DYNAMIC PRICING SCHEME FOR RESOURCE ALLOCATION IN MULTI-CLOUD ENVIRONMENT

***Nurul Ainaa binti Muhamad Shaari*[1], *Tan Fong Ang*[2], *Lip Yee Por*[3], *Chee Sun Liew*[4]**

[1]School of Information Technology, SEGi University, Petaling Jaya, 47810 Selangor Darul Ehsan, Malaysia.

[2,3,4]Department of Computer System and Technology, Faculty of Computer Science & Information Technology,
University of Malaya, 50603 Kuala Lumpur, Malaysia.

Email: ainaashaari@segi.edu.my[1], angtf@um.edu.my[2], porlip@um.edu.my[3], csliew@um.edu.my[4]

ABSTRACT

Cloud computing has taken the IT and business world by storm through its cost savings and quick-and-easy adoption. Currently, fixed price model is dominating the pricing schemes in the cloud market. However, this model is unable to reflect the current market needs for cost savings as the number of cloud provider and user increases. As a result, a dynamic pricing scheme has emerged as an attractive strategy to better cope with the unpredictable cloud demand. This paper proposed a dynamic pricing scheme that provides fairness among the service providers in a multi-cloud environment. The scheme adjusts the price accordingly to encourage the low utilized resources' usage and discourage over utilized resources' usage. The evaluation results showed that the proposed scheme is able to reduce the cost of the end user when running compute-intensive and data-intensive jobs in the multi-cloud environment.

Keywords: Cloud Computing, Dynamic Pricing, Multi-Cloud, Resource Allocation, Resource Utilization

1.0 INTRODUCTION

The emergence of a new computing paradigm, Cloud Computing has become an essential utility computing resource that enables sharing and dynamic allocation of resources for various applications. The resource allocation in Cloud Computing is economic driven that utilizing all the resources to maximize service providers profit while minimizing end user cost. Cloud Computing allows the end users to pay for the usage while enabling the service providers to track usage for billing purpose. As the resources demand is dynamic and unpredictable, various challenges have arisen due to the increase number of service providers and end users. Since Cloud Computing provides pay-per-use services, the ability to offer an optimal pricing scheme is essential to maximize the service providers' profit while guarantee user satisfaction.

As the number of virtual distributed resources increases, designing an effective resource allocation algorithm in Cloud Computing is challenging. Due to the heterogeneous nature of Cloud in terms of applications and resources, the problems with resource allocations in Cloud Computing are dynamic [1]. The static allocation policies have led to inefficient resource sharing, poor utilization, waste of resources and revenue loss.

Recent works allow the end users to select resources from multiple service providers to increase the scalability and reliability of a multi-cloud environment [2]. However, the fundamental problem faced by the users is to select the most suitable service provider that fulfills their job requests with least cost.

This paper proposes a dynamic pricing scheme that provides fairness among the service providers in a multi-cloud environment. The scheme adjusts the price accordingly to encourage the usage of resources with low utilization rate and discourage the usage of resources with high utilization rate. In particular, we highlight the significance of considering the essential job's value metric that encompass the actual service resource requirements. The key contributions of this paper are to:

- Consider the essential job's value metric based on job's type: compute-intensive job and data-intensive job.
- Describe how the dynamic pricing can be implemented in a multi-cloud environment with integration of cloud brokering mechanism.
- Analyze the performance of static pricing mechanism for resource allocation through experimental results to demonstrate the cost effectiveness of the proposed scheme.

This paper is organized as follows: Section 2 discusses the related works. Section 3 presents the proposed scheme in depth. Section 4 describes the system architecture of the proposed scheme. Section 5 discusses the evaluation methods and results. The final section concludes the paper.

2.0 RELATED WORKS

Although Cloud Computing provides many new opportunities to the computational world, the technology is still immature. There are still various issues and challenges that need to be addressed especially in the area of pricing model [3-7]. The fixed pricing schemes that currently used by the service providers are not able to adapt to the rapid changes between supply and demand in the market. Besides, the schemes are unfair to the service providers and end users with specific quality of service (QoS) requirements [8].

