and the execution time of each job is equal (J_{dur} =10s). We suppose each VM has a fixed booting time (VM_{bt}=10) and after the initialization process the VM can be scheduled to the cluster and execute user jobs.

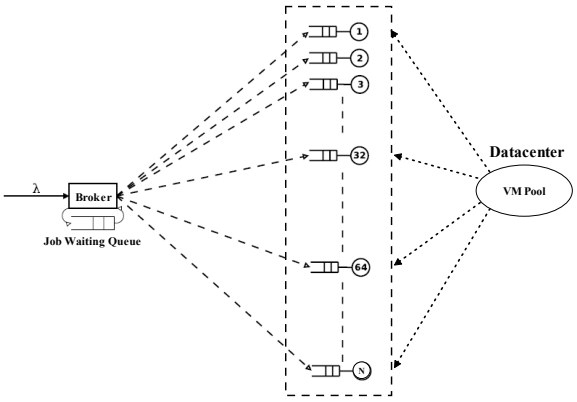

Fig. 1. The System Model.

Since reallocating virtual resources for cloud applications cost computing time and physical resources, over-frequent allocation of resources may reduce the efficiency of cloud applications.

Two dynamic VMs scheduling and allocating schemes are presented and compared:

Initially, the system leases no VMs so the cluster is empty.

Then following analog Exponential distribution of arriving jobs, at each moment, random number of jobs come and request for resources.

Scheme 1. Adaptive scheduling without threshold

In this scheme Broke sends VM request to Datacenter as soon as jobs arrive. The Datacenter then boots a VM and schedules it to the cluster and executes the job until the current workload exceeds the VM capacity, then the Datacenter boot another VM, and so forth. The Datacenter monitors the cluster of VMs, whenever a VM is idle (without running any jobs), the Datacenter withdraws it back to the VM pool.

Scheme 2. Adaptive scheduling with optimal threshold

In the second scheme, we have tried different thresholds for the Datacenter to monitor the workload of VMs in the cluster with the aim to ensure the effective utilization of resources on the one hand, to get better cost-effectiveness (by decreasing VM leased time) for the users on the other hand. The detailed process is like that: the Broke sends VM request to the Datacenter as soon as first jobs arrive. The Datacenter then boots a VM, schedules it to the cluster and executes the job. At that time if the current workload of a VM exceeds the predefined threshold value, then Datacenter will pre-boot a new VM and add it to cluster for running jobs.

We have tried several kinds of threshold so as to find the optimal one with better performance and cost-effectiveness. One threshold scheme means that when the current workload of a VM exceeds the predefined threshold value, then the Datacenter will pre-boot a new VM and add it to cluster for running jobs. This will decrease job waiting time for better QoS. We also tried two thresholds scheme which means that when the current workload of a VM exceeds the predefined upper threshold value, then the Datacenter will pre-boot a new VM and add it to cluster for running jobs; when the current workload of a VM goes below the predefined lower threshold value, then the Datacenter will try to finish the work of that VM, remove it from the cluster and withdraw it back to the VM pool in order to save leasing cost for the users.

With the Broker and Datacenter work together through adaptively scheduling jobs and VMs, both the system and the users get following benefits:

-Improving Agility of VM delivery by dynamical on-demand provisioning;

-Arresting VM Sprawl with demand-based threshold controlling of VMs' acquiring and releasing;

-Increasing system performance with intelligent allocation and de-allocation of virtual resources;

-Presenting cost-saving approach for users.

-Preventing performance and cost issues caused by long running VMs through releasing virtual machines according to predefined threshold.

4. MODEL ANALYSIS

4.1 Measures of Interest

In order to obtain the interested measures numerically from the models, SimPy simulation package is adopted. SimPy (Simulation in Python) [10] is adopted to build our simulation models to get analytical data, which is an open source simulation package for process-oriented discrete events simulation (DES) based on Python. SimPy is process-oriented, which means that it provides the modeler a number of tools including processes to model active entities, three kinds of resource facilities (Resources, Levels, and Stores) and ways of recording results, which allows us to create models of complex discrete event systems conveniently and easily.

