ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.4 , No.5, Pages : 34-39 (2015)
Special Issue of ICCTE 2015 - Held during 23 – 24 November, 2015 in Hotel Crowne Plaza, Dubai
http://warse.org/IJSAIT/static/pdf/Issue/iccte2015sp06.pdf

Economy-Aware Cloud Resource Provisioning Under SLA Constraints

Fatimah Alghamdi[1], Muhammad Mostafa Monowar[2]
Faculty of Computing and Information Technology at King Abdul Aziz University, Saudi Arabia,
[1]fmalghamdi2@kau.edu.sa
[2]m_monowar@yahoo.com

Abstract: Cloud computing is a promising technology to build the architecture of huge IT systems with the feature of offering its customers with dynamic resources provisioning, according to their fluctuations on demand. A fundamental problem faced by any service provider is how to maximize their profits by allocating resources dynamically among the users' requests and providing differentiated performance levels based on Service Level Agreement (SLA) constraints. In this paper, we present a dynamic resource provisioning model, in which the service provider can satisfy the SLA requirements for different users and maximize his net profits via efficient resource utilization. To achieve the efficiency in resource utilization, we introduce an effective scheduling algorithm for users' requests. This algorithm intends to execute the user request within the processing time limit and to reduce the need to rent more VMs as many as possible. Through simulations using CloudSim tool, we have proved the efficiency of our algorithm in terms of various performance metrics including the total request processing time, virtual resources utilization and the operating profits.

Key words: cloud computing market, dynamic resource provisioning, request scheduling, SLA, resource management, operating profit.

INTRODUCTION

During the last decade, we have noticed significant growth in cloud computing paradigm as a solution to hide IT complexity from end users with lower operating and capital costs. Cloud computing is defined [1] as a model to enable a convenient, on demand network access to a shared set of configurable computing resources in the form of services rapidly provisioned and released with minimal management effort or service provider interaction. Generally, Cloud computing offers three service models which are Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). NIST [1] gave a meaning of provisioning for cloud computing services, which essentially let end users obtain and remove cloud services transparently and automatically.

The cloud environment by its nature is dynamic since the end users can reach and leave the cloud at any time. Furthermore, the resource requirements from the subscribed users are obviously varied along the time which leads to that provisioning and management of resources should be implemented in efficient manner to satisfy these requirements.

Dynamic resource provisioning in the cloud computing has gained extensive attention from the industry and the research community. Many of the recent papers have proposed mechanisms to ensure the quality of service with minimized lease cost in both of customer provided cloud [2] and datacenter-based cloud [3]. Other papers have seek to strike a balance between cost saving for the providers and the performance of the provided services [4] or between the energy saving and the performance in order to minimize the expenditure and the environmental effects [5]. Basic dynamic provisioning techniques focus in providing better resource utilization while guaranteeing the QoS requirements of applications via different proposed strategies [6, 7] without considering the costs incurred by the providers.

By considering the provisioning issue and the market in cloud, we find three typical parties in the cloud [8, 9] as shown in Fig. 1: infrastructure service providers, business service providers (service providers or SaaS providers) and customers (end users). Through these parties, the cloud service provisioning process can be simply done in this sequence: the end user sends service request to the service provider; the service provider receives this request and then orders virtual resources as he demands from the infrastructure service provider; the infrastructure service provider responds to the rental request and then allocates VM instances to the intended service provider for processing the user request. The business in this process is disclosed as follows: Infrastructure service providers charge service providers for renting VM instances to deploy service capacity while service providers charge end users for processing his requests.

Each party in the cloud has its own responsibility and interest [4, 8]. The infrastructure service provider is responsible to improve its physical resources utilization and guarantee their availability in order to maximize its profit. While the service provider aims to improve its virtualized resources utilization to meet its customers' requirements with its business goals and the customer just care about his service and its SLA.

SLA (Service Level Agreement) is one of the most common features in the cloud environment. It is defined [4] as a contract between a service provider and its customers, in which the costumer indicate their required service level or quality of service while the provider is responsible for guarantee them and the transaction between both parties is done according to this agreement. SLA violation is the case of failure to achieve agreed performance by service provider and as a result for this violation the provider will pay a penalty to its customer based on the clause defined in the

ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.4 , No.5, Pages : 34-39 (2015)
Special Issue of ICCTE 2015 - Held during 23 – 24 November, 2015 in Hotel Crowne Plaza, Dubai
http://warse.org/IJSAIT/static/pdf/Issue/iccte2015sp06.pdf

SLA contract. In our paper, we only focus on the interests of cloud service providers and end users.