In contrast, the dynamic pricing scheme is better than the fixed pricing scheme in terms of unpredictable demand [2, 8-12]. A strategy-proof dynamic scheme for resource allocation on federated clouds is proposed by [2]. The authors concluded that the dynamic pricing scheme is more suitable for federated clouds. The pros and cons of charging fixed prices over variable prices for cloud resources are investigated in [8]. The authors evaluated the fixed price scheme with the proposed autonomic pricing mechanisms that self-adjust pricing parameters to meet users' requirements. The results showed that the autonomic pricing mechanism is able to achieve higher revenue.

From economic point of view, every service provider adopts different pricing schemes on the similar features resources. The distinction between various cost models is essential to derive more accurate business strategies by the service providers [13]. The potential benefits and the influence of mixed pricing are studied in [14]. The authors concluded that having different pricing schemes allow end users to reserves resources from different service providers. However, this has created the challenges for the end users to choose the best among the service providers in the cloud computing market.

Although dynamic pricing scheme is suitable for cloud environment, the unpredictable price fluctuation may cause the end users to select the costly service provider. Henzinger et al. [3] proposed a flexible framework called FlexPRICE that provides enough flexibility to the end users. Lucas Simarro et al. proposed an optimize price prediction model to address the above problems [15].

There exist quite a number of researches that enhanced the research allocation in cloud [16-23]. A matchmaking strategy that base on job execution time was proposed for resource allocation in [16]. Goudarzi, H. et al., Hien Nguyen, V. et al, and Ang Tan Fong et al. proposed the SLA-Based resource allocation that mainly focus on the number and type of computing, data storage and communication resources [17-20]. The utility based resource allocation strategy that maximize the service providers profit was proposed by Goudarzi, H. et al. and Minarolli, D. et al. [21-22]. Khatua, S. et al., proposed a resource allocation technique that optimizes the utilization of virtual resources in Cloud. However, none of them propose the pricing model for resource allocation that can save the end user cost while provide fairness to the service providers.

Due to the above findings and shortcomings, we proposed a dynamic pricing scheme for resource allocation that provides fairness among the cloud service providers and save end user cost in a multi-cloud environment. The scheme adjusts the price accordingly to encourage low utilized resources' usage and discourage over utilized resources' usage.

3.0 DYNAMIC PRICING SCHEME

The proposed scheme supports both the time-shared and space-shared resources. If a job is compute intensive, it will be executed in the time-shared resource. The time-shared resources allow the sharing of processing power using multiple cores as well as supporting multithreading. On the other hand, if a request is data intensive job, the job will be executed in the space-shared resource. The space-shared resources allow the sharing of memory space such as memory or storage. Our proposed scheme is mainly based on the utilization of the pool of these resources. The resource utilization determines the current load in each pool. The price increases as the level of resource utilization increases. This scheme will influence the end users whether to switch from the congested resource pools to other pools or not.

The service provider specifies the fixed price rate per unit of a resource which we consider as our base price $PBase$. $REStotal_i$ is the total resource unit in a datacenter C_i and $RESfree_i$ is the total free resource in a datacenter C_i. The dynamic price $PDyn$ is derived as indicated in the following equation:

$$PDyn_i = \frac{REStotal_i}{RESfree_i} \times PBase_i \tag{1}$$

According to [24], servers run at low utilization may cause idle power wasted. Similarly, when the average load approaches 100%, the servers will have difficulty to achieve the desire throughput and may cause latency [25]. Thus, the reasonable range of utilization level that is acceptable in the proposed scheme is in between 30% and 70%. The idea is to encourage the usage of low utilization resources and discourage the usage of high utilization resources. Fig. 1 denotes the pricing scheme that based on the utilization level. In the proposed scheme, discount rate is given for the low utilization resources to encourage its usage. On the other hand, additional charge is added to discourage usage of the high utilization resources.

