INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Threshold-Based Workload Control for an Under-Utilized Virtual Machine in Cloud Computing

M. LawanyaShri[1*] Balamurugan Balusamy[2] S. Subha[3]

[1]*SITE, VIT University, India*
[2]*VIT University, Vellore, Tamil Nadu, India*
[3]*VIT University, Vellore, India*
Corresponding author's Email : lawanyaraj@gmail.com

Abstract: Cloud computing empowers sharing of computing resources dynamically to a wide range of end users. The workload on the cloud resources is amassed immensely with the advancement of new applications. To increase the utilisation of the virtual machine in the cloud data center, an efficient load balancing technique is predominantly crucial. Load balancing is the heart of the data centre in cloud environment which makes all the virtual machines accomplish the same amount of workload and helps the virtual machines deliver the services with minimal time delay. To ensure optimum resource usage and fast processing time, M/G/1 is developed with vacation queuing model (VQM-LB) for adjusting the workload of the under-loaded virtual machine in a cloud environment. The proposed system uses vacation and threshold policy to efficiently control the workload level of each virtual machine in the data center , and reduces the energy consumption and cost accordingly. The result obtained in our proposed model assuredly bounces optimum solutions and apparently indicates the triumph of an efficient load balancing of tasks on minimal energy consumption and cost.

Keywords: Cloud Computing; load balancing; virtual machine; M/G/1 queueing model; server vacation; cost function

1. Introduction

Cloud computing is a modern, cutting-edge computing model connected over the Internet. Cloud computing carries the amendments and revolution of the IT industry with its emerged popularisation and applications. Plentiful converging and congruent issues are bumping the further rise of the cloud computing. The growing maturity of the technologies and utilities of the cloud make the users hasten the adoption of the cloud. Cloud providers offer a large scale of cloud resources for computing services dynamically in a profitable way. The mounting demand for various resources makes the cloud technology with virtualization-centric [1]. Resource management in the cloud computing is the most challenging task. The environment provides virtual computing resources that are used to accomplish the user tasks with a minimum completion time and cost. Cloud computing provisions a hugely demanded service for the customers, due to high computing availability, power, scalability and cheap of cost [2]. The huge advancement in a cloud affords different platforms and services by creating virtual machines that support the users to accomplish the tasks within a reasonable period without mislaying the Quality of Service. The cloud service providers have a collection of the abundant data center at distinctive geographical locations, to serve the users request in an excellent way over the Internet [3]. Resource provisioning is mainly concerned about the source of enormous computing resource pools called cloud data center [4]. The data center is an integrated repository either virtual or physical for the storage, information, and management, structured with an extensive amount of hardware [5]. The virtual

machine is a significant component in the cloud data center to execute the tasks allocated to it. Each server in the cloud environment is interconnected and retrieved through virtual machines [6]. Virtual machines promote the resource utilisation and consolidation. The virtual machine should finish the execution of the user task assigned to it as early as possible [7]. The user's task abundantly overloads a virtual machine in the cloud. So there is a need for balancing the load by migrating to the tasks from overloaded to the under loaded virtual machines [8]. The thriving progress of cloud paradigm imposes accurate performance evaluation of the virtual machines in the cloud data center. The Load balancing technique ought to follow three major strides, including finding the load of all virtual machines, adjusting the load, the discovery of virtual machine which is under-utilised and migration [9]. Load balancing strategies are intended to adjust the loads by redistribution of jobs among the virtual machines in the data center [10]. An effective load balancing mechanism should guarantee to reduce the response time, execution time and maximise the utilisation of resources and throughput [11].The data center for the most of the time keeps the virtual machines powered on for the arrival of tasks in order to satisfy the requirements of the user. This leads in consuming more energy and cause the optimisation problem to be more complex in the cloud environment.