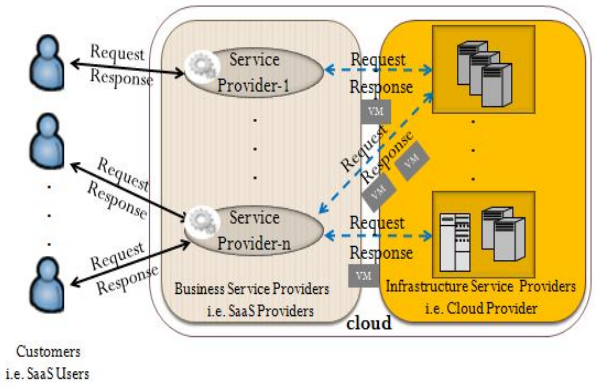

Fig 1: Resource Provisioning Process in Cloud Computing Market

A fundamental problem faced by any service provider is how to maximize their profits by allocating resources dynamically among the users' requests and providing differentiated performance levels based on SLA requirements [10]. In this paper we consider that case of increasing in demand in cloud with loss of efficient resource management. In this case the service provider will rent more and more resources to satisfy the user requirements which results in reducing its profit. We work to find dynamic provisioning mechanism realizes a successful job execution for the user, good profit for the provider and better utilization of the resources. We work to consider these three factors in order to solve the tradeoff between the Quality of Service (QoS) and the resource costs incurred by providers while guarantee the optimum resource utilization.

BACKGROUND

In cloud computing, the cloud providers organize a various types of computing recourses in a pool with different prices and provision theses resources for cloud consumers with the aim of processing their jobs or storing their data.

In general, cloud providers can offer the cloud consumer with two provisioning plans for computing resources, namely (long-term) reservation plan and (short-term) on-demand plan [11]. For instance, Amazon EC2 cloud providers offer both plans but on-demand instances are most popular than reserved instances.

On the reservation plan, we talk about static resource provisioning in which the consumers pay by a one-time fee (ex. for 1 year) before utilizing the resources and under this constraint the providers can be challenged by over-provisioning problem or under-provisioning problem as illustrated in Fig. 2 and Fig. 3 respectively.

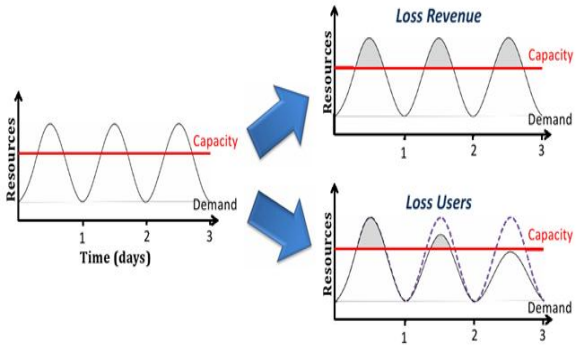

Fig 2: Under-Provisioning Problem

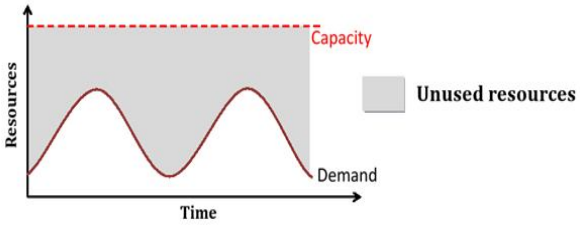

Fig 3: Over-Provisioning Problem

The resource provisioning cost by reservation plan is less than on-demand plan, but in contrast on-demand plan can easier meet uncertainty of consumer's workload and providers' resource prices via dynamic resource provisioning, show Fig. 4. In this plan, the consumer pay per use and the provider provision resource for him at the moment of need with highly scalable and cost-effective manner.

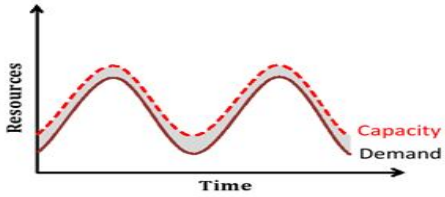

Fig 4: Dynamic Provisioning

So since the resource provisioning has an impact on the service performance as well as requests a significant cost, the service provider should find proper resource provisioning mechanism in which he can maximize his revenue while he provides differentiated performance levels among users.

