An Intelligent Approach for Virtual Machine and QoS Provisioning in Cloud Computing

Amit Kumar Das, Tamal Adhikary
and Md. Abdur Razzaque
Dept. of Computer Science and Engg.
University of Dhaka, Bangladesh
Email: (amit.csedu,tamal.csedu)@gmail.com
Email:razzaque@cse.univdhaka.edu

Choong Seon Hong
School of Electronics and Information
Kyung Hee University, Suwon, South Korea
Email: cshong@khu.ac.kr

***Abstract*—Cloud Computing has become the most popular distributed computing environment because it does not require any user level management and controlling on the low-level implementation of the system. However, efficient resource provisioning is a key challenge for cloud computing and resolving such kind of problem can reduce under or over utilization of resources, increase user satisfaction by serving more users during peak hours, reduce implementation cost for providers and service cost for users. Existing works on cloud computing focuses to accurate estimation of the capacity needs, static or dynamic VM (Virtual Machine) creation and scheduling. But significant amount of time is required to create and destroy VMs which could be used to serve more user requests. In this paper, an adaptive QoS (Quality of Service) aware VM provisioning mechanism is developed that ensures efficient utilization of the system resources. The VM for similar type of requests has been recycled so that the VM creation time can be minimized and used to serve more user requests. In the proposed model, QoS is ensured by serving all the tasks within the requirements described in SLA. Tasks are separated using multilevel queue and the most urgent task is given high priority. The simulation-based experimental results shows that a great number of tasks can be served compared to others which will help to satisfy customers during the peak hour.**

I. Introduction

Over the last few years, cloud computing has gained high prominence as a distributed computing environment and has been paid wide attention by academic organizations, government as well as small, medium and large scale industries. Cloud computing has become an essential technology trend and commercial cloud platforms have begun to be deployed [1], [2], [3]. The providers of cloud computing technology offer different kinds of services to users which include programs, storage, application-development platforms over the Internet, hardware resources for deploying user friendly platform etc. Users can access cloud computing services using a variety of devices including PCs, smartphones, laptops, PDAs etc.

The cloud services can broadly be classified into three categories: 1) IaaS (Infrastructure as a Service), in which the service providers render physical infrastructure for provisioning computational platform; 2) PaaS (Platform as a Service), where, software systems are provided that manage the hardware resources; and 3) SaaS (Software as a Service), where, application services are deployed and managed in a ubiquitous cloud infrastructure. Cloud computing can provide the same computing facilities like a supercomputer with a relatively cheaper cost and reduced computational expense. The services rendered by cloud platform are much more reliable than services in grid computing and are much more scalable than services that can be provisioned in the largest of commodity clusters. However, cloud computing still encounters a number of challenges including an efficient model of virtual machine provisioning which will ensure QoS (Quality of Service). Achieving QoS includes a number of parameters and properties to be fulfilled which encompasses subjective ones (packet loss, transmission rate, delay variance, cost and reputation etc.) as well as objective ones (data security, trust, privacy concern and user experience as well as degree of satisfaction etc.).

To enhance user satisfaction and to justify the investment in cloud based deployments, meeting up target QoS is necessary. Some existing works on QoS [4]-[9] have tried to provide assurance in meeting the SLA (Service Level Agreement). Some other works including [14] tried to control VM provisioning in proactive or reactive manner. Efficient resource management through VM multiplexing has been examined in [22]. However, the target of fulfilling SLA is a great challenge because of the uncertain and dynamic characteristics of network and IT resources in the distributed cloud platform.

In this paper, a QoS Aware Adaptive VM Recycling and Provisioning approach have been presented that will serve as an automated, flexible and efficient management of the cloud resources. The model ascertains that target QoS has been met by controlling the admittance of the requests so that the system does not get overloaded. The model also helps to ensure budget minimization by the optimized provisioning of IT resources. Here, multiple input queues has been designed for requests of similar requirement metrics of cloud resources and VM created for serving a request can be recycled or reused by other jobs of the same queue. Therefore, the VM creation and destroying time can be minimized to some extent. Here, the most urgent VM is selected using a priority metric and depending on the priority of requests and the resources availability, new VM is created.

The rest of the paper is organized as follows. The Section II describes some of the works related to our topics of interest. In

section III, the proposed model of QoS aware VM provisioning has been described along with the provisioning algorithm. The Section IV presents the result of performance evaluation and simulation. In section V, conclusion along with the direction for future research has been provided.

