Journal of Computer Science 10 (12): 2628-2635, 2014
ISSN: 1549-3636

doi:10.3844/jcssp.2014.2628.2635 Published Online 10 (12) 2014 (http://www.thescipub.com/jcs.toc)

MULTI-LEVEL PRIORITY-BASED SCHEDULING MODEL IN HETEROGENEOUS CLOUD

Siti Fajar Jalal and Masnida Hussin

Department of Communication Technology and Networks,
Faculty of Computer Science and Information Technology,
University Putra Malaysia, Selangor, Malaysia

Received 2014-08-26; Revised 2014-08-28; Accepted 2014-11-24

ABSTRACT

Cloud has emerged into another enhancement that manipulates the diversity of resources in order to expand the ability of Cloud facility. Cloud is no longer being seen as one-type service provider. The advantages of the heterogeneous Cloud give great benefits to the users and have business potential in service market. To cope with the dynamic nature of heterogeneous Cloud, the Cloud provider needs to have strategies to efficiently allocate tasks to the resources. Also, to charge the services is another challenge to the Cloud provider as the resources in the Cloud system are heterogeneous. In this study, we suggest the implementation of a. Multi-level priority-based scheduling and dynamic pricing into the heterogeneous Cloud model. We perform an extensive performance evaluation on the model through simulations. We define the attribute of the Cloud simulation as dynamic and random to address the heterogeneous feature of the Cloud. Our simulation result shows that the multi-level priority-based is significantly increasing the resource utilization rate and its integration with the dynamic pricing successfully improves the performance of the Cloud service in term of satisfaction rate.

Keywords: Task Scheduling, Satisfaction Rate, Utilization Rate, Heterogeneous Cloud

1. INTRODUCTION

Cloud computing is defined as a model to allow ubiquitous, convenient, on-demand network access to a shared pool configurable computing resources in which with minimal management effort or service provider interaction (Mell and Grance, 2011). The Cloud facility has created a new trend of modern information system. With the rapid development of today's technologies, many business organizations and even non profit parties have shifted to utilize the Cloud computing. Basically in Cloud computing, the resources can be rapidly provisioned and released. Users subscribe and pay to Cloud providers in order to acquire computing resources that provided by the Cloud provider. Clouds are served in several categories; which are in term of software, platform or infrastructure where each of the type of service has its own service model and the service models are Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) (Mell and Grance, 2011; Liu *et al*., 2011). There are few types of Cloud pricing schemes that can be used by the Cloud provider to charge the services subscribed by the users.

Pricing scheme is used by the Cloud provider to charge the services that are used by the user and then produce billing. Different pricing scheme can be used to charge different types of service (Choi and Hong, 2007). For example, the SaaS users can be charged based on

Corresponding Author: Siti Fajar Jalal, Department of Communication Technology and Networks, Faculty of Computer Science and Information Technology, University Putra Malaysia, Selangor, Malaysia

the network bandwidth consumed or amount of data stored; the PaaS users can be billed based on the processing resources consumed or duration of the platform used; and the IaaS users can be charged according to volume and duration of data stored or CPU hours used (Liu *et al*., 2011). A more detailed discussion regarding the Cloud pricing scheme is discussed in a study by Ruiz-Agundez *et al*. (2011).

Static pricing model is still dominantly being used today in Cloud services (Xu and Li, 2013) where the user pays the Cloud service according to the usage of the resources the user used (pay-per-use basis). Windows Azure, Google Cloud and Amazon use pay-per-use pricing approach. However, there are several disadvantages of implementing the static pricing highlighted in a study by Shang *et al*. (2010). The static pricing may result to the waste of resource if the user only requires to run the application once a month for hours and in some scenarios, the fixed rate pricing model can get expensive. If the Cloud uses pre-pay method, it will cause the user to be locked to certain providers in a range of time where we might consider that probably, there are better Cloud services with significant prices provided by other Cloud providers.