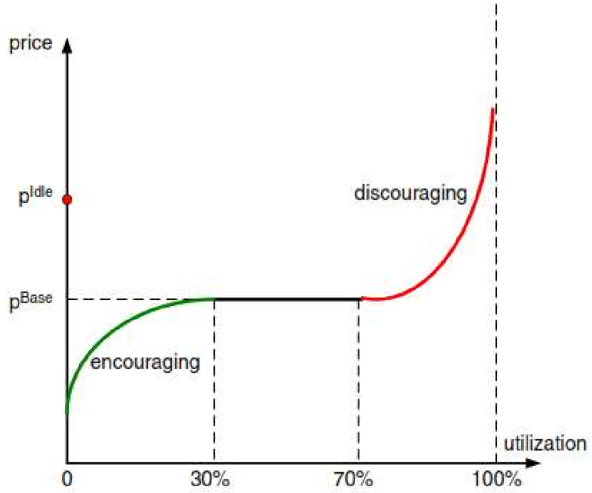

Fig. 1: Pricing scheme according to utilization level

Let define the utilization *Util* at reasonable level is between $Util^{low}$ and $Util^{high}$. The proposed pricing function $PDisc_i$ is given as follows:

$$PDisc_i = \begin{cases} PBase_i - discount, \ Util_i < Util_i^{low} \\ PBase_i, \ Util_i^{low} \le Util_i \le Util_i^{high} \\ PBa\quad_i + Extra, \ Util_i > Util_i^{high} \end{cases} \tag{2}$$

The pricing function $PDisc$ comprises of two components which are the static component base pricing rate $PBase$, and the dynamic component using the discount factor. $P(Util)$ is computed using the following polynomial function:

$$P_i(Util_i) = \sqrt{PBase_i \cdot 10}\,\left(Util_i - {}^{1}\!/_{2}\right)^3 \tag{3}$$

$$PDisc_i = PBase_i + P_i(Util_i) \tag{4}$$

The aim of the proposed scheme is to provide end users with the least cost for the resource consumed. Hence, using the cost function as the main metric in the resource allocation strategy is important to achieve the objective as well as provide fairness to the service providers. In order to identify the least cost resources, we need to estimate the cost of running a job on each resource. Let et_{ji} is the execution time of job j running on resource i. The estimated execution time is given by:

$$et_{ji} = \frac{rl_j}{capacity_i \times cores_i} \tag{5}$$

rl is the total number of instructions of job j that is executed on the resource i. The total $capacity$ of a compute resource is defined by million instructions per second (mips) per virtual machine. To estimate the cost of running a job on the resources, $cost_{ji}$ is given by:

$$cost_{ji} = et_{ji} \times price_i \tag{6}$$

where the $price_i$ denote the pricing scheme adopted by resource i. The pseudocode of the proposed dynamic pricing scheme is outlined in Fig. 2. The job requests are sorted according to the longest job first (LJF) for each interval. The utilization level in each datacenter DC_i is identified to derive the $price_i$. Then, the $price_i$ is used to determine $cost_{ji}$ for each DC_i and each cost is compared to get DC_i with the least amount of $cost_{ji}$. Finally, the job request will be allocated to the resources on the selected DC_i. The process is repeated until all requests are served.

```
Input:
   $J_j$ the job request submit by the user
   $DC_i$ the datacenter's list in the cloud market with the available resource r
   associated with price. <$r_i$, $price_i$>

Output:
   A schedule of $J_j$ into resource r at $DC_i$ with estimated job cost, $cost_{ji}$

∀ Sort the request $J_j$ in descending order  based on parameter value
   Generate a sorted job request list $J_j$, sortJ
    ∀ sortJ
      ∀ DC
        Calculate the utilization level at each $DC_i$, $Util_i$
        //Adopting the proposed pricing scheme Pdyn or PDisc to calculate $price_i$
        Get current price $price_i$
        Compute the estimated execution time $et_j$ at each $DC_i$
        Compute the estimated cost $cost_{ji}$ based on $et_j$ and $price_i$
       Get the minimum $cost_{ji}$ and allocate $J_j$ to the selected $DC_i$
```

Fig. 2: Pseudocode of the dynamic pricing scheme

4.0 SYSTEM ARCHITECTURE

Fig. 3 depicts the system architecture for the proposed scheme. As shown in the architecture, different service providers offer the resources associated to their own pricing schemes. The end user can submit job requests to the service providers in the Cloud market by specifying the requirements to the broker. The broker acts as intermediary between the end user and the service providers. The proposed dynamic pricing scheme for resource allocation is implemented on the broker. The process of resource allocation is transparent to the end users. The broker is responsible to select the service provider that minimizes the end user cost.

