IJCTA, 10(8), 2017, pp. 73-84 ISSN: 0974-5572

Scheduling based Cost Optimization in Cloud Computing

Subhadarshini Mohanty*, Prashanta Kumar Patra, Mitrabinda Ray* and Subasish Mohapatra****

ABSTRACT

Now a day's cloud computing has become one of the promising areas of research and technological advancement. It has become an integral part of the new era. This distributed computing paradigm makes the resources in the form of software, application, storage, and services available to remote users on demand basis. Virtualization technique enables cloud computing to provide abundant opportunities to avail the resources with a pricing model. Price calculation has become a critical issue in cloud environment due to dynamic change in user demand and available resource. Authors have proposed different pricing model to minimize the cost of cloud services. In this paper, we have highlighted different issues concerning the cost of cloud resource and proposed a pricing model to optimize the cloud cost. Simulations are carried out in the CloudSim environment, and results are demonstrated.

***Index Terms*:** Cloud Computing, Virtual Machine, FCFS, RR, Min-Min, Max-min, Pricing Model.

1. INTRODUCTION

Now a day's price calculation has become an important research topic in cloud computing. Different cloud providers have used different pricing models and schemes to provide services. Cloud provides alternate subscription model and pricing plans for the user flexibility. These are of different modes such as on-demand, reservation, and ad-hoc market plans. This permits them to effectively focus on an assortment of client cluster with particular inclinations and to create more income likewise. For these computing resources, execution platform and enterprise software is also expected to be rented as a service along with the cloud infrastructure. With the existing large variety of options, it is extremely hard for a consumer to assess different offers and choose an alternate provider which will fit their requirements.

In cloud computing demands and associated costs are controlled by several factors (i.e. storage, bandwidth, transaction, processing speed, etc.) and "pricing" is considered to be the most important one. Cloud providers always use pricing model to know how much service provisioning could be done for different consumers and the relationship between pricing and other issues such as provisioning period and grant discounts.

Pricing is the parameter through which a publisher determines what will be received from a consumer in exchange for their services [1]. Authors have claimed that by developing adequate pricing policies cloud computing can achieve great success in a different domain. Pricing models are divided into two types: such as Fixed Pricing Model and Dynamic Pricing Model [2]. In Fixed Pricing Model the price charging doesn't change, and the cloud provider is someone who determines the price to the resource type in advance.

Consumers avail the services at any instance of time as pay-per-use model. According to Yeoa et al., fixed pricing model is more straight forward and easy to understand, but it is unfair to all customers because

* Department of Computer Science and Engineering, Siksha 'O' Anusandhan University, Bhubaneswar, Odisha, *Emails: sdmohantycse@cet.edu.in , mitrabindaray@soauniversity.a.cin*

** College of Engineering & Technology, Bhubaneswar, Odisha, India, *Emails: pkpatra@cet.edu.in, smohapatra@cet.edu.in*

they do not have the same needs [3]. Subscription and list price are another form of fixed price models. In subscription model the consumers pay a fixed amount by utilizing the services for a longer period of time. Similarly, in the list price model a fixed price is found in a catalog or a list. In another hand, the Dynamic Pricing Model, the charging of price changes dynamically according to market status quo. The different parameters that influence the dynamic pricing model are service characteristics, customer requirement, real-time market environment such as bargaining, auctioning, demand behavior, etc [4]. Apart from this other driving factors that impact the pricing models are Initial costs, lease period, Quality of service, the age of resources, the cost of maintenance. Furthermore, determining the cost to pay for these benefits faces many challenges such as (a) Understanding the cloud resources and their significantly different pricing models. (b) Understanding the deployment of the enterprise software in the cloud and (c) Most importantly, estimating the potential cost of obtaining different services [5].

2. LITERATURE SURVEY

Pay-as-you-go is a static pricing model. In this price determined by the cloud provider by charging resources based on usage, i.e. the customer pays a fixed price and reserves resources for the paid period [6]. But in this pricing model over provisioning and under provisioning problems may occur. This provides on-demand-services.

For reservation, the subscription model is used. Here price depends on subscription period (i.e. month or year). The consumer may pay more than the real utilization cost when resources/services do not use properly. It is good for the consumer when resources/services are utilized extensively. Prepaid per-use pricing model requires consumers to pay the money in advance and the amount is deducted from the credit as per the usage [7]. Similarly, to maximize the service provider's resource utilization, the pay-for-resource model is used. It is static in nature and hard to implement.