RELATED WORKS

Recently, many works have discussed different ways to achieve the SLA requirements, while guaranteeing the cost saving or revenue for service providers, using suitable resource provisioning. One of these works [10] addressed the problem of revenue maximization using resource provisioning. It formalized its problem mathematically using Queuing Theory considering proposed pricing model and different QoS parameters such as arrival rate, service rate and resource quantity and from this formulation it derived the overall revenues of service provisions then it resolved its problem using Lagrange Multiplier Method to find how many servers should be assigned for each request to achieve

ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.4 , No.5, Pages : 34-39 (2015)
Special Issue of ICCTE 2015 - Held during 23 – 24 November, 2015 in Hotel Crowne Plaza, Dubai
http://warse.org/IJSAIT/static/pdf/Issue/iccte2015sp06.pdf

maximum revenue under a proposed pricing model. So the resource allocation strategy on this work was based on the pricing model while similar work, was presented in [12], built *Heuristic* and *Greedy* strategies for resource allocation to achieve same objective. Other work [13] has discussed a way of avoiding SLA violation's costs using consumer-centric dynamic provisioning mechanisms. It presented an end-to-end framework that enables the cloud consumer applications to satisfy and manage their SLA performance requirements for the cloud-hosted database tiers then applied adaptive and dynamic provisioning according to these requirements. In contrast, Ran *et al.* [4] have focused on introducing dynamic resource provisioning for service providers to meet their costs saving without violating on SLA requirements. They have applied online unavailability probability estimation model based on the large deviation principle and then they used the estimated unavailability probability as SLA metric to drive the optimal number of VMs for the future workload.

As shown, all the above reviewed works employed SLA and QoS parameters for resource provisioning and ignored considering user's characteristics although they have important effect in profitability of cloud service providers, as appears in [14] and [8]. In [14] some user's characteristics have been employed to present a new proactive resource allocation approach with aim to decrease the impact of SLA violations on customers' satisfaction level. According to [2], there is some tradeoff between the service provider's profit and customer satisfaction by the way of handling the customer satisfaction as variable in SLA.

SYSTEM MODEL

In this paper, we present the dynamic resource provisioning model, in which the service provider (SaaS provider) can satisfy the SLA requirements for different users and maximize his net profits, via efficient resource management.

To achieve the efficiency in resource utilization, we introduce effective scheduling algorithm for users' requests.

This algorithm intends to execute the user request within the processing time limit and to reduce the provider's need to rent more VMs as many as possible.

Our work considers the current workload and differs from others in two aspects:

- We consider VM utilization as an explicit metric when making schedules, while existing algorithms only focus on profit or in QoS.
- Regarding the provisioning of resources we consider the processing time (includes maximum preferred processing time and the deadline) of the service request as a SLA metric which used to determine the best time to provision the service with minimum penalty incurred by service provider. Those two metrics will be explained in detail through the following sections.

Since the profit of cloud service provider depends on the revenue obtained from processing the requests and the cost incurred to rent VMs from IaaS vendors, our objective is help the service provider to save its profit by considering both sides. We intend to minimize the VM rental costs through efficient utilization of existing VMs and to achieve the processing revenue with minimum penalty.

A. System Architecture

System architecture for the dynamic resource provisioning is represented in Fig. 5. For the service provider (SaaS provider), the major components are Receiving unit, Queuing unit, VMs pool and Scheduling unit consists of two vital units which are virtual machine monitoring (VMM) and virtual machine provisioning (VMP).

- Receiving unit receives user request then it sends this request to Queuing unit and notifies Scheduling unit about the received request.
- Scheduling unit is to execute the VM allocation strategy for the user requests. It employs VMM unit to collect state information of the rented VMs in the VM pool, and VMP unit to send rental request for cloud providers (IaaS providers) as needed and then insatiate the provisioned VMs in the VM pool.