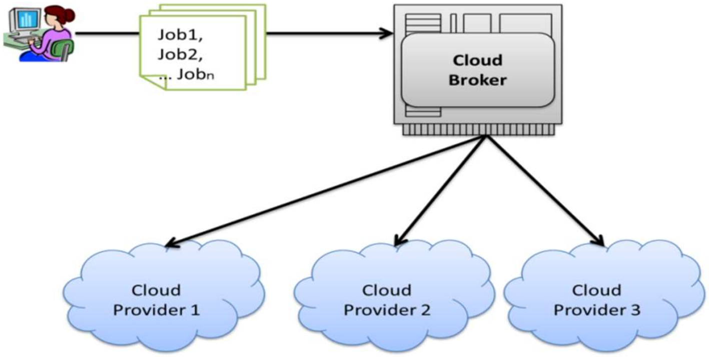

Fig. 3: System architecture of the proposed scheme

The proposed dynamic pricing scheme is built on top of the CloudSim simulator [26]. CloudSim simulator supports the modeling of the Cloud economic entities such as end users, brokers and the service providers. The simulator also provides support for resource management and scheduling as well as multi-provider Cloud environment. Fig. 4 shows the communication in between the service provider (DataCenter) and the broker (DataCenterBroker). CIS entity provides the match making services to map user/broker requests to the suitable service provider.

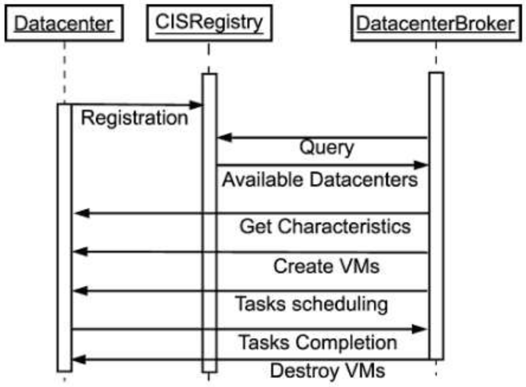

Fig. 4: Communication between service provider and broker

In this study, the DataCenterBroker entity is enhanced to include our proposed scheme. The enhanced broker not only can manage task scheduling process and resource allocation, but also able to monitor the resource utilization state. When the broker receive the job requests, the broker estimates the execution time of each request running in each service provider based on the resource utilization state. Then, the broker can allocate the job to the service provider that offers the least cost. The process flow of the enhanced module is outlined in Fig. 5.

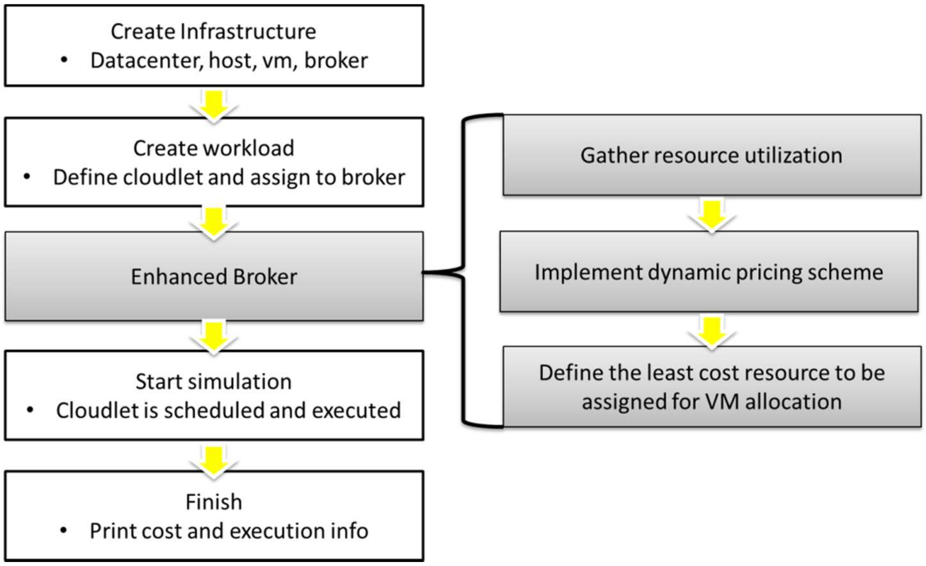
Create Infrastructure
• Datacenter, host, vm, broker
Create workload
• Define cloudlet and assign to broker
Enhanced Broker
Start simulation
• Cloudlet is scheduled and executed
Finish
• Print cost and execution info
Gather resource utilization
Implement dynamic pricing scheme
Define the least cost resource to be assigned for VM allocation