- Mean Response Time (MRT)

Response time ***Rj*** of a job *j* is the time interval between the arrival and the departure of the job. Its average is defined as:

$MRT = \frac{\sum_{j=1}^{N} Rj}{N}$, Where N is the total number of jobs arrived.

- VM Mean Leased Time (MLT)

Leased time Vl*i* is the total usage time of a VM *i* by users applications. Its average is defined as:

$MLT = \frac{\sum_{i=1}^{M} Vli}{M}$, Where M is the total number of VM leased.

- Cost-Performance Efficiency (CPE)

We then define a formula to represent Cost-Performance Efficiency which is evaluated by combining MLT with the MRT metric. ***w*** is the weight to take use of MLT. Here we just let ***w***=10. The formula is as the following:

$$CPE = MRT + MLT / w$$

4.2 Input Data

We suppose the Datacenter manages distributed virtual machines total number V_{max} =300, each VM needs 10s booting time to initialize and become active for use, and each VM has the same capacity to processing C_{VM} =10 jobs at most; each job has the same execution time J_{dur} =10s, and the arriving of jobs follows some Exponential-analog distributions which are defined functions in our algorithm. We change the arriving rate λ value to get different random number of arriving jobs to emulate the uncertainty of realistic environment.

The parameters we used are listed in Table 1.

Table 1. Simulation Parameters

Parameters	Meaning	Values
λ	Job arriving rate	[0.1, 1.0]
V_{max}	Total number of VMs in VM pool	200
C_{VM}	Capacity of each VM	10 jobs
J_{dur}	Execution time of each job	10s
VM_{bt}	Booting time of each VM	10s

We first try different threshold values to get the optimal one. Fig. 2 shows the CPE changes when using different thresholds. We got the optimal value when setting one threshold with 0.5. So the following experiments are based on this value with one threshold for Scheme 2.

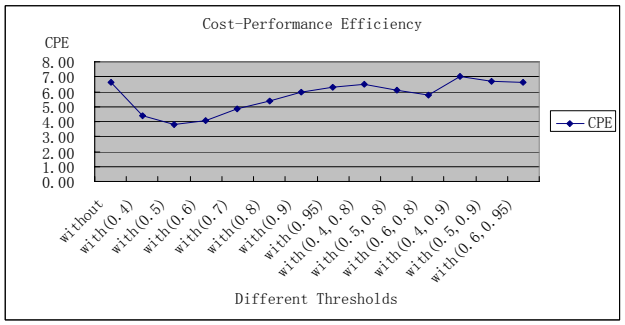

Fig. 2. CPE with Different Thresholds.

4.3 Numerical Results

We change the job arriving rate λ, and compare the two adaptive scheduling schemes we present. Fig. 3 shows the average number of jobs at time interval with different arriving rate λ. Fig. 4~6 shows the simulation results of MRT, MLT and CPE with different jobs arriving rate λ, respectively. S1 and S2 in the legend refer to Scheme 1 and Scheme 2, respectively.

With the increasing of jobs arriving rate, both scheme can keep better MRT, while Scheme 2 with optimal threshold shows better results than Scheme 1 because it can pre-booting VM and schedule it to the VMs pool in time which decreases the system response time. This pre-scheduling and allocating VMs also shortens the MLT which effectively saves users cost. Thus the total CPE of Scheme 2 is better than Scheme 1.

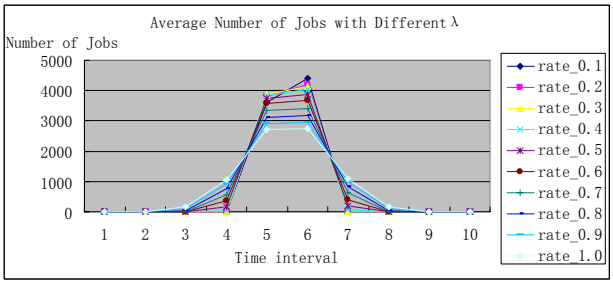

Fig. 3. Average Number of Jobs at Time Interval with Different Arriving Rate λ .

Fig. 4. Jobs Mean Response Time.