As a result, many studies are comprehensively done to propose new dynamic pricing models. Besides, the existing Cloud provider has started implementing the dynamic pricing model, giving more billing options to the users. For example, the Amazon EC2 has introduced the spot instances which the users are able to bid on the unused Amazon EC2 resource (AWS, 2014). Dynamic pricing has benefits over the static pricing. This approach facilitates the Cloud providers to supply a range of resource types while the users can request for custom configuration with multiple resource types (Teng and Magoulès, 2010). Furthermore, from the aspect of economy, this pricing technique is able to handle the scenario when the supply and demand fluctuates. The Cloud provider will also be able to cope with unpredictable user demand and at the same time, maximizes the revenue (Xu and Li, 2013; Tsai and Qi, 2012). Therefore, it is important for us to consider the dynamic pricing and include it into the model to study its effect on the Cloud satisfaction rate.

Additionally, there are several business-related services that are entailed to Business Support in Cloud field. The business-related services include the customer management, contract management, inventory management, accounting and billing, reporting and auditing and, pricing and rating (Liu *et al*., 2011). Our scope of interest focuses on the pricing and rating of Cloud services. The pricing and rating services emphasize more on determining Cloud service prices and evaluating the Cloud services that are leased to users.

The performance of Cloud resources might be differed to each other (Schad *et al*., 2010; Armbrust *et al*., 2009). Thus, it is important to wisely assign task to suitable resource in order to achieve high utilization and the agreed Service Level Agreement (SLA). Good performance of services gives good satisfaction rate, but however, the price of the service is also one issue to be considered as well. Customer satisfaction is important to ensure a provider sustain in the Cloud business with profit return (Chen *et al*., 2011). Although the aim of most cloud providers is to gain maximum profit in their business, but this will result the increase of costs that the users have to spend for the service. Therefore, optimum prices for certain level of resource performance should be identified to maintain the business, in which this will not be focused in this study.

In our study, we study the satisfaction rate of Cloud service that uses double schedulers to distribute tasks to the available resources in a heterogeneous environment. We are basically expanding a study of a proposed idea about a scheduling technique in an existing study by Hussin *et al*. (2011) into a Cloud implementation. The multi-level scheduling technique has been proven to be able to improve the processing time in a complete heterogeneous environment. Thus, by implementing it into the Cloud system environment, we study how much satisfying the performance of the model with and without local schedulers. Our formula to calculate the satisfaction rate used in this study takes into account the discrepancy of prices calculated upon certain Cloud service performance level. We perform the performance evaluation by using simulation developed using the C++ programming language. Real workload is used as the tasks to be submitted by the user into the Cloud.

Firstly, we verify that our simulation has competitive performance by evaluating the total service time required to process certain number of tasks. To show that the multi-level model can provide user satisfaction, we compare the model with a uni-level model (i.e., the local scheduler is excluded) in term of satisfaction rate. Later, we select another parameter to compare both of the two models in which the utilization rate.

Our paper is organized as follows: Section 2 discuss about several related works to our study. Section 3

includes the model description that is used as our system environment. Next, we describe the performance evaluation and the discussions of results in Section 4. Section 5represents the conclusion of our paper work.

2. RELATED WORK

There are several works which are focusing on resource pricing and allocation on Cloud. A Double Auction Bayesian Game-Based Pricing Model is introduced in a study by Shang *et al*. (2010). This model allows the Cloud users to utilize the idle resources in more flexible way. The Cloud providers and buyers can decide whether to exchange user requirements of resources even though they may not know each other but this, however, will lead to truthfulness issue (Samimi *et al*., 2014). Meanwhile, Pal and Hui (2013) proposed the Cloud economic model that allows the Cloud provider to know what prices and QoS level to set for the end-users, so that the provider could exist in the Cloud market. The drawback of this study is that the authors focus more on how to maximize the revenue, in which we would like to highlight here that cloud cooperation is not only about the profit growth (Zhuang, 2009).

With today's technologies, Cloud starts to implement heterogeneous type of resources in providing their services. The discussed studies, however, focused more into pricing and allocating one type of services to the clients and paid less attention upon this resource heterogeneous criterion. Thus, in order to map this issue, in our study, we apply the heterogeneity of Cloud resources into the system architecture.

Teng and Magoulès (2010) presented a resource pricing and equilibrium allocation policy based on the consideration of Cloud users' competition for limited resources with different financial capacities. Users will be able to predict resource price based on the task size, priority and QoS requirement, as well as satisfy budget and deadline constraint, which is similar to study performed by Mihailescu and Teo (2010). However, the study paid fewer attentions to the strategy of providers where the attention was given more into the user's perspective. In addition, this study involves the users that were associating with a single resource site. In response to these, we address the matters by focusing on the strategy from the providers' perspective and including the dynamic numbers of resource sites into our system architecture.