Competition-based pricing model is a popular dynamic approach. It determines price depends on competition. Some of Consumer's and providers participate in the cloud market [8]. This approach maintains market's elasticity and decreases investment risks.

Aarti Singh et. al. describes three cases of optimizing resource allotment between consumer and provider to reduce the price. That, if consumers request equals to the resources provides by the cloud provider then use Hungarians method for one-to-one resource allocation [9]. If the request is less than the resources, first come first serve (FCFS) method is used. Else resource is less than a request, then ask the user to modify the request or use Belief-Desire-Intention (BDI) architecture. That is defined by the equation $\zeta i\ (\alpha iIi\ ) \leq \zeta i\ (\beta i)$ and i =1, 2, ...n. Where αi represents the total number of resources available, βi represents the number of requests by the end user, αiIi is the resource instance matrix which represents the instances of every resource and ζi describes the cost of the resource. Authors have not specified any cost function to calculate the cost.

Authors have proposed a cloud bank agent model and iterative price update algorithm that utilize the previous data to reduce the cost of the provider and maximizes the profit [10]. This model is not able to manage the dynamic change in producer and consumer behavior in the market.

Wang et.al. Proposed a deadline based pricing model which work on Real-time tasks. In deadline dependent pricing charges users have to pay based on when they have consumed the resources rather how much resources are consumed [11]. This maximizes revenue and minimizes cost by job scheduling, which decreases the use of electricity on datacenters. For job scheduling, it uses the shortest deadline first method. Static behavior and homogeneous data centers are the disadvantages of this algorithm.

Amelie Chi Zhou and Bingsheng have used transformation-based optimization framework which organizes tasks flow to optimize cost [12]. Direct Acyclic Graph (DAG) is used to model these workflows of tasks. The time estimation and price units are in hours. The hourly price of type-i instance is Pi. Since partial hours are rounded, a task A with execution time tA on the instance costs tA * Pi. This method optimizes performance and cost of the workflow. But it was not considered other parameter for the cost (i.e. storage, bandwidth, ram).

Luca Ferretti, Fabio Pierazzi, and Michele Colajanni [13] calculate cost on a database. User's data divided into two type plaintext and encrypted text. If there is no demand for data security, then cost calculates based on plain data store in the database. Otherwise, cost calculates on data encryption and storing of encrypted data.

A customer based flexible pricing model is proposed by Igor Ruiz-Agundez, Yoseba K. Penya, and Pablo G. Bringas. Price is put according to what the customer is prepared to pay. This model also gives flexibility to business operators to pertain different pricing schemes [14]. On the turn, it also enables better accounting on the usage of the cloud computing. Furthermore, this model could be expanded to different activities in the business support system. This model is constrained with calculating the response times and revenue calculations.

R. Piro, A. Guarise and A. Werbrouck proposed a Hybrid pricing model in which price is changed according to the waiting time in the job queue [15]. Here the price authorities are responsible for dynamically adjusting the prices. It is a simple model with low computational overhead.

Authors have proposed compound Moore's law to develop Clabacus architecture in which a computational block is used to calculate price by using different price computation algorithms [16]. Compounded Moore's Law has been used to improve the hardware deployment. This architecture also calculates Value-at-Risk by using Genetic Algorithm approach.

M. Macias and J. Guitart have proposed a genetic algorithm-based model where the naive pricing function evolves to a pricing function that offers suitable prices in function of the system status [17]. The proposed model can be used to make complex decisions by providing flexible pricing model.

In this model, the users are divided based on the resource purchase i.e. Reservation and on-demand [18]. In On-demand scheme the consumers are charges based on actual usage. The service is charged at a fixed rate per billing cycle (i.e. hourly) from the time the VM instance is lunched until it terminated. But for reservation plan consumer can avail on-demand service for one month or one year with a reserve capacity. In this scheme, the customer has to pay in advance. The on-time fee must be paid irrespective of how much the instance is used during the reservation period. Pseudo-Optimal and heuristic algorithms are used to reduce the computational complexity.

Goa et. al. has proposed a dynamic resource allocation using Intelligent Economic Approach. This method consists of five components such as Cloud Service Provider, Provider Agent, Cloud Service Consumer, Consumer Agent, and Auction Intermediary. In this CSC requests for services and CSP provide services [19]. CA and PA make initial tender. Then they both submitted the tender to AI. It specifies numbers of users request for the resource by predicting price and numbers of providers participate with different types of resource combination by asking price. AI work as third party broker and run Winner Determination Algorithm to map user request to provider's resource. After completion of work CSC and CSP evaluate partner's performance and give a rating as per the quality of service. This method helps to maintain stability in the market.