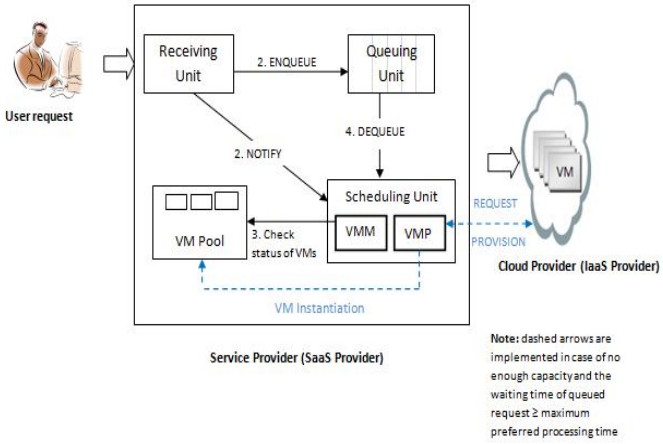

Fig 5: System Architecture for Dynamic Resource Provisioning

In this paper, the main work is to propose efficient scheduling algorithm for users' requests implemented in Scheduling unit. In the subsequent section, we describe our algorithm and explain the steps that are figured in the architecture (Fig. 5) after formulation the service request according to its contribution in our work.

B. User Service Request Formulation

As we mentioned before, SLA is a contract, between a service provider and its customer, holds the agreed service performance metrics and the corresponding costs as well as the penalties for non performance. In our work, we assume that SLA mainly considers the processing time of the request as a performance metric for all services of end users.
According to that assumption, we quantified some SLA constraints related to provider's profit and then formulate the user service request as following:

User Req. = (service cost, *mppt*, deadline, penalty rate)

- **service cost**: the amount of money that the user pays to the service provider for processing its request.
- ***mppt***: the maximum preferred processing time for the request. If the request is processed before the end of

ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.4 , No.5, Pages : 34-39 (2015)
Special Issue of ICCTE 2015 - Held during 23 – 24 November, 2015 in Hotel Crowne Plaza, Dubai
http://warse.org/IJSAIT/static/pdf/Issue/iccte2015sp06.pdf

this time then the provider will not pay any penalty. Otherwise, the provider will lose some of his revenue for the delay.

Revenue per charged service = service cost – penalty value

- **deadline**: the upper limit of the processing time. If the request processing is not completed before this limit then the provider will pay penalty for SLA violation without get any revenue.

Revenue per charged service = 0

Figure 6 represents the relation between the processing time of the request and the provider's revenue per processing it.

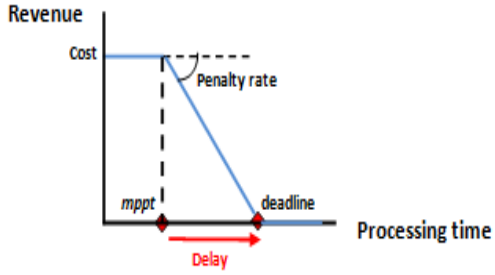

Fig 6: Service Request Model with SLA Constraints

- **penalty rate**: it is used to compute the penalty value that incurred by provider in case of failure to meet request preferred processing time (*mppt*) by this formula:

Penalty value= *Delay* * penalty rate

Regarding ***Delay*** quantity in this formula, it is computed as following:

Delay= T - mppt, where T is the actual request processing time.

C. Service Request Scheduling Algorithm

In this section, we illustrate our approach for scheduling the user requests and provisioning virtual machine on demand.

The basis of our Economy-Aware Service Request Scheduling (**ESRS**) Algorithm is dividing the user request into independent several subtasks processed in parallel on the all existing VM instances. Employing the divisibility feature of request by this way will guarantee no idle VM instance in the VM pool which results on optimizing the resource utilization.

Firstly, we assume the service provider has ***N*** VMs in the VM pool, and he will not rent more VMs from IaaS vendor unless there is no enough capacity to process the accepted request before the deadline limit **and** this request is queued for more than its maximum preferred processing time, as we have mentioned in Fig. 5.