Fig. 5: Process flow of the enhanced DataCenterBroker module

5.0 EVALUATION AND RESULTS

In the evaluation phase, the compute intensive and data intensive jobs are used as the input. A benchmark dataset SPEC CINT2006 suite [27] is used as the compute-intensive jobs whilst the MalStone [28] is used as the data-intensive jobs. Both dataset define the jobs' length and size as a comparative measure of performance workload based on the real applications. Table 1 shows the predefined set of resources and job parameters that are used to evaluate the performance of the proposed scheme.

Table 1: A predefined set of resources and job parameters

Number of Cloud provider	3
Type of Job	Compute-Intensive and Data-Intensive
Number of hosts on each datacenter	10
Number of PE on each host	4
Number of jobs running	100 - 180
Length of job	3000 – 38000 MI
Size of job	1800 – 4800 GB
Interval	5 seconds

The multi-cloud environment consists of three service providers from the well-known public cloud such as Amazon EC2, GoGrid and Rackspace. Each service provider consists of 10 hosts with 4 processor elements (PE) each. The number of running job is within 100 - 180. The length of compute-intensive job is within the range of 3000 to 38000 million instructions (MI) and the data size is in the range of 1800 to 4800 gigabytes (GB). The range is based on the benchmark dataset as above [27-28]. The number of resources and hosts is scaled down to reduce the simulation time. The time interval used for the experiment is five seconds. Every five seconds, the broker processes the job request from the buffer.

Table 2 defines the details of resource information for each datacenter. The resource parameter is characterized based on the small virtual instance offered by public Cloud provider Amazon EC2, GoGrid and Rackspace. We assumed that each datacenter are using the homogeneous resources and consists of the same number of resources. The processing element (PE) of each resource is fixed to one to simplify the analysis. As for the simulation purpose, the cost model is scaled down to cost per second instead of cost per hour and the cost per storage is scaled up by multiplying each cost by ten. The cost per storage is calculated based on the number of terabytes. The cost per storage is scaled up for all the providers since the price is too small to show the differences given the benchmark dataset used are in between 1.8 to 4.8 terabytes.

Table 2: VM resource characteristics

Datacenter	Amazon EC2	GoGrid	Rackspace
RAM	1740	1024	1024
MIPS	2660	2260	2200
CostPerSecond	$ 0.60	$ 0.80	$ 0.60
CostPerStorage	$ 0.095	$ 0.120	$ 0.150

Several experiments are conducted to compare the performance in terms of total cost of the job execution. The experiments include 1) Static pricing scheme versus dynamic pricing scheme; 2) dynamic pricing scheme with different resource allocation algorithms namely Random allocation, Round Robin allocation and proposed method.

Fig. 6 shows the total cost of running compute-intensive jobs using static and dynamic pricing scheme in the resource reservation. The figure shows that the total cost increases steadily as the number of jobs increased. However, the total cost of static pricing scheme is higher than the dynamic pricing scheme. The dynamic pricing scheme reduces the total cost by 55% – 68% as compared to the static pricing scheme. This is due to the static pricing policy that reserved all the resource until the jobs execution is completed. On the other hand, the resources are released once the jobs are completed in the dynamic pricing scheme.