II. Related Works

The context of cloud computing environment has already been shifted to Data centers and Virtual Machines from the category of improving battery lifetime [10], [11]. Efficient resource sharing and scaling which are the key advantages of cloud platforms are mainly obtained by using virtualization which is mainly employed for fault isolation and improved manageability [12].

In [13], the authors proposed an energy management system for virtualized data centers where resource sharing is categorized into local and global policies. Virtual Machine provisioning based on some analytical results has been proposed by the authors of [14]. Automation, Adaptation, Performance assurance was the key factor in this paper. They have proposed a dynamic scenario for virtual machine by using SaaS, PaaS and IaaS layer of cloud. However, they didn't differentiate the type and requirement of every request and every time after finishing the task of virtual machine they have destroyed them. They chose to create new VM for every new task although the new task might have the same degree of resource requirement compared with the previous one. This type of approach is so much time consuming and is not suitable in every situation and in our approach we have tried to solve this problem by reusing the old VMs rather than creating.

In [15], the researchers have proposed the framework of adaptive QoS management Process, QoS framework for mobile cloud computing and they have modeled QoS management system based on FCM (Fuzzy Cognitive Map). How many requests will be accepted by the system, in what way the request is handled, what the system will do if it gets congested etc. are not clearly defined in this paper. In our approach, we have described the scenarios clearly.

Based on queuing networks the authors of [16] have proposed an architecture form provisioning multitier applications in cloud data centers. But the problem is that, such kind of model does not recalculate the number of required VMs based on expected load and monitor performance, as does our approach. A reactive algorithm for dynamic VM provisioning of PaaS and SaaS applications is proposed by the authors of[17] in a proactive manner.

A queuing infrastructure is proposed using SaaS mashup applications by [18], targeting to optimizing the benefit of reduced costs of the SaaS provider by finding an optimal number of instances for the application. A system 'Claudia' by name is developed by [19], where user defined cloud provisioning, based on performance indicators and elasticity rules. In this model the authors also used reactive approach whereas we applied proactive model for obtaining QoS.

In cloud, data center host level to manage power consumption of resources and performance of applications has proposed in 'Mistral system' in [20]. However, this method requires access to the physical infrastructure, which typical IaaS providers do not provide to consumers. When both infrastructure and application are offered by the same provider, then 'Mistral' is suitable to be applied. But our approach can be applied in both cases whether the services are provided by the same provider or IaaS and PaaS/SaaS providers are different organizations.

An architecture of energy management system for virtualized data centers where resource management is divided into local and global policies have proposed in [21]. Though local policies are described here in a way that the system leverages guest operating systems power management strategies, the global policies are not discussed in detail considering QoS requirements. We focus on VM allocation policies over the cloud, considering strict SLA (Service Level Agreement).

III. Proposed Model

A. System Architecture

In this section, a brief description of the working environment, the assumptions and meaning of the notations used for describing different parameter are provided. The system consists of a set of data centers named 'D'. Each of the data centers contains n physical servers. The set of physical server is given by 'S'. It is assumed that each of these servers has equal capacity of computing resources (e.g., servers, networks, storage, applications, and services etc.). The set of application instances is given by 'A' and the set of virtual machine is given by 'V'. The number of virtual machines required for serving an application depends on the application type and workload variance of the application with time. T_{req} is the requested negotiated time by the user within which the service of the application needs to be provided and T_{act} is the actual time needed to complete the task. To meet the requirement of QoS T_{act} should be less than or equal to T_{req}. The tasks need to be served is classified into groups depending on the requirement. Some of the existing cloud service providers support this type of facilities. For example Amazon EC2 provides 11 types of VMs, where the processor, memory and I/O performance of each type is different from others.

B. System Model

For controlling the uncertain behavior of the network elements and to synchronize with the dynamic workload changing in the cloud computing environment, an adaptive approach for VM provisioning and management for meeting QoS requirement is proposed in this section. The self adaptive QoS aware VM provisioning mechanism is shown in figure 1. There are a number of input queues depending on the types of requested services. The proposed model consists of the following components: 1) Admission Controller/Requirement Analyzer, the entry point of the service requests. When the cloud computing system is congested and QoS provisioning will not be possible for additional works, no SLA is promised with the customers and the admission of new application request is rejected. The task of the analyzer is to determine