3. THE MODEL

We employed the Cloud system environment (**Fig. 1**) for dealing with heterogeneous resources in the Cloud. Heterogeneous nature of the Cloud is an important aspect to be considered for providing plentiful benefits to the users. Specifically, the heterogeneous attribute in Cloud allows specialized devices to be efficiently optimized, provides dynamic provisioning, economies of scale ad comparatively lower capital expenditures (Crago *et al*., 2011).

We develop the Cloud model with random number of resource sites and processors to demonstrate the heterogeneous feature in the system. In addition, every available processor located in the sites also has various numbers of cores which is set randomly with mutual speed within it. Thus, by this, every of the resource site gives different performance and ability in processing specific tasks. A strategy is important to assign any task to the most suitable resources in order to maximize the resource utilization, reduce the processing time and accelerate the overall cloud performance.

User sends tasks into the Cloud through an interface. The interface allows the user to communicate with the Cloud in the existence of a broker that acts as an intermediate party between the user and the Cloud resources. A broker is responsible to receive the tasks submitted by the user and forward the tasks to the appropriate site of resources. For the sake of simplicity, the earliest tasks arrive at broker are the earliest tasks to be released by the broker (i.e., First in First Out). Every task is assigned with priority. Hence, when it reaches the broker, the priority of each task is calculated before it is scheduled into the resource site. This scheduling approach is aiming for reliable task execution (i.e., meeting deadlines for high priority tasks) besides maximizing the resource utilization (Hussin *et al*., 2011).

Once the task reaches the resource site, a Local Scheduler (LS) then receives the task and scheduled the task into the waiting queue. The LS has full knowledge about the available resources in its site in which the broker has not. Based on the priority defined by the broker, LS assigns the task to the resources according to two different policies. The scheduling policies are as below (Hussin *et al*., 2011):

- Policy 1: LS assigns task to processor based on the processing capacity.
- Policy 2: LS simply and randomly assigns task to any unoccupied processor.

The LS uses policy 1 to assign tasks with high priority to the resources while applying the policy 2 the tasks with low priority. Meanwhile, the tasks with medium priority is assigned to the resources according to the status of waiting time, if the waiting time is continuously increasing, the LS then uses the policy 2 or otherwise, it uses policy 1.

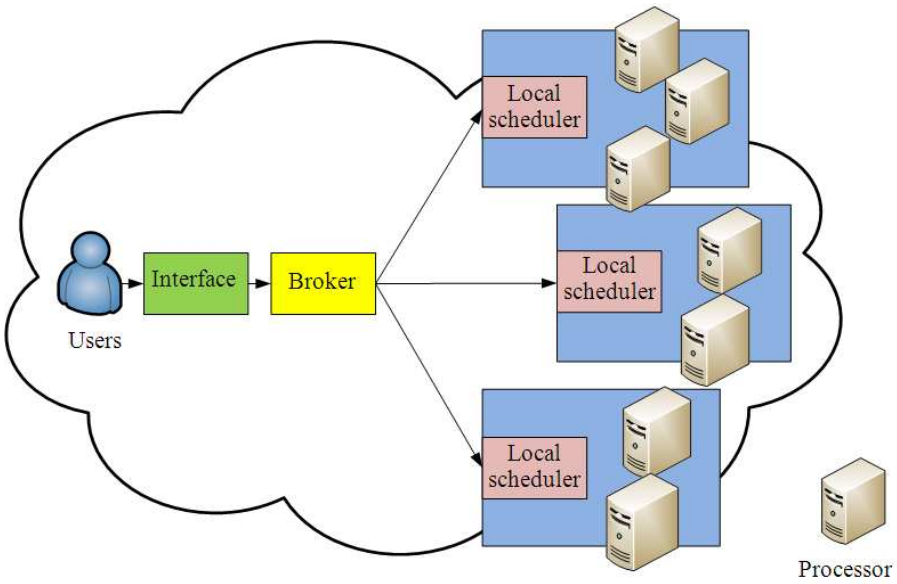

Fig. 1. Cloud model

4. PERFORMANCE EVALUATION AND DISCUSSION

In this section, we discuss the detail of the simulation configuration and setup. To construct the performance evaluation, we perform a simulation of the Cloud system environment and used real workload trace (Feitelson, 2005) as the users' tasks.