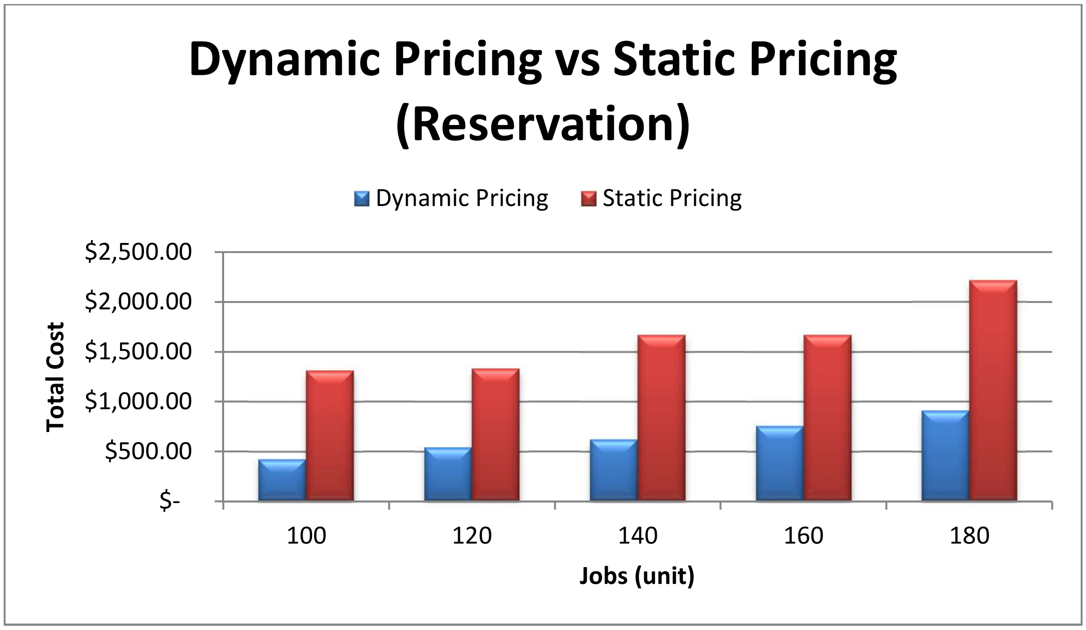

Fig. 6: Total cost when running compute-intensive job

Fig. 7 depicts the total cost of running compute-intensive jobs using different allocation algorithms. The figure shows that the proposed scheme reduced the total cost by 12% - 14% and 10% – 17% as compared to the round robin allocation and random allocation respectively. The proposed scheme changes the price according to the resources' load utilization. By reducing the cost for the lower utilize resource provides fairness among multiple service providers as well as minimize the total end users' cost. Round Robin also provides fairness allocation among service providers but the cost incurred is higher than as compared to the proposed scheme. Since Random scheme allocates job without considering the state of resource utilization. The scheme has the highest execution cost as compared to others.

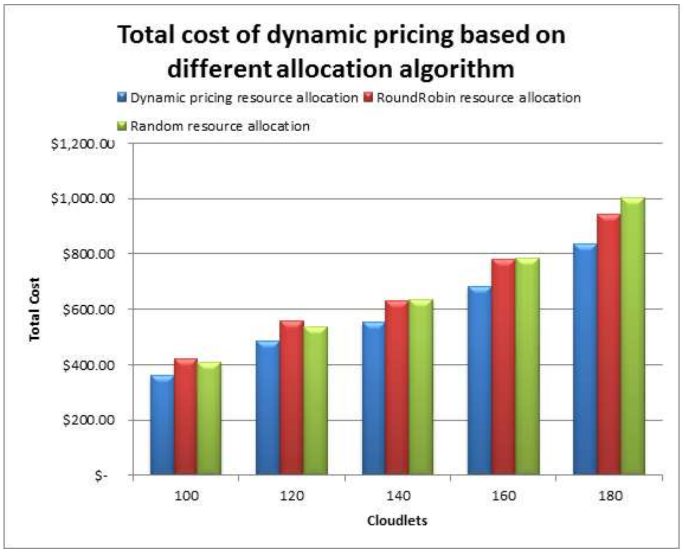

Fig. 7: Total cost when running compute-intensive job using different allocation algorithms

Fig. 8 depicts the total cost of running data-intensive jobs using different allocation algorithms. The proposed scheme reduces cost by 5 – 11 % and 2 – 16 % with respect to the Round Robin and Random scheme. The proposed scheme is better in terms of cost effectiveness as compared to others. The proposed scheme can direct the job to less congested service providers to reduce the latency. Since Round Robin allocates the jobs equally among the service provider, disregard the network state, the congested service provider may cause bottleneck which led to a longer time to transfer the data. The same goes to Random scheme that allocates the resources randomly among the service providers.