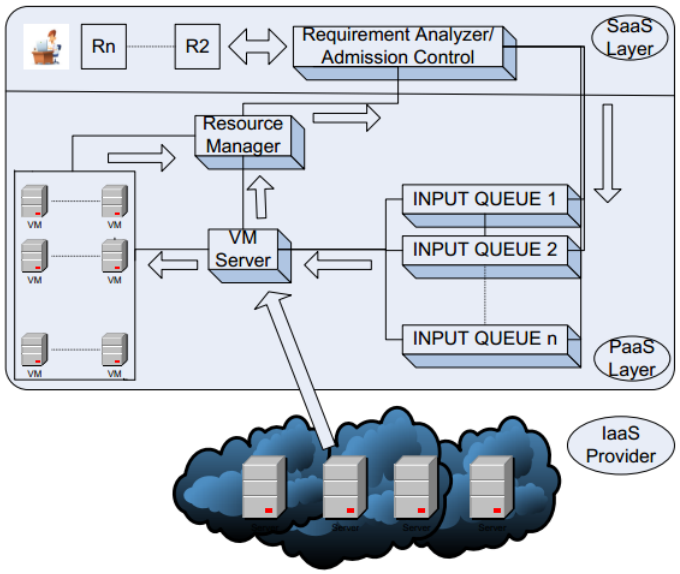

Fig. 1. Network Model

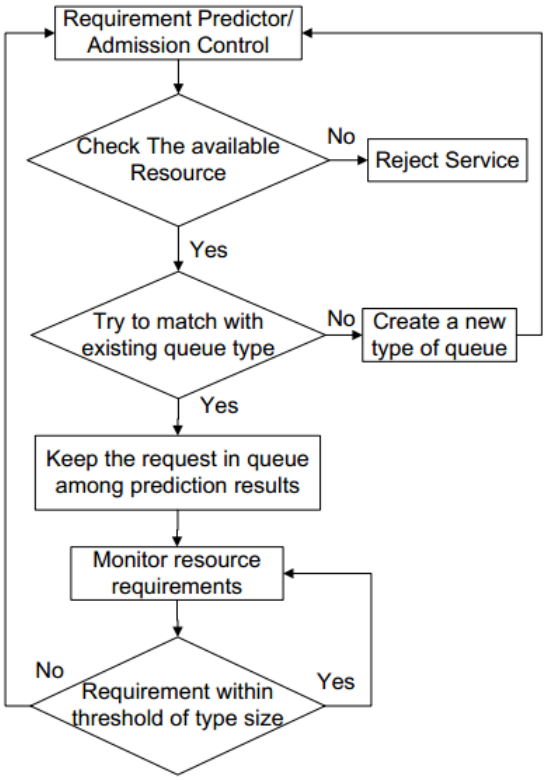

Fig. 2. QoS Management System

the amount of computational resource required and to predict the type of queue where a service needs to be forwarded. 2) Resource Manager, which determines the system contentedness and advice Admission Controller/Requirement Analyzer in making decision to let jobs enter into the system. 3) VM Server, which determines which of the request is time critical among the requests in all queues and provide virtual machine to that applications, before creating VM wait for those tasks which are waiting for their last request to be finished.

C. Adaptive VM Provisioning and QoS Management

Figure 2 gives an overview of the VM provisioning and QoS management Process. The admission controller/requirement analyzer first checks if the available resources are sufficient for servicing the newly requested job. If it finds that the job can be served within the time promised in SLA, it lets the job to enter the computing premise. Otherwise, it does not make any promise about the SLA. After entering into the system, the job is placed into one of the queues depending of the requirement necessary for the job. If no such queue is not found, new queue is formed and the job is placed in the new queue. Since the requirement changes dynamically with time and prediction may sometimes be wrong, if the requirement changes within a predefined threshold value, the VM of that job is resized to match the size of the job. If the size of the job becomes larger than the threshold value, new VM is provisioned and the two VM are then connected.

The main benefit of this mechanism is that new VM is not necessary to be created for all the jobs in a queue. Creation of VM is only necessary for time critical jobs and the other jobs can reuse the VM created for a time critical job of the same type. VM Server determines which jobs are time critical, which virtual machine can be deleted and which VM needs to be created. Hence, all the jobs can be completed within the negotiated time and QoS is maintained in this way.