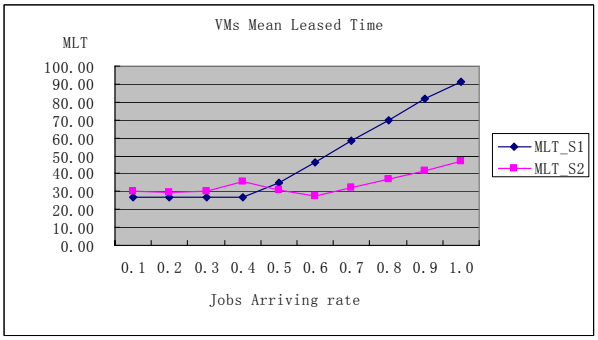

Fig. 5. VMs Mean Leased Time.

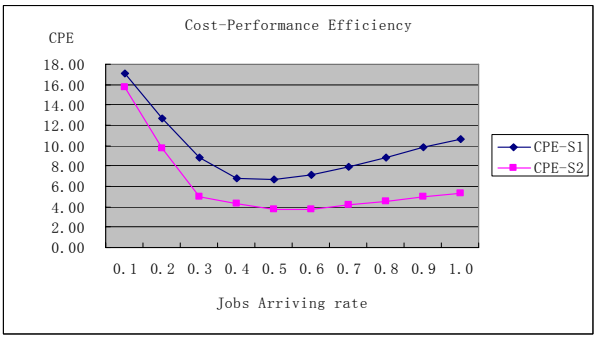

Fig. 6. Cost-Performance Efficiency.

5. CONCLUSION

Virtual resource management is a key issue in Cloud Computing environment, especially the VM scheduling and allocating is crucial not only to the service providers, but also to the Cloud users. In this paper, two adaptive scheduling schemes for VM scheduling and allocating are presented and compared. Simulation results reveal that the proposed adaptive scheme with one threshold can be effectively applied in a Cloud Computing environment both performance-wise and cost-wise. Our future work will further focus on different scheduling policies as well as the consideration of the heterogeneity of user application and Cloud resources.

REFERENCES

[1] D. Niyato, Z. Kun and P. Wang. Cooperative Virtual Machine Management for Multi-Organization Cloud Computing Environment. *Proc. ICST-PEMT'11*, 2011, pp. 528-537.

[2] Amazon Elastic Compute Cloud (Amazon EC2). http://aws.amazon.com/ec2/, 2012.

[3] Virtualization Basics. http://www.vmware.com/virtualization/ virtual-machine.html.

[4] I. A. Moschakis and H. D. Karatza. Evaluation of Gang Scheduling Performance and Cost in a Cloud Computing System. Journal of Supercomputing, vol.59, no.2, Feb. 2012, pp. 975-992.

[5] A. C. Sodan. Adaptive Scheduling for QoS Virtual Machines under Different Resource Allocation--Performance Effects and Predictability. Job Scheduling Strategies for Parallel Processing, pp.259-279, Springer-Verlag Berlin, 2009.
[6] H. M. Kyi and T. T. Naing. Stochastic Markov Model Approach for Efficient Virtual Machines Scheduling on Private Cloud. International Journal on Cloud Computing: Services and Architecture, vol.1, no.3, Nov. 2011, pp. 1-13.
[7] W. Lin, J. Z. Wang, C. Liang and D. Qi. A Threshold-based Dynamic Resource Allocation Scheme for Cloud Computing. Procedia Engineering, vol. 23, Dec. 2011, pp.695-703.
[8] V. V. Kumar and S. Palaniswami. A Dynamic Resource Allocation Method for Parallel Data Processing in Cloud Computing. Journal of Computer Science, vol. 8, no. 5, Aug. 2012, pp.780-788.
[9] I. A. Moschakis and H. D. Karatza. Performance and Cost evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling. *Proc. ISCC'11*, 2011, pp. 418-423.
[10] N.S. Matloff. Introduction to Discrete-Event Simulation and the SimPy Language. http://heather.cs.ucdavis.edu/~matloff/156/PLN/DESimIntro.pdf, 2008.

Cheul Woo Ro
Refer to IJOC, Vol.4, No. 3, 2008.9

Yang Cao
Refer to IJOC, Vol.8, No. 2, 2012.6