4.1. Simulation Setup

We build the simulation Cloud, *Double Selection* simulation by using C++ programming language. **Table 1** shows the simulation parameters and their values that we applied into the simulation.

In our simulation program, we define the number of resource sites, number of processors and number of cores in random between specified ranges as mentioned in **Table 1** in order to map the heterogeneous characteristic of Cloud services. The rand () function is used to define the speed of core in the range of 50 to 100 MIPS where the speed of cores under similar processor is the identical.

The San Diego Supercomputer Center (SDSC) Blue Horizon real workload log is used in this simulation. This workload was logged from April 2000 until January 2003 with involving 250 400 tasks in it (Feitelson, 2005). However, in our simulation, we only use up to 1000 tasks from the workload log. About 144 nodes were involved in the system including 8 processors for each node.

The inter arrival between tasks is defined to be in random manner, the entrance of tasks into the system is in Poisson distribution. The simulation model is heterogeneous with various number of resource sites, number of processors in every sites and number of cores in every processors on every run. Thus, we carry out the experiments in five cycles to obtain results in different capacity of resources. During every turn of experiments, we collect the service time and the satisfaction rate of the system performance. Later, the final result is obtained by averaging the outcomes gathered from each experiment.

One common factor affecting the service satisfaction rate is the price. There are two prices that we calculate in this study: The actual service price *act-price* and the service price ρ. We use Equation 1 to calculate the actual service price that comprises the price we obtain by considering the values in the original workload. The Equation 1 is as below:

$$act\_price = \frac{Total\ number\ of\ jobs}{\sum actual\ service\ time} \quad (1)$$

Meanwhile, the service price is the price that we calculate with the values of the service time obtained through our simulation. The Equation 2 used to calculate the service price is as shown:

$$\rho = \frac{Total\ number\ of\ jobs}{\sum service\ time} \quad (2)$$

The *act-price* is usually smaller than ρ, rationally due to the performance of the original workload is lesser compared to our simulation. To address this variance, we will then deduct *act-price* from ρ to identify the difference of the prices *d*.

We compute the satisfaction rate on every turn of experiments by using the Equation 3 below:

$$Satisfaction\ Rate = \frac{\rho + \delta}{\rho + A} - d \quad (3)$$

The satisfaction rate takes into account a price ρ, the difference of prices *d*, actual service time *A* and also the difference of service time between the workload log and *Double Selection* simulation, which is δ. We compare the values with the difference of service time δ and service price with the significant value of the original actual service time *A* along with the price ρ. This is to quantify the improvement of the Cloud performance in *Double Selection* simulation compared to the original workload. However, as the service time decreases in *Double Selection* simulation, it leads to higher service prices. Therefore, we take into consideration of this issue by deducting the significant difference of the price unit between the *Double Selection* simulation and the original workload which is *d*.

Equation 4 is used to simply calculate another parameter which is the utilization rate. The Equation 4 is as below:

$$UtilizationRate = \sum \frac{exe\_time}{exe\_time + wait\_time} \quad (4)$$

The *exe-time* is the execution time of all tasks meanwhile the *wait-time* is the total waiting time of all tasks in queues before being processed by resources.

5. RESULTS AND DISCUSSION

In this section, we present the results that obtained through the experiments. Firstly, we study the total service time of different number of tasks submitted by the user between the simulation and the original workload. The average service times are compared to verify the performance of our scheduling approach in the aspect of service time. After that, we continue by studying the satisfaction rate of the services on a range of number of tasks. We consider the service time and price as factors affecting the satisfaction rate. To identify the effectiveness of the LS existence in the system model, we compare two conditions of model: With and without the LS in two parameters. The parameters are satisfaction rate in and utilization rate.

5.1. Result 1: Average Service Time

Figure 2 describes the comparison of average service times of 200, 400, 600, 800 and 1000 numbers of tasks. The service time comprises the total time starting from the time when task is being submitted by the user until the time when the task exits the system environment. It shows in **Fig. 2** that, the average service time are increases proportionally with the number of total tasks. The result proves that *Double Selection* simulation gives better performance in term of service time.