D. VM and QoS Provisioning Algorithm

Here, QoS is maintained by letting requests to enter the system in a controlled way and judicious provisioning of VM. Before allowing a job to be served, the summation of negotiated time for all the jobs in service and in the queue is calculated and denoted by $T_{negotiated}$. Then, the total time required for completing all the jobs in service and in queue is estimated from the monitored mean execution time of a work, T_{mean}. This total required time is given by T_{total} and is added to a reserved value of time to cope with the uncertain behavior of the network elements and dynamic workload of the requests. Whenever a new job comes to admission controller/Requirement Predictor, it's actual working time T_{new_act} is predicted. If it's negotiated time T_{new_neg} combining with $T_{negotiated}$ becomes greater than the summation of T_{total}, T_{new_act}, threshold time and the total affordable service time $T_{service}$ is greater than the summation of $T_{negotiated}$ and $Tnew_act$, then the new job is allowed to enter into the queues to get service. The requests then enter into a queue that correspond its requirement of resources. Some VM for a queue is created and the jobs of major priority have the chance of getting executed early. Whenever the task of a VM completes, if it have followers in its queue, this VM is allocated to the job of most priority from

the queue and thus the time for creating and destroying VM becomes limited and performance of the system will upgrade. The requests are served according to the arrival time and the time hungriness of the job. All the queues are priority queue and the priority factor is given by:

$$Priority\_factor = arrival\ time + Negotiated\ time\ limit \quad (1)$$

VM server provides new virtual machine for the queue which has maximum total priority_factor for all of its jobs.

Algorithm 1 Adaptive VM *recycling and Provisioning algorithm*

INPUT: T_{mean} : *Monitored mean execution time*,
$T_{service}$: *Total affordable service time by the service provider*,
$ReservedTime$: *Time for secured QoS provisioning*,
n : *Number of jobs*,
j : *number of queues*
i : *number of jobs in queue j*
OUTPUT: *QoS aware VM provisioning*

1. $T_{negotiated} \Leftarrow \sum T_{req}$;
2. $T_{total} \Leftarrow n * T_{mean}$;
3. $T_{estimate} \Leftarrow T_{new\_act} + T_{total} + Reserved\_Time$;
4. $T_{max\_limit} \Leftarrow T_{negotiated} + T_{new\_neg}$;
5. **if** $T_{max\_limit} >= T_{estimate} \&\& T_{service} >= T_{max\_limit}$ **then**
6. Allow job to enter into input queue;
7. **else**
8. Reject job to enter into input queue;
9. **end if**
10. Calculate priority_factor;
11. **if** resources available for creating VM **then**
12. Repeat for all queue;
13. Calculate $priority_j \Leftarrow \sum priority\_factor_i$;
14. Find maximum $priority_j$ for creating new VM of type j;
15. **else**
16. Wait for VM of same type for completing current job;
17. **end if**

IV. Performance Evaluation

The result of simulation found from the experimental implementation of our proposed QoS aware VM recycling and provisioning algorithm is given in this section. CloudSim discrete event Cloud simulation tool was used for performing the experiments. The simulation environment was set up by a data center containing 100 hosts only each having quad-core processor and 8GB of RAM. Arriving rate of requests of jobs is considered to be 500 requests per second. The time for creating and deallocating a VM is considered to be 2-3 minute where the time for serving a request is considered to be 30-50 minute. The results show a comparative study of system performance between our proposed Intelligent Approach for Virtual Machine and QoS Provisioning (IVQ) and Virtual Machine Provisioning Based on Analytical Performance and QoS (AVM). The results show that significant amount of time can be saved from creating new VM which can be used to serve more requests and reduce request rejection ratio.

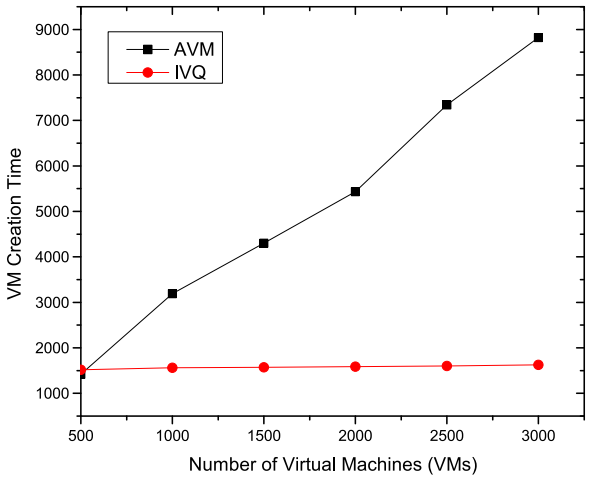

Fig. 3. VM number vs Creation Time

In the figure 3, the relation of time for VM creation with the number of request is shown. In the best case scenario, all the tasks are of a single type therefore all the created VMs can be recycled and the time for creation of VM decreases. In the worst case scenario, all the tasks are of different type and new VM is required for all the serving requests. Hence the performance of the proposed model will be similar to the existing models. Here, the simulation results show that if all the requests are of same type, our proposed model will take time only for creating VMs initially. The created VMs will then be recycled and therefore no time is spent for creating VMs. In AVM time for best case scenario and worst case scenario is the same. Our proposed IVQ will work like AVM in worst case.