3. OBSERVATIONS

From the above survey, it is summarized that user can demand two type of services i.e. on-demand service and reservation service. In these method tasks submitted by the user are either static or real-time based. Both types of tasks executed in virtual machines. Task may take single or more virtual machines for its execution. The mapping between tasks and the virtual machines are managed by a third party, known as a broker. This help to schedule task, check for its availability of resources, and assigning the resources for execution. Different methods for cost calculation have been proposed by different authors. Cost can be calculated based on transaction, database, execution, and scheduling along with market characteristics. Cost calculated on both side; consumer side it is known as "predictive price" and on provider side, it is known as "ask price". These two price models help the broker for mapping the tasks to cloud resources. Services are rated by its user, which define the quality of service of the provider. A large number of user and provider maintain market stability and flexibility. It increases competition in the market. This helps the consumer to get resources in a generalized range. Developing a generalized pricing model in view to resource utilization is one of the promising researches in the field of cloud computing.

4. OBJECTIVES

The main objectives are to

[1] Add time factor, (i.e. execution time and waiting time) in the cost function. Usually, users start utilizing the resources when the tasks submitted to the cloud. Till resources have not given, the task has to wait on the ready queue, which increases the cost and decreases the quality of service. So by minimizing waiting time of the tasks price can be minimized.

[2] Minimization of waiting time and execution time can be done by a good mapping scheme between the user's task and provider's resource.

[3] The user has to assign only that amount of resource, which needs for completion of tasks and has to pay only for the used resource.

5. PROPOSED PRICING MODEL

In this proposed work the cloudlets are assigned to VMs using different heuristic approach. These algorithms help to achieve a Quality of service (QoS) by reducing the time required to complete the tasks. After that, the proposed start simulation cost function is called. The cost function is based on the require resources by the task, execution time to complete the task, and waiting time (i.e. task has to wait to get the VM's CPU).

Cloud infrastructure is divided into datacenters (DCs). The DCs are further subdivided into hosts, and virtual machines (VMs) are created within the host. When the user submits the tasks called cloudlets then that are executed in the VMs.

Here broker work as the third party, which gets the list of the cloudlets (cls) and the list of the VMs. The job of the broker is the mapping of cls with VMs. For this, it uses scheduling algorithm. Here First Come First Serve (FCFS), Round Robin (RR), Min-Min and Max-Min algorithms are used for mapping.

5.1. Start Simulation ()

After mapping of tasks or cls with VM, list of mapping is returned to the broker and start simulation executes. It allocates each cls required amount of Storage, RAM, Bandwidth, and CPU. When cloudlet submitted to the VM, it gets the storage in the VM. Then at the time of execution, it uses RAM, Bandwidth, and CPU as per its requirement. Based on this, a cost model has proposed.

Calculate each cloudlet cost by

TET = cludlet.getActualCPUTimeO;// total Execution time (1)

length = cludlet.getFinshTimeO–cloudlet.getSubm,issionTimeO (2)

cost[i]= (cloudlet.getUtilizationOfCpu(TET)*cloudlet.getProcessingCostO+
cloudlet.getUtilizationOfRam(TET) (3)

*0.05 + cloudlet.getUtilizationOfBw(TET) * 0.0001) * t
+ (cloudlet.getUtilizationOfBw(TET) * 0.001) * length (3)

$$\text{Total\_cost} = \sum_{i=0}^{k} \text{cost}\left[i\right] \tag{4}$$

Where,

cloudlet.getUtilizationofCpu(TET) = Returns the utilization of CPU for the time TET.

cloudlet.getProcessingCost() = Returns the total cost of processing

Cloudlet. cloudlet.getUtilizationofRam(te) = Returns the utilization of RAM for the time TET.

cloudlet.getUtilizationofBw(TET) = Returns the utilization of BW for the time TET.

cloudlet.getCloudletTotalLength() = Returns the total length of this Cloudlet.

i = It represents the number of cloudlets.

0.05 is the cost of memory.

0.0001 is the cost of Bandwidth.

0.001 is the cost for storage.

5.2. Stop Simulation ()

After execution of all tasks on their allocated VM, simulation gets the end. Between start and stop simulation all the execution and allocation result has printed. It has given the resource amount utilized by each cloudlet and calculates the cost for each. The cost has calculated using the above cost function. At last, it gives the total cost for execution of all assign cloudlets.