In overall, *Double Selection* simulation scheduling approach improves about 30% of the average service time compared to the original service times of the workload. That is due to lower waiting time consumed by each submitted task before being processed by the available resources in *Double Selection* simulation. The waiting time is lowered when the Broker responsibilities to distribute the tasks to the suitable resources is reduced with the existence of the LS in the system. Thus, with this, less time is required for the Cloud to finish processing the task.

5.2. Result 2: The Cloud Service Satisfaction Rate With and Without the Implementation of LS

Figure 3 shows that the average of satisfaction rates with and without the existence of LS in resource sites. The results illustrates that the satisfaction rates for both conditions decelerate as the number of tasks submitted into the model increases. We consider the improvement of performance in *Double Selection* simulation can provide satisfying service to the user. From **Fig. 3**, we can see that the satisfaction rate is highest when the number of tasks is 200, with or without the LS. When the number of tasks is lower, the load of tasks in the queue is less crowded and the waiting time before the task is being processed is lower too, thus, the complete time of each task is faster, making the service time smaller as well as the price. However, as the number of the tasks increases by 200, the satisfaction rate also decreases by about 15% in average at the same time. This happens

because when the number of tasks is higher in the model, the processing rate is slower and the service price is higher. However, our experiment is consistently showing that the existence of LS gives higher satisfaction rate compared to the condition without the LS.

5.3. Result 3: The Resource Utilization Rate With and Without the Implementation of LS

We illustrate the performance of the Cloud in term of utilization rate with and without the presence of LS in **Fig. 4**. The graph shows that the average of utilization rates with and without the existence of LS in resource sites varies with the number of tasks submitted. Evidently, the utilization rate falls when the number of tasks is 400 for both with and without LS model. Nevertheless, the utilization rate with the LS remains undeniably superior. This is mainly because of the ability of the LS to evaluate every processor before assigning the task to the most suitable resources. LS provides better task distribution and reduce the burden of the broker.

Without the LS, the broker requires more time to identify which is the most capable resources to process the task. This increases the waiting time of each task in submitted into the cloud. With the existence of the LS in the model, the waiting time is reduced, allowing the Cloud to finish processing faster.

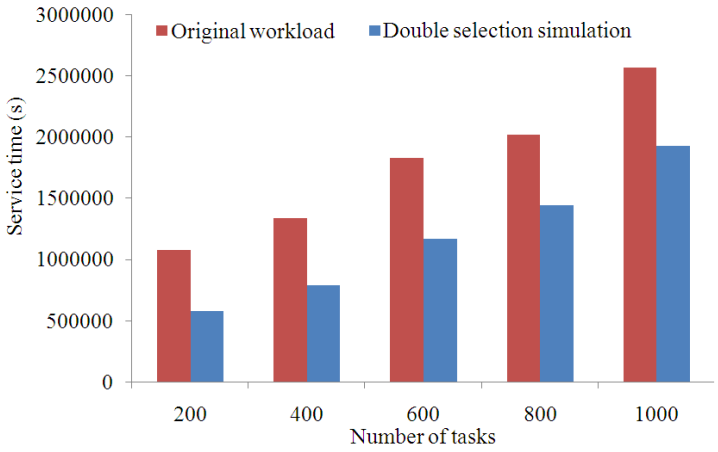

Fig. 2. Comparison of the average service time (s) obtained from our simulation and original workload