Our scheduling algorithm is implemented in Scheduling unit through the following steps:

1- It receives notification about the incoming queued request in the form of the proposed user request formula to know the maximum preferred processing time (*mppt*) and the deadline limit for processing this request.
2- Through VMM unit, it checks if there is enough capacity to process this request before the deadline limit. If there is enough capacity, go to 3 otherwise go to 4. To know if there is enough capacity with this constraint, it experimentally divides the request into ***N*** subtasks and computes the required time to execute each subtask on the associated VM. For simplicity, we assume that rented VM instances are standard instances and the time needed for processing particular request on it is T_s. Then T_s/N represents the processing time of each subtask.
3- It gets the request from the queue, divides it into ***N*** subtasks and then allocates these subtasks to VM instances for parallel processing.
4- It checks the waiting time of the queued request. If the waiting time is greater than or equal the maximum preferred processing time (*mppt*) of this request, go to (5) otherwise go to (6).
5- Through VMP unit, it sends rental request to IaaS vendor for provisioning more VM and then insatiate the provisioned VM in the VM pool, then go to step 3.
6- The request can be waited in the queue until there is enough capacity, go to step 2.

Figure 7 displays the procedure of the proposed algorithm. In the following section, we evaluate the performance of this procedure.

PERFORMANCE EVOLUTION

To prove the efficiency of our proposed **ESRS** algorithm, we conduct some simulation experiments using CloudSim tool. CloudSim is the toolkit enables the developers for modeling and simulating the cloud computing environments and evaluating the performance of their own provisioning policies [15]. In this section, we present the experiment with its results and then evaluate theses results according to particular performance metrics.

A. Simulation Experiment

In our experiment, we extend some existing classes included in the simulator to create 100 VM instances with same characteristics and 200 service requests with different SLA requirements (i.e. service cost, maximum preferred processing time- *mppt*, deadline, and penalty rate). Every request is divided into 100 subtasks with same parameters (i.e. length, file size, output size, etc.). We modify the scheduler policy to dispatch those subtasks to VM instances for parallel processing so the request can be completed within the deadline specified. All the simulation experiments are conducted on the same computer with Windows 7 Home Premium SP1, its processor is Intel Core5 CPU 2.27 GHz and its RAM capacity is 4.0 GB. The simulation program is CloudSim 3.0.3 with Java codes based on eclipse-jee-lun-SR1-win32-x86_64 with JDK 1.8.0_25 as runtime environment.

ISSN 2278-3083

International Journal of Science and Applied Information Technology (IJSAIT), Vol.4 , No.5, Pages : 34-39 (2015)
Special Issue of ICCTE 2015 - Held during 23 – 24 November, 2015 in Hotel Crowne Plaza, Dubai
http://warse.org/IJSAIT/static/pdf/Issue/iccte2015sp06.pdf

B. Results Evaluation

We consider the following performance metrics to evaluate our **ESRS** algorithm:

- **Request processing time**: represents the total execution time of the request's subtasks.
- **Number of idle VMs**: represent those VMs which are rented from IaaS vendors and instantiated in VM pool but not used for processing any request and they are unexpired. This metric is used to evaluate the resource utilization.
- **Operating profit**: represents the total profit that service provider gets from providing the cloud services. It can be calculated according to this formula:

Operational Profit= revenue of processing request – resource rental cost

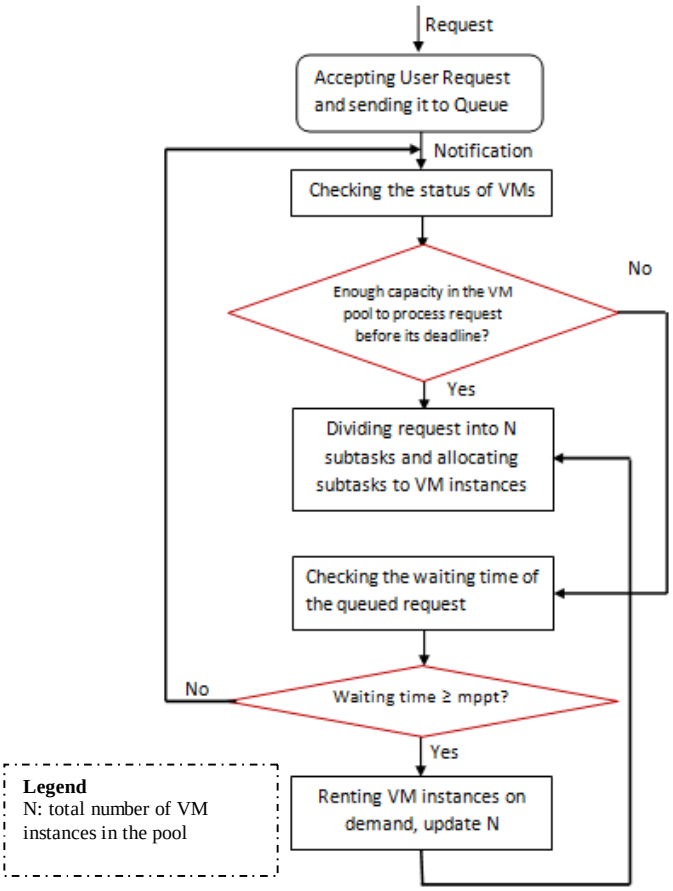

Fig 7: Flowchart of Proposed Scheduling Algorithm

And as we mentioned before, the revenue for processing request has three different cases according to the time spent to execute this request.

The following figures represent the results of the experiments according to those performance metrics. They show that our **ESRS** algorithm provides great results among all the considered metrics.

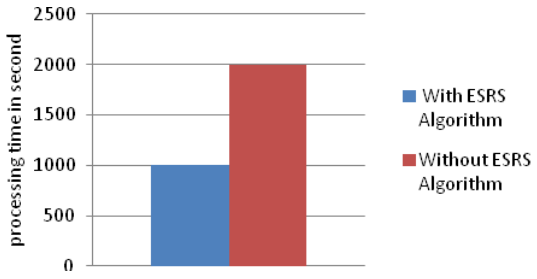

Fig 8: Request Processing Time

According to Fig. 8, the time needed to process the single request with ESRS algorithm is less than the time needed without applying ESRS algorithm. This difference appears as a result of allocating the subtasks among all VM instances for parallel processing, so each VM instance processes single subtask without consuming more time for movement between two or more subtasks, according to scheduler time-shared policy.

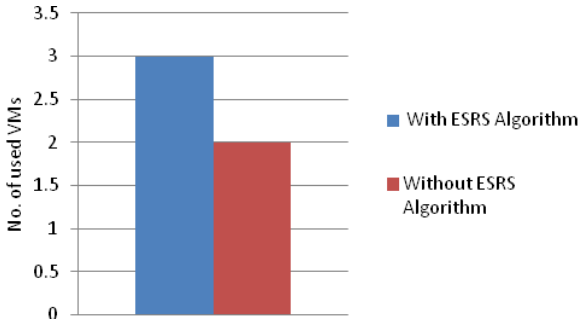

Fig 9:VM Utilization

According to Fig. 9, number of VM instances used to process the request with ESRS algorithm is greater than the number of VM instances used without applying ESRS algorithm. This difference appears as a result of dispatching the subtasks among all VM instances, so no idle VM instance in the VM pool.

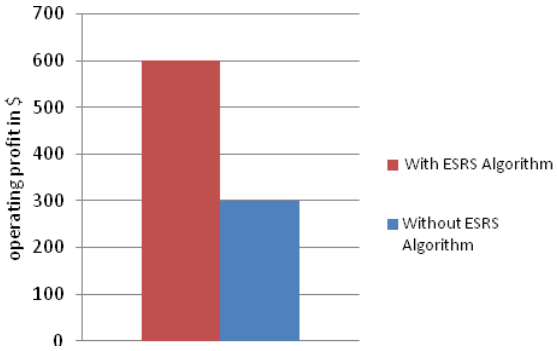

Fig 10: Operating Profit

According to Fig. 10, the operating profit obtained from processing all the requests with ESRS algorithm is greater than the profit obtained without applying ESRS algorithm. This difference appears as a result of above mentioned results

of processing time and VM utilization. Regarding the processing time, while the request can be processed within the processing time limit (i.e. before deadline) then the provider can get the revenue of processing with minimum SLA violation penalty. Regarding the VM utilization, while the scheduling algorithm guarantee the maximum utilization of existing VMs then the provider can reduce the cost incurred to rent VMs from IaaS vendors. Based on operational profit formula, achieving both goals will obviously increase the operating profit.

CONCLUSIONS AND FUTURE WORK

Cloud computing has become a widely technology among all types of modern utility services. In this paper, we have addressed the problem of dynamic resource provisioning for the trade-off between the Quality of Service (QoS) and the resource costs incurred by service providers in cloud environment. We have presented a dynamic resource provisioning model, in which the service provider (SaaS provider) can satisfy the SLA requirements for different users as well as maximize his net profits, via efficient resource utilization.