Here the cloudlets are assigned to VMs using FCFS, RR, Max-Min and Min-Min approach. These algorithms help to achieve a Quality of service (QoS). Here QoS define by less time required to complete

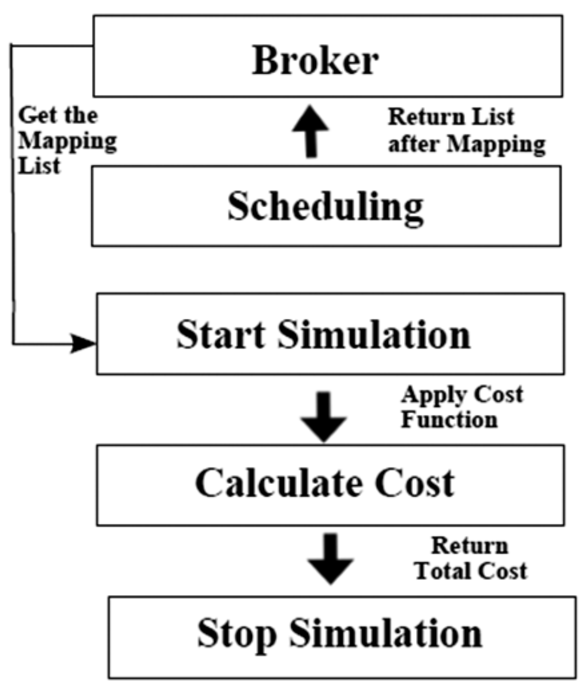

Figure 1: The Proposed Pricing Model

the user tasks. Then after start simulation proposed cost function has been called. The cost function is based on the require resources by the task, execution time to complete the task and waiting time (i.e. task has to wait to get the VM's CPU). Next chapter shows the result after applying this workflow.

6. SIMULATION PARAMETERS

To implement above work flow model, CloudSim frame work is used. First, a datacenter (DC) has been created, the single dc has divided into three hosts, and again hosts are virtually divided into virtual machines (VMs). Cloudlets are the tasks and created as per requirement. CloudSim parameter and assumption, which are taken for the simulation, are given below.

A. Data Center Characteristics

Architecture = "x86" // system architecture

Operating System= "Windows" // operating system

Cost = 3.0 // the cost of using processing in this resource

CostPerMemory = 0.05 // the cost of using memory

CostPerStorage = 0.001 // the cost of using storage

CostPerBw = 0.0001 // the cost of using bandwidth

B. Host Parameters

Ram = 2048 // host memory (MB)

Mips = 1000 //Memory instructions per second

Storage = 1000000 // host storage (MB)

Bandwidth = 10000 //host bandwidth

C. Host Virtual Machine (VM) Parameters

Size = 10000 // VM storage (MB)

Ram = 512 // VM memory (MB)

Bandwidth = 1000 // VM bandwidth

PesNumber = 1 // number of cpus

D. Cloudlet Parameters

Length = length* (i/2) // length of the cloudlet (in MI), where length=10000, i = no. of cloudlets, FileSize = 300//, OutputSize = 300 //, PesNumber = 1.

E. Utilized System Specification

The system here use for simulation is Dell Inspiron N4050, in which processor is Intel(R) Core(TM) i3-2350M CPU with processing capacity 2.30 Ghz. RAM is 2.00 GB. System type is 32-bit Operating System. Windows edition is Windows 7 Ultimate. For each cloudlet, the length gets updated as per the cloudlet parameter. The increase in cloudlet number proportionately increases the length.

7. SIMULATION RESULTS

It is observed from the simulation that FCFS mapping is always a space shared mapping where as RR is always a time shared mapping. But Min-Min and Max-Min mapping algorithms use the both space and time shared mapping. We have generated four different cases based on the time and space shared concept and considered the configuration simulation using Cloudsim simulation environment with 01 DC, 05 VM, and 10 Cloudlets.

7.1. Case I: (VM space and Cloudlet space shared)

In this, both VMs assignment to the host and cloudlets assignment to the VM are done in FCFS basis. The cost calculated in the above configuration is shown in Table 1.