To address the load balancing issues, we take advantage of the M/G/1 vacation queuing model to analyse and balance the load among virtual machines to minimise the energy consumption, cost and increase the performance. The energy consumption is reduced by sending the virtual machines to vacations based on the load and threshold value. We here present the cloud data center as M/G/1 system with vacation queuing model. The model allows the cloud data center to balance the underutilised virtual machines. Here our proposed approach considers a single data center with an n number of virtual machines with vacations. The vacations are time periods that the virtual machine does not serve the task in the queue. The proposed approach uses two vacation types; Type1 vacation symbolises the durations of additional non-queue tasks assigned to the virtual machine. Type2 vacation symbolises the non-productive period for the virtual machine such as idle time. We consider two threshold policy used to control effectively the workloads assigned to the virtual machines in the data center. Simulation result shows that the proposed model can effectively balance the loads and can reduce the energy consumption and cost compared to existing FCFS and OLB load balancing algorithms. As the proposed work deals with load balancing algorithms, the pre-eminent and commonly used load balancing algorithms like OLB (Opportunistic Load Balancing) and FCFS (First Come First Services) techniques are studied. The FCFS algorithm dispatches the user's task based on the arrival time. This results in severe fragmentation issue. The OLB algorithm tries to keep each node as busy as possible without considering execution time. It allocates each task to a virtual machine in random order. This results in poor makespan [12].

The remainder of the paper is organised as follows. Section 2 introduces the related work on load balancing in cloud environment based on mathematical modelling. Section 3 presents our Modelling M/G/1 with vacations. Section 4 presents a procedure for workload control policy. Section 5 illustrates the analysis of workload control policy and finally; we gave our conclusion in Section 6.

2. Related Work

B Yang et al. [13] discussed the performance analysis of services based on M/M/m/m+r queueing system for fault recovery in cloud computing. Recovery is considered for both communication links and processing nodes. Precedence rules for subtasks and distribution function of response time for the service are determined. Their technique uses only fixed size $m+r$ of buffer and response time is splintered into execution, service and waiting period and all three are independent with each other, which is impractical according to author's argument. Khazaei et al. [14] presented an analytical model based on M/G/m for evaluating the performance measures such as server utilisation ratio and waiting time for service in cloud servers. At the outset, the service time and inter-arrival time are not exponential and the probability distribution for the response time cannot be attained in the closed form. Miyazawa et al. [15] analysed the performance evaluation of M/GI/s queueing system in the case of service time and inter-arrival time. They presented with several causes for a high degree of accuracy. Kimura et al. [16] elegantly developed a transform free technique for M/G/s queueing system. They developed a method for steady state distribution with finite waiting time. The approximation used in their approach is found based on some heuristics and conservation law. Their innovative model gives approximation in explicit form with easy numerical computation. Bacigalupo et al. [17] focused on

dynamic enterprise computing with financial penalties and SLA hosted in a cloud environment. They coordinated usage of performance estimation for QOS measures for the customers in cloud and cloud framework. Vakilinia et al. [18] developed performance modelling for cloud and analysed trade-offs for switching idle servers on to off for power conservation. Thus they model by distributing the jobs in the cloud, and service time of a job is with the finite and infinite amount of cloud resources. Khazaei et al. [19] presented a novel analytical model for the performance analysis of the cloud data center. Their model was intentionally devised to find the relationship between the servers and the buffer size, and is used for the performance metrics such as blocking probability and mean number of jobs in the server. Their approach fails to focus on energy efficiency and cost of the data center. Bouterse et al. [20] proposed a method to study the cloud computing capacity with time-varying traffic workload by the usage of historical traces. The idle capacity is switched off by simulation. The arriving tasks will be blocked, if there is no resource to serve the task. Rathore et al. [21] presented a broad classification for analysing the migration techniques in a grid computing with various parameters like strength, gap, and research focus, and compared model with the proposed load balancing method along with task migrations. Goyal et al. [22] presented a dynamic ant colony based load balancing technique in grid computing. The technique associates the pheromone with the resources instead of the path. The major objective of the proposed algorithm is to map the jobs with the computing resources to balance the workloads and utilise the resources efficiently. Moradi et al. [23] proposed a novel probabilistic algorithm based on time constraints. Their proposed optimisation procedure is used to reduce the response time based on the best status and earliest completion time. Emerson et al. [24] proposed an efficient migration technique for balancing the load dynamically by Lagrange Multiplier, based on the Euclidean model. Ali et al. [25] discussed a guide for load balancing by moving the task from overloaded machine to the underloaded one which increases the performance effectively. Lu et al. [26] proposed balancing method which focuses only on the extra jobs, and is migrated to underloaded different host virtual machine, using PSO algorithm.

Reni et al. [27] modelled a geometric matrix technique with N-policy