Our work opens the way for a number of research lines in the Cloud economy field. Through simulations using CloudSim tool, we have proved the efficiency of our **ESRS** algorithm in terms of various performance metrics including the total request processing time, virtual resources utilization and the operating profits. In the future, we intend to simulate our algorithm to show its efficiency in contrast with other cost-effective scheduling algorithms. In addition, we plan to extend this work to investigate our algorithm by considering the user satisfaction as an important factor in the profitability of cloud service providers.

ACKNOWLEDGMENT

We would like to thank our supervisor Dr. Muhammad Mostafa Monowar for his valuable comments and suggestions that helped improving this works. His effort is greatly appreciated.

REFERENCES

[1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, National Institute of Standards and Technology(NIST),U.S. Department of Commerce, Gaithersburg, MD, Sep. 2011, [Online]. Available: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf [Accessed: 19 Nov.2014].

[2] H. Wang, F.Wang ; J.Liu, K.Xu, D.Wu and Q.Lin, "Resource provisioning on customer-provided clouds: Optimization of service availability," in *Proc. 2013 IEEE International Conference on Communications (ICC)*, Budapest, 2013, pp. 2954 – 2958.

[3] S. Vijayakumar, Q. Zhu, and G. Agrawal, "Dynamic Resource Provisioning for Data Streaming Applications in a Cloud Environment," in *Proc. 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom)*, 2010, pp. 441–448.

[4] Y. Ran, J. Yang, S. Zhang, and H. Xi, "SLA-driven dynamic resource provisioning for service provider in cloud computing," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Atlanta, GA, 2013, pp. 408–413.

[5] Q. Zhang, M. Zhani, R. Boutaba, and J. L. Hellerstein, "HARMONY: Dynamic Heterogeneity−Aware Resource Provisioning in the Cloud," in *Proc. 2013 IEEE 33rd International Conference on Distributed Computing Systems*, Philadelphia, PA, 2013, pp. 510 – 519.

[6] C. S. Pawar and R. B. Wagh, "Priority Based Dynamic resource allocation in Cloud computing with modified waiting queue," in *Proc. 2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*, 2013, pp.311-316.

[7] D. Tammaro, E. A. Doumith, S. A. Zahr, J.-P. Smets, and M. Gagnaire, "Dynamic Resource Allocation in Cloud Environment Under Time-variant Job Requests," in *Proc. 2011 IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom)*,2011, pp. 592–598.

[8] J. Chen, C. Wang, B. B. Zhou, L. Sun, Y. C. Lee, and A. Y. Zomaya, "Tradeoffs between profit and customer satisfaction for service provisioning in the cloud," in *Proc. 20th international symposium on High performance distributed computing*, 2011, pp. 229–238.

[9] Z. Liu, Q. Sun, S. Wang, H. Zou, and F. Yang, "Profit-driven Cloud Service Request Scheduling Under SLA Constraints," *Journal of Information & Computational Science*, vol. 9, no. 14, pp. 4065–4073, 2012.

[10] G. Feng, S. Garg, R. Buyya, and W. Li, "Revenue Maximization Using Adaptive Resource Provisioning in Cloud Computing Environments," in *Proc. ACM/IEEE 13th International Conference on Grid Computing (GRID)*, Beijing, Sept.2012, pp. 192–200.

[11] S. Chaisiri, B.-S. Lee, and D. Niyato, "Optimization of Resource Provisioning Cost in Cloud Computing," *IEEE Transactions on Services Computing*, vol. 5, no. 2, pp. 164–177, Apr. 2012.

[12] M. Mazzucco, "Revenue maximization problems in commercial data centers," PhD Thesis, Newcastle University, 2009.

[13] L. Zhao, S. Sakr, and A. Liu, "A Framework for Consumer-Centric SLA Management of Cloud-Hosted Databases," *IEEE Transactions on Services Computing*, vol. PP, no. 99, 2013, pp.1-14.

[14] H. Morshedlou and M. R. Meybodi, "Decreasing Impact of SLA Violations: A Proactive Resource Allocation Approach for Cloud Computing Environments," *IEEE Trans. Cloud Comput.*, vol. 2, no. 2, pp. 156–167, Apr. 2014.

[15] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R. Buyya, "CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23-50, Jan. 2011.