In the figure 4, the comparative study on number of request

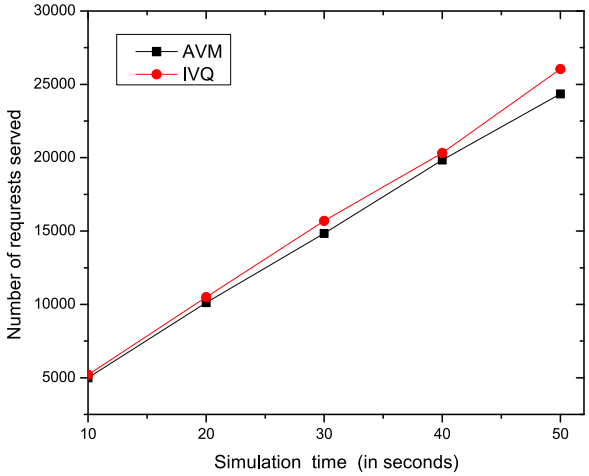

Fig. 4. Simulation Time vs Number of request served

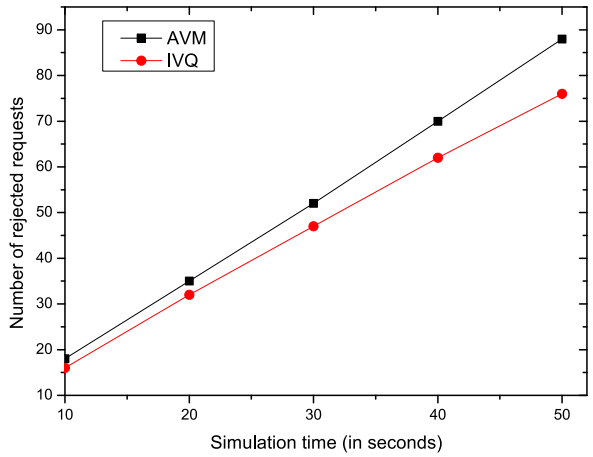

Fig. 5. Simulation Time vs No. of Rejection of requests

served with time between our proposed IVQ and AVM is given. Here we see that, as the time for creation of VMs can be eliminated in our proposed model, more number of requests can be served compared to AVM and hence user satisfaction and economic profit can be achieved. Here, as the time of service increases, the difference in serving requests between IVQ and AVM also increases.

In the figure 5, the rate of rejection of requests with time is given. Here again the comparison is made with AVM and it is found that less number of requests need to be rejected. This is due to the fact that,the time saved from creating new VM can be uesd to serve new request and hence the rate of service rejection decreases. Here the time stamps are of minutes for counting the number of rejections. Though the difference between IVQ and AVM is not very large for samll time stamps, the difference becomes bigger as the serving time, request rage or workload increases. Therefor, during the peak hour IVQ will serve more user by decreasing the rejection rate which will increase user satisfaction and resource utilization hence increase the profit of providers.

V. Conclusion

Though adoption of Cloud computing platforms as application provisioning environments has several benefits, there are still a lot of obstacles and complexities, for getting smooth performance and provisioning in this cloud sector. For removing some complexities in this provisioning environments in cloud computing we have proposed an intelligent approach for VM and OoS provisioning system.

In this paper we have presented an intelligent approach for VM and QoS provisioning system for Cloud data centers. We have defined the problem of making VM repeatedly and stated a QoS model to recover this problem. Moreover, we have proposed an algorithm for minimizing the rejection rate in Cloud data centers. The goal of the model is to meet QoS targets by optimizing rejection time in clouds. Our simulation-based experimental results indicate that our model will give a more reliable performance as well as it meets the QoS.

As future work we are planning to do resource allocation policies among VM and also we are planning to do resource allocation in case of VM multiplexing. We will work to make a new model including this model that will act as resource allocation model. Efficient memory access mechanism will also get priority as our future work derective.

Acknowledgment

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2012-0006421). Dr. CS Hong and Dr. Md Abdur Razzaque are the corresponding author.

References

[1] Amazon Elastic Compute Cloud, “http://aws.amazon.com/ec2,” *accessed date 10th August 2012*.

[2] Google App Engine, “http://code.google.com/appengine,” in *accessed date 10th August 2012*.