Table 1
Cost Table: VM space and Cloudlet space shared

Mapping	*Cost ($)*
FCFS	34439.82
RR	26699.11
Max-Min	24114.03
Min-Min	24450.02

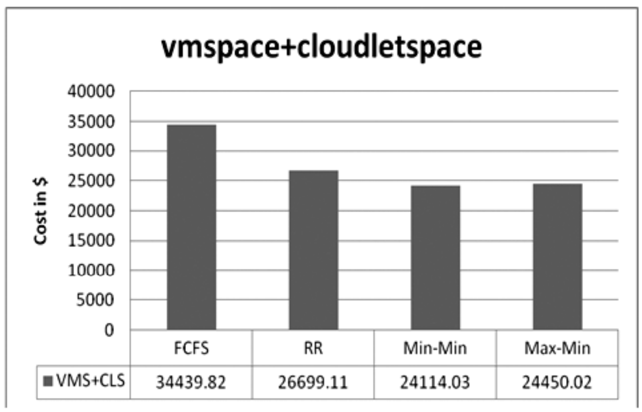

Figure 2: Cost of different Algorithms used for Case I

7.2. Case II: (VM time and Cloudlet space shared)

In this VMs are assigned to the host using RR method and cloudlets are assigned to the VM using FCFS method.

Table 2
Cost Table: VM time and Cloudlet space shared costs

Mapping	*Cost ($)*
FCFS	34439.82
RR	26699.11
Max-Min	18697.28
Min-Min	16611.72

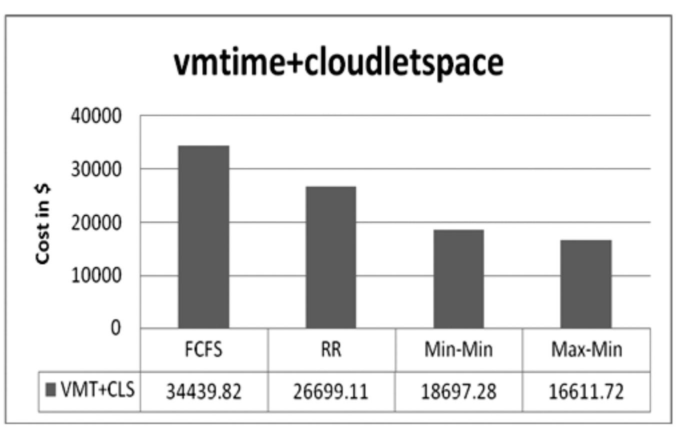

Figure 3: Cost graph for different algorithms used for Case II

7.3. Case III: (VM time and Cloudlet time shared)

In this both VMs assignment to the host and cloudlets assignment to the VM, done by RR method.

Table 3
Cost Table: VM time and Cloudlet time shared costs

Mapping	*Cost ($)*
FCFS	34439.82
RR	26699.11
Max-Min	18697.28
Min-Min	19053.42

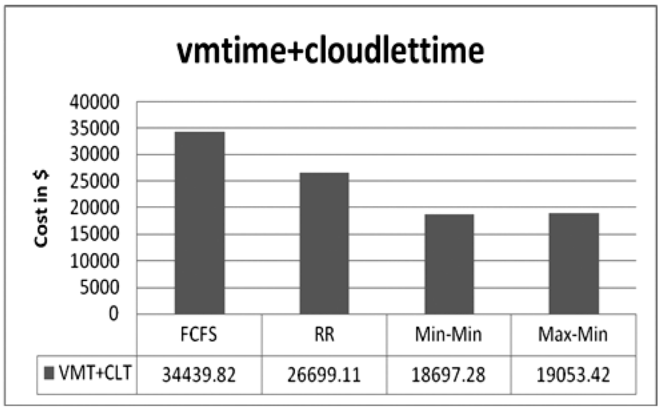

Figure 4: Cost graph of different Algorithms used for III

7.4. Case IV: (VM space and Cloudlet time shared)

In this VMs are assigned to the host by FCFS method and cloudlets are assigned to the VM in RR basis.

It is concluded that for every case FCFS and RR costs are almost equal but Max-Min and Min-Min of every case we get for VM time shred and Cloudlet space shared gives us a better result.