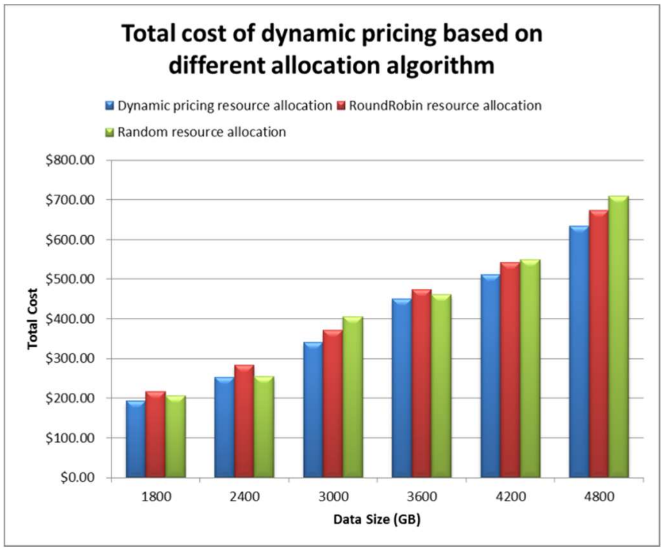

Fig. 8: Total cost when running data-intensive job using different allocation algorithms

6.0 CONCLUSION

This paper focuses on the challenging issues of the pricing schemes and the efficiency of resource utilization across multiple-clouds service providers. A dynamic pricing scheme is proposed to minimize the end users' cost and provide fairness among the service providers. The scheme adjusts the price accordingly to encourage the usage of resources with low utilization rate and discourage the usage of resources with high utilization rate. The experimental results show that the proposed scheme outperforms the static pricing scheme and other common resource allocation algorithms in terms of total end user cost. For future work, we would like to incorporate others optimization parameter such as power consumption and user feedback in the dynamic pricing scheme. Besides, we also plan to consider real-world workload and real time dataset in the future.

ACKNOWLEDGEMENTS

This work was funded by Ministry of Education Malaysia (FRGS FP051-2013A) and UMRG grant (RP001F-13ICT).

REFERENCES

[1] V.Vinothina, V., Dr.R.Sridaran, & Ganapathi, D. P., "A Survey on Resource Allocation Strategies in Cloud Computing.", International Journal of Advanced Computer Science and Applications (IJACSA), Vol. 3, No. 6, 2012, pp: 97-104.

[2] Mihailescu, M., & Teo, Y. M., "Dynamic Resource Pricing on Federated Clouds.", 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid), 2010, pp: 513-517.

[3] Henzinger, T. A., Singh, A. V., Singh, V., Wies, T., & Zufferey, D., "FlexPRICE: Flexible Provisioning of Resources in a Cloud Environment.", IEEE 3rd International Conference on Cloud Computing (CLOUD), 2010, pp: 83 - 90.

[4] Anandasivam, A., & Premm, M., "Bid Price Control and Dynamic Pricing in Clouds.", ECIS 2009 Proceedings, 2009, paper 284.

[5] Buyya, R., "Market-oriented cloud computing: vision, hype, and reality of delivering computing as the 5th utility.", IEEE International Symposium on Cluster Computing and the Grid, 2009, pp: 12-15.

[6] Buyya, R., Abramson, D., Giddy, J., & Stockinger, H., "Economic models for resource management and scheduling in Grid computing.", Concurrency and Computation: Practice and Experience, 2002, Vol. 14 No.13-15, pp: 1507-1542.

[7] May Al-Roomi, Shaikha Al-Ebrahim, Sabika Buqrais and Imtiaz Ahmad, "Cloud Computing Pricing Models: A Survey", International Journal of Grid and Distributed Computing, 2013, Vol.6, No.5, pp: 93-106.

[8] Yeo, C. S., Venugopal, S., Chu, X., & Buyya, R., "Autonomic metered pricing for a utility computing service. ", Future Generation Computer Systems, 2010, Vol. 26, No. 8, pp: 1368-1380.

[9] Teo, Y. M., & Mihailescu, M., "A Strategy-proof Pricing Scheme for Multiple Resource Type Allocations.", Proceedings of the 2009 International Conference on Parallel Processing, 2009, pp: 172-179.