[3] Microsoft Azure Services Platform, “http://www.microsoft.com/azure,” *accessed date 10th August 2012*.

[4] G. Lodi, F. Panzieri, D. Rossi and E. Turrini, “SLA-Driven Clustering of QoS-Aware Application Servers,” in *IEEE Transactions on Software Engineering*, VOL. 33, NO. 3, pp. 186-197, March 2007.

[5] V. Stantchev and C. Schrofer, “Negotiating and Enforcing QoS and SLAs in Grid and Cloud Computing,” in *GPC ’09 Proceedings of the 4th International Conference on Advances in Grid and Pervasive Computing*, November 2009.

[6] X. Wang, Z. Du, X. Liu, H. Xie, X. Jia, “An adaptive QoS management framework for VoD cloud service centers. 2010 International Conference on Computer Application and System Modeling (ICCASM),Volume: 1, 2010, pp. 527-532.

[7] Y. Ye, N. Jain, L. Xia, S. Joshi, I-L. Yen, F. Bastani, K. L. Cureton, M. K. Bowler, “A Framework for QoS and Power Management in a Service Cloud Environment with Mobile Devices” in *2010 Fifth IEEE International Symposium on Service Oriented System Engineering (SOSE)*, pp. 236 - 243.

[8] Q. Li, Q. Hao, L. Xiao and Z. Li, “Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control,” in *2009 1st International Conference on Information Science and Engineering (ICISE)*, pp. 99 - 102, 2009.

[9] Y. Xiao, C. Lin, Y. Jiang, X Chu and X. Shen, “Reputation-Based QoS Provisioning in Cloud Computing via Dirichlet Multinomial Model,” in *2010 IEEE International Conference on Communications (ICC)*, pp. 1 - 5, 2010.

[10] R. Neugebauer and D. McAuley, “Energy is just another re-source: Energy accounting and energy pricing in the nemesis OS,” *in Proceedings of the 8th IEEE Workshop on Hot Topics in Operating Systems* , 2001, pp. 5964.

[11] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, “ECOSys-tem: managing energy as a first class operating system resource,” *ACM SIGPLAN Notices*, vol. 37, no. 10, p. 132,2002.

[12] R. Nathuji, A. Kansal and A. Ghaffarkhah, “Q-Clouds: Managing PerformanceInterference Effects for QoS-Aware Clouds,” in *EuroSys10*, 2010.

[13] A. Beloglazov and R. Buyya, “Energy Efficient Resource Management in Virtualized Cloud Data Centers,” in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*,2010.

[14] R. N. Calheiros, R. Ranjan and R. Buyya, “Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments,” in *Parallel Processing (ICPP), 2011 International Conference*, 2011.

[15] P. Zhang and Z. Yan, “A QoS-AWARE SYSTEM FOR MOBILE CLOUD COMPUTING,” in *Proceedings of IEEE CCIS2011*, 2011.

[16] J. Bi, Z. Zhu, R. Tian, and Q. Wang, “Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center,” in *Proceedings of the 3rd International Conference on Cloud Computing(CLOUD10)*, 2010.

[17] T. C. Chieu, A. Mohindra, A. A. Karve, and A. Segal, "Dynamic scaling of web applications in a virtualized cloud computing environment," in *Proceedings of the 6th International Conference on e-Business Engineering (ICEBE09)*, 2009.

[18] Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. hou, "Profit-driven service request scheduling in clouds," in *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid10)*, 2010.

[19] L. Rodero-Merino, L. M. Vaquero, V. Gil, F. Galan, J. Fontan, R. S. Montero, and I. M. Llorente, "From infrastructure delivery to service management in clouds," in *Future Generation Computer Systems*, vol. 26, no. 8, pp. 12261240, 2010.

[20] G. Jung, M. A. Hiltunen, K. R. Joshi, R. D. Schlichting, and C. Pu, "Mistral:Dynamically managing power, performance, and adaptation cost in cloud infrastructures," in *Proceedings of the 30th International Conference on Distributed Computing Systems (ICDCS10)*, 2010.

[21] R. Nathuji and K. Schwan, "Virtualpower: Coordinated power management in virtualized enterprise systems," in *ACM SIGOPS Operating Systems Review*, vol. 41, no.6, pp. 265278, 2007.

[22] X. Meng, C. Isci, J. Kephart, L. Zhang and E. Bouillet, "Efficient Resource Provisioning in Compute Clouds via VM Multiplexing," in *ICAC10*, 2010.