Table 4
Cost table when, VM space and Cloudlet time shared costs

Mapping	*Cost ($)*
FCFS	34439.82
RR	26699.11
Max-Min	28974.22
Min-Min	29315.61

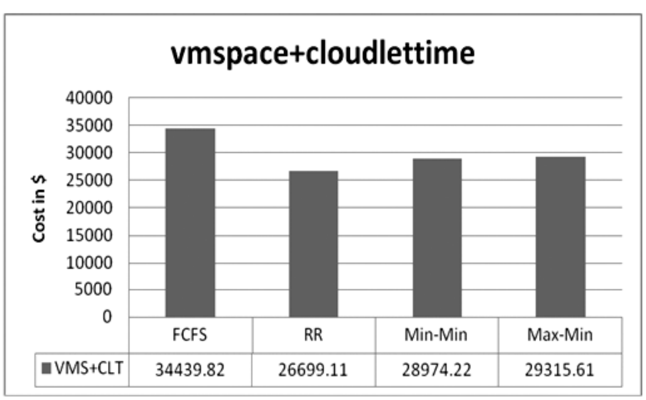

Figure 5: Cost table for different algorithms for IV

QoS depends on the tasks completion time. Time taken by the above mapping schemes are given below. Max-Min mapping gives less cost with minimum completion time to complete the tasks. Hence Max-Min mapping is the best mapping as compare to other three heuristics.

7.5. Case V: (Effects on cost by varying different parameters)

The variations in the costs are recorded by changing number of cloudlets, VMs, and datacenters. For every case, the VM is time shared, and cloudlet is space shared as it gives batter result comparison to all cases.

Table 5
Completion time of Different Algorithms.

Mapping	*Completion Time (ms)*
FCFS	400.10
RR	300.09
Max-Min	116.77
Min-Min	175.09

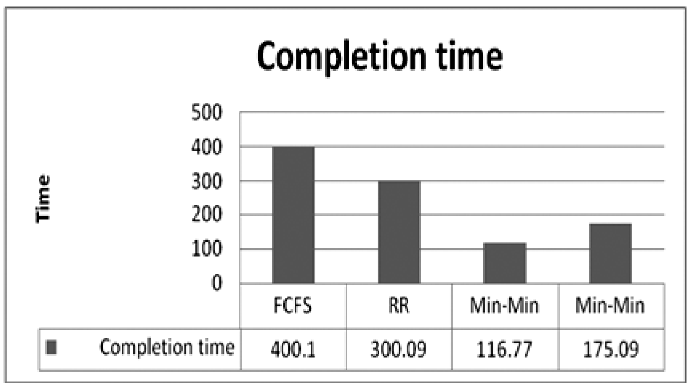

Figure 6: Graph to represent QoS Completion time of tasks

7.6. Case VI: (Effects on cost by changing number of cloudlets, when DC = 1 and VMs = 5).

Table 6
Cost comparison of different heuristics by varying cloudlets.

	Cost Calculation using Different Heuristics ($)			
Cloudlets	*FCFS*	*RR*	*Max-Min*	*Min-Min*
10	34439.82	26699.11	18697.28	16611.72
50	19703563.9	19159588.4	18880870.3	17188691.6
100	322731879.2	317459497.6	318706770.0	299973222.0
1000	3187481360921.91	3285519461377.77	3289465149766.87	3266309025844.44

Table 7
Completion time comparison of different heuristics by varying cloudlets.

	Completion time of Different Heuristics (ms)			
Cloudlets	*FCFS*	*RR*	*Max-Min*	*Min-Min*
10	400.10	300.09	116.77	175.09
50	10000.09	6500.07	5900.09	5250.09
100	41650.08	25500.05	24650.08	23000.09
1000	4166649.8	2504999.3	2499649.9	2480000.0

7.7. Case VII: (Effects on cost by changing number of VMs, when DC = 1 and cl = 50)

Here different mapping heuristics are used by varying the VMs. The effect of completion time and cost are shown in Figure 7 and Figure 8 respectively.

Hence here observed that increase in the number of VMs decreases the cost and compilation time. But when the number of VMs is equal to the number of cloudlets or more from the number cloudlets then the cost never decreases further. It stops when the cloudlets number and VMs number are equal.

Hence it is observed that changes some of datacenters have no affect the cost and completion time per mapping. From the above simulation, it has been observed that Max-Min algorithm provides low cost with less completion time. By varying parameters, it is observed that the increase in a number of cloudlets, increase the total cost for each scheduling technique, as the resource utilization, execution time and waiting time also increases. In other case increase in the number of VMs, decrease the total cost, because waiting time of cloudlets decreases.