[10] Du, L., "Pricing and Resource Allocation in a Cloud Computing Market.", Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012), 2012, pp: 817-822.

[11] Samimi, P., & Patel, A., "Review of pricing models for grid & cloud computing.", IEEE Symposium on

Computers & Informatics (ISCI), 2011, pp: 634-639.

[12] S. Lehmann and P. Buxmann, "Pricing Strategies of Software Vendors", Business and Information Systems Engineering, Vol. 1, No.6, 2009, pp: 452-462.

[13] Mach, W., & Schikuta, E., "A Consumer-Provider Cloud Cost Model Considering Variable Cost.", IEEE Ninth International Conference on the Dependable, Autonomic and Secure Computing (DASC), 2011, pp: 628-635.

[14] Ma, D., & Huang, J., "The pricing model of cloud computing services.", Proceedings of the 14th Annual International Conference on Electronic Commerce, 2012, pp: 263-269.

[15] Lucas Simarro, J. L., Moreno-Vozmediano, R., Montero, R. S., & Llorente, I. M. "Dynamic placement of virtual machines for cost optimization in multi-cloud environments.", International Conference on High Performance Computing and Simulation (HPCS), 2011, pp: 1-7.

[16] Melendez, J. O., & Majumdar, S., "Matchmaking with limited knowledge of resources on clouds and grids.", International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010, pp: 102-110.

[17] Goudarzi, H., & Pedram, M., "Multi-dimensional SLA-Based Resource Allocation for Multi-tier Cloud Computing Systems.", IEEE International Conference on Cloud Computing (CLOUD), 2011, pp: 4-7.

[18] Hien Nguyen, V., Tran, F. D., & Menaud, J. M., "SLA-Aware Virtual Resource Management for Cloud Infrastructures.", The Ninth IEEE International Conference on Computer and Information Technology, 2009, pp: 357-362.

[19] Ang Tan Fong, Ling Teck Chaw, Phang Keat Keong, "An Adaptive QoS Scheduling in Service Oriented Grid Environment", Turkish Journal of Electrical Engineering & Computer Sciences, Vol. 20 No. 3, 2012, pp: 413-424.

[20] W.K. Ng, T.F. Ang, T.C. Ling, C.S. Liew, "Scheduling Framework For Bandwidth-Aware Job Grouping-Based Scheduling In Grid Computing.", Malaysian Journal of Computer Science, Vol. 19, No. 2, 2006, pp: 117-126.

[21] Goudarzi, H., & Pedram, M., "Maximizing Profit in Cloud Computing System via Resource Allocation.", 31st International Conference on Distributed Computing Systems Workshops (ICDCSW), 2011, pp: 1 - 6.

[22] Minarolli, D., & Freisleben, B., "Utility-based resource allocation for virtual machines in Cloud computing.", IEEE Symposium on Computers and Communications (ISCC), 2011, pp: 410 - 417.

[23] Khatua, S., Ghosh, A., & Mukherjee, N., "Optimizing the utilization of virtual resources in Cloud environment", IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS), 2010, pp: 82-87.

[24] Moghe, U., Lakkadwala, P., & Mishra, D. K., "Cloud computing: Survey of different utilization techniques.", Sixth International Conference on Software Engineering (CONSEG), 2012, pp: 1-4.

[25] Zheng, Q., & Veeravalli, B., "Utilization-based pricing for power management and profit optimization in data centers.", Journal of Parallel and Distributed Computing, Vol. 72, No. 1, 2012, pp: 27-34.

[26] Calheiros, R. N., Ranjan, R., Beloglazov, A., Rose, s. A. F. D., & Buyya, R., "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning

algorithms.", Softw. Pract. Exper., Vol. 41 No. 1, 2011, pp: 23-50.

[27] Phansalkar, A. S., "Measuring program similarity for efficient benchmarking and performance analysis of computer systems", University of Texas at Austin, 2007.

[28] Bennett, C., Grossman, R. L., Locke, D., Seidman, J., & Vejcik, S., "Malstone: towards a benchmark for analytics on large data clouds", Proceedings of the 16th ACM SIGKDD International conference on Knowledge discovery and data mining, 2010, pp: 145-152.