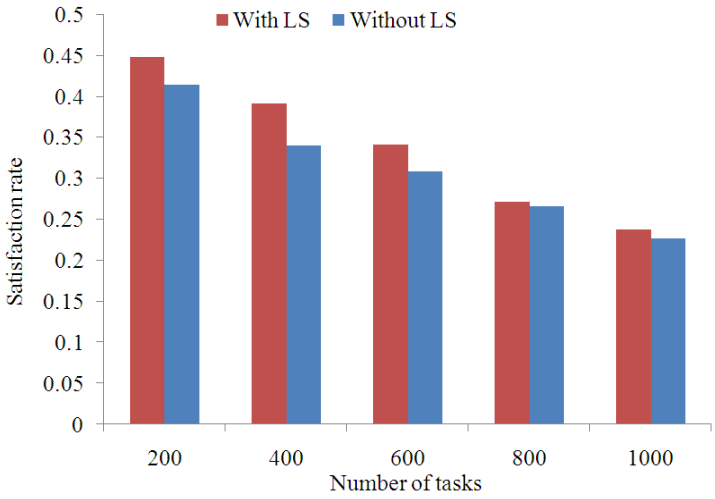

Fig. 3. Average satisfaction rate with-and without-LS

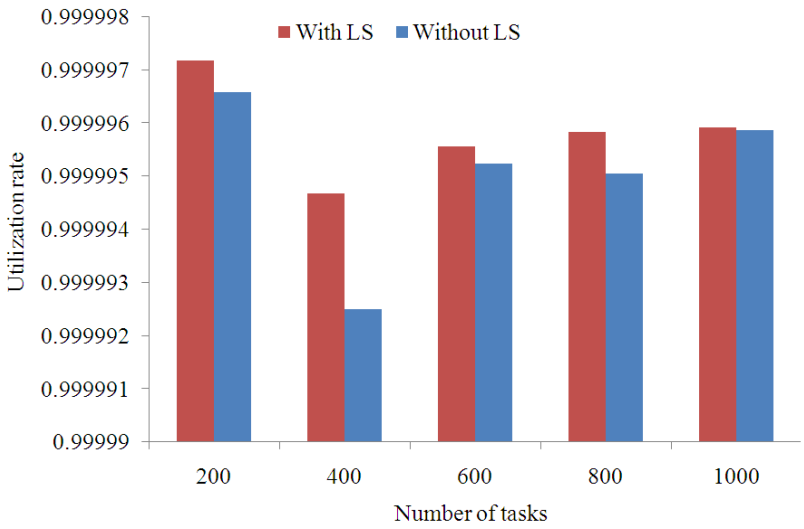

Fig. 4. Average utilization rate with-and without-LS

Table 1. Simulation parameters

Parameter	Description
Workload trace	SDSC blue horizon
Resource sites	4 to 8
Number of processors	8 to 20
Number of cores	2 to 8
Core speed of a processor	50 to 100 MIPS
Number of tasks	200, 400, 600, 800 and 1000 tasks
Programming language	C++
Inter arrival time	Poisson distribution

6. CONCLUSION

Cloud computing creates a new trend in recent technologies for individuals or organizations, whether it is for commercial or personal benefit. The utilization of Cloud facilities is increased with the implementation of the idea of heterogeneous features into the Cloud. In fact, this technology is shifting to dynamic pricing due to its efficiencies in various aspects that were found in several studies. With the ability to allocate the tasks to suitable heterogeneous resources, we implement the model and scheduling approach suggested by Hussin *et al*. (2011) into a Cloud environment. From the extensive simulation, we prove that the integration of the multi-level scheduling model and dynamic pricing gives positive influences and significant performance in two parameter aspects of satisfaction rate and utilization rate. We successfully emphasize that effective scheduling and pricing techniques is important for the performance of heterogeneous Cloud. In future, we are aiming to extend this study by considering additional aspects for computing the satisfaction rate such as SLA.

7. ACKNOWLEDGEMENTS

The authors appreciatively acknowledge the support of Ministry of Education Malaysia for funding the present study.

8. ADDITIONAL INFORMATION

8.1. Funding Information

This study manuscript has been funded by the Ministry of Education Malaysia Grant 08-02-13-1363FR.

8.2. Author's Contributions

Siti Fajar Jalal: Author contributed in planning, development of study, analysis and preparing the manuscript.

Masnida Hussin: Author proposed the idea of the study and coordinated the study.

8.3. Ethics

The idea of this work is originally from the authors and to the extend of authors' knowledge, no similar study has been done in previous works

9. REFERENCE

Armbrust, M., A. Fox, R. Griffith, A.D. Joseph and M. Zaharia *et al*., 2009. Above the clouds: A berkeley view of cloud computing. EECS Department, University of California.

AWS, 2014. Amazon Elastic Compute Cloud (Amazon EC2). Amazon Web Service.

Chen, J., C. Wang, B. Zhou, L. Sun and A.Y. Zomaya *et al*., 2011. Tradeoffs between profit and customer satisfaction for service provisioning in the cloud. Proceedings of the 20th International Symposium on High performance Distributed Computing, Jun. 8-11, ACM Press, New York, USA, pp: 229-238. DOI: 10.1145/1996130.1996161

Choi, M.J. and J.W.K. Hong, 2007. Towards management of next generation networks. IEICE Trans. Commun., 90: 3004-3014. DOI: 10.1093/ietcom/e90-b.11.3004

Crago, S., K. Dunn, P. Eads, L. Hochstein and J.P. Walters *et al*., 2011. Heterogeneous Cloud Computing. Proceedings of the IEEE International Conference on Cluster Computing, Sept. 26-30, IEEE Xplore Press, Austin, TX, pp: 378-385. DOI: 10.1109/CLUSTER.2011.49

Feitelson, D., 2005. Parallel Workloads Archive. The Rachel and Selim Benin School of Computer Science and Engineering, Hebrew University of Jerusalem, Israel.

Hussin, M., Y.C. Lee and A.Y. Zomaya, 2011. Priority-based scheduling for large-scale distribute systems with energy awareness. Proceedings of the IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Dec. 12-14, IEEE Xplore Press, Sydney, NSW, pp: 503-509. DOI: 10.1109/DASC.2011.96

Liu, F., J. Tong, J. Mao, R. Bohn and D. Leaf *et al*., 2011. NIST cloud computing reference architecture. National Institute of Standards and Technology, Gaithersburg, Maryland.

Mell, P. and T. Grance, 2011. The NIST definition of cloud computing. Special Publication 800-145 ed. National Institute of Standards and Technology, Garthersburg, Maryland.

Mihailescu, M. and Y.M. Teo, 2010. On economic and computational-efficient resource pricing in large distributed systems. Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, May 17-20, IEEE Xplore Press, Melbourne, Australia, pp: 838-843. DOI: 10.1109/CCGRID.2010.124

Pal, R. and P. Hui, 2013. Economic models for cloud service markets: Pricing and capacity planning. Theoretical Comput. Sci., 496: 113-124. DOI: 10.1016/j.tcs.2012.11.001

Ruiz-Agundez, I., Y.K. Penya and P.G. Bringas, 2011. A flexible accounting model for cloud computing. Proceedings of the Annual SRII Global Conference, Mar.-Apr. 29-02, IEEE Xplore Press, San Jose, CA, pp: 277-284. DOI: 10.1109/SRII.2011.38

Samimi, P., Y. Teimouri and M. Mukhtar, 2014. A combinatorial double auction resource allocation model in cloud computing. Inform. Sci., DOI: 10.1016/j.ins.2014.02.008

Schad, J., J. Dittrich and J.A. Quiané-Ruiz, 2010. Runtime measurements in the cloud: Observing, analyzing and reducing variance. Proc. VLDB Endowment, 3: 460-471. DOI: 10.14778/1920841.1920902

Shang, S., J. Jiang, Y. Wu, Z. Huang and W. Zheng *et al*., 2010. DABGPM: A Double Auction Bayesian Game-Based Pricing Model in Cloud Market. In: Network and Parallel Computing, Ding, C., Z. Shao and R. Zheng (Eds.)., Springer Berlin Heidelberg, ISBN-10: 978-3-642-15671-7, pp: 155-164.

Teng, F. and F. Magoulès, 2010. Resource pricing and equilibrium allocation policy in cloud computing. Proceedings of the IEEE 10th International Conference on Computer and Information Technology, IEEE Xplore Press, Bradford, pp: 195-202. DOI: 10.1109/CIT.2010.70

Tsai, W.T. and G. Qi, 2012. DICB: Dynamic intelligent customizable benign pricing strategy for cloud computing. Proceedings of the IEEE 5th International Conference on Cloud Computing, Jun. 24-29, IEEE Xplore Press, Honolulu, HI, pp: 654-661. DOI: 10.1109/CLOUD.2012.49

Xu, H. and B. Li, 2013. Dynamic cloud pricing for revenue maximization. IEEE Trans. Cloud Comput., 1: 158-171. DOI: 10.1109/TCC.2013.15

Zhuang, H., 2009. Cost-efficient resource allocation for decentralized, EDIC research proposal.