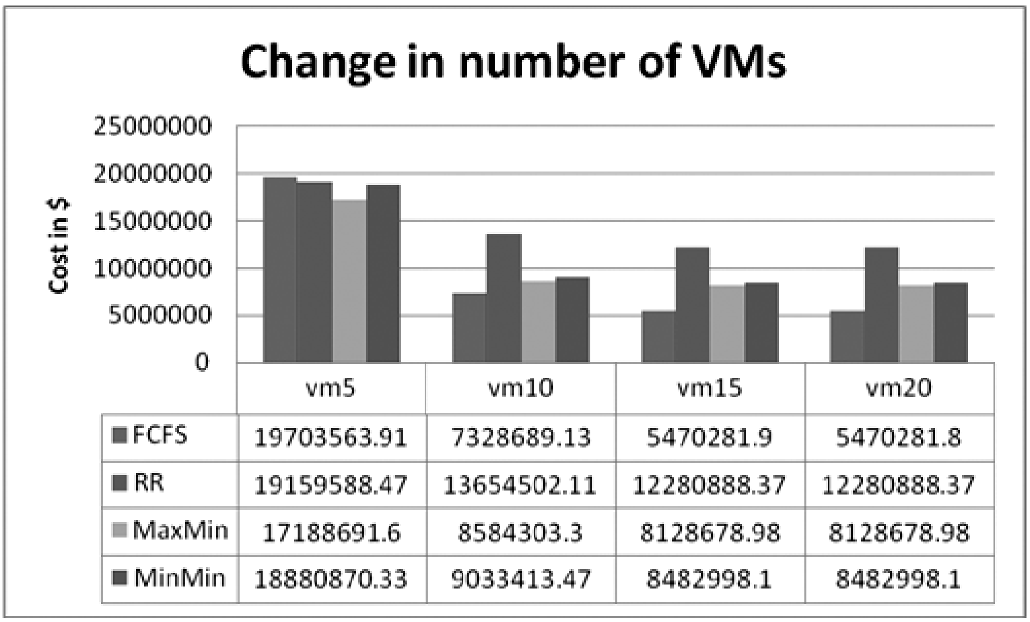

Figure 7: Cost effects due to change in number of VMs

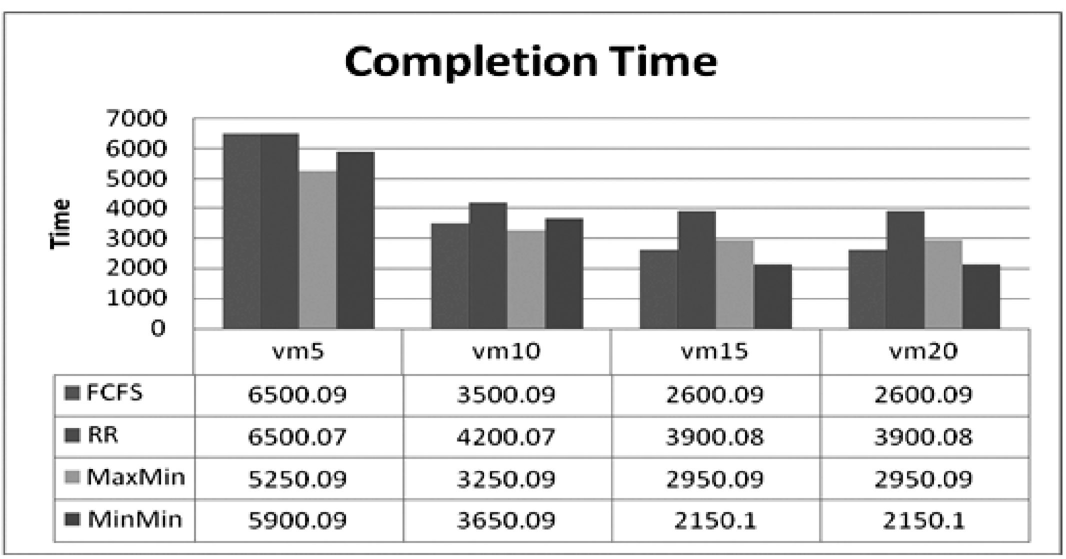

Figure 8: Completion time due to increasing in number of VMs for above four mapping

8. CONCLUSION

In this paper cost minimization is done by considering resource utilization by cloudlets. Here the authors have utilized popular heuristics for optimal mapping of tasks to resources and the proposed generalized cost function has been utilized to calculate the cost of cloud resources.

From the simulation it is concluded that Max-Min gives less cost with minimum completion time. Total cost of cloudlet never decreases further, if the number of cloudlets and virtual machine are equal or number of cloudlets less than the virtual machine. But number of DCs has not affected the total cost. Here we have considered static mapping technique. But in real time task may arise at any instant of time. So the above cost function can be applied on the dynamic arrival of tasks by changing the mapping heuristics.

REFERENCES

[1] C. Weinhardt, A. Anandasivam, B. Blau, N. Borissov, T. Meini, W. Michalk and J. Stosser, “Cloud Computing–A Classification,” Business Models, and Research Directions, Business Models and Information System Engineering, vol. 1, no. 5, pp. 391-395, 2009.

[2] Eldin Suliman M. Taj, Sudan Khartoum, and Hany H. Ammar, “Pricing Models for Cloud Computing Services, a Survey,” vol. 5, no. 3, pp. 126-171, 2016.

[3] C. S. Yeoa, S. Venugopalb, X. Chua and R. Buyyaa, “Autonomic Metered Pricing for a Utility Computing Service,” Future Generation Computer System, vol. 26, no. 8, pp. 1368-1380, 2010.

[4] A. Osterwalder, “The Business Model Ontology – A Proposition in a Design Science Approach,” Doctoral thesis, University of Lausanne, 2004.

[5] L. Xia, K. B. Monroe and J. L. Cox, “The Price Is Unfair! A Conceptual Framework of Price Fairness Perceptions,” Journal of Marketing, vol. 68, no. 4, pp. 1-15, 2004.

[6] Thomas Sandholm and Dongman Lee, “Notes on Cloud computing principles,” Journal of Cloud Computing, Springer, vol 3, no 1, pp. 1-10, 2014.

[7] Sharlene Aminullah, and Carlos Molina-Jimenez, “Cost Estimation of Service Delivery in Cloud Computing,” pp. 26-32, 2012.

[8] Juthasit Rohitratana and Jorn Altmann, “Agent-based simulations of the software market under different pricing schemes for software-as-a-service and perpetual software,” Economics of Grids, Clouds, Systems, and Services, Springer Berlin Heidelberg, pp. 62-77, 2010.

[9] Aarti Singh, Dimple Juneja, and Manisha Malhotra, “A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing,” Journal of King Saud University – Computer and Information Sciences, pp. 1-10, 2015.

[10] Hao Li, Jianhui Liu, and Guo Tang. “A pricing algorithm for cloud computing resources.” In the proceedings of the conference on Network Computing and Information Security (NCIS), IEEE, vol. 1, pp. 69-73, 2011.

[11] Wei Wang, et al, “Datacenter net profit optimization with deadline dependent pricing,” In the proceedings of the annual conference on Information Sciences and Systems (CISS), IEEE, pp. 1-6, 2012.

[12] Amelie Chi Zhou and Bingsheng He. “Transformation-Based Monetary Cost Optimizations for Workflows in the Cloud,” IEEE Transaction on cloud computing, vol. 2, no. 1, pp. 85-98, 2014.

[13] Luca Ferretti, Fabio Pierazzi, and Michele Colajanni, “Performance and Cost Evaluation of an Adaptive Encryption Architecture for Cloud Databases,” IEEE Transaction on cloud computing, vol. 2, no. 2, pp. 143-155, 2014.

[14] Igor Ruiz-Agundez, Yoseba K. Penya, and Pablo G. Bringas, “A flexible accounting model for cloud Computing,” SRII Global Conference (SRII), IEEE, pp. 273-284, 2011.

[15] RM Piro, Rosario M., Andrea Guarise, and Albert Werbrouck, “An economy-based accounting infrastructure for the datagrid,” In the proceedings of the 4th International Workshop on Grid Computing, IEEE Computer Society, pp. 202, 2003.

[16] Bhanu Sharma, Ruppa K. Thulasiram, Parimala Thulasiram and, Rajkumar Buyya. “Clabacus: A Risk-Adjusted Cloud Resources Pricing Model Using Financial Option Theory.” IEEE Trasaction on cloud computing, vol. 3, no. 3, pp. 332-344, 2015.

[17] Mario Macías, and Jordi Guitart, “A genetic model for pricing in cloud computing markets,” In the proceedings of the 2011 ACM Symposium on Applied Computing. ACM, pp. 113-118, 2011.

[18] Adel Nadjaran Toosi, Kurt Vanmechelen, Kotagiri Ramamohanarao and Rajkumar Buyya, “Revenue Maximization with Optimal Capacity Control in Infrastructure as a Service Cloud Markets,” IEEE Transaction on cloud computing, vol. 3, no. 3, pp. 261-274, 2015.

[19] Xingwei Wang, Xueyi Wang, Hao Che, Keqin Li, Min Hung, and Chengxi Gao, “An Intelligent Economic Approach for Dynamic Resource Allocation in Cloud Services,” IEEE Transaction on cloud computing, vol. 3, no. 3, pp. 275-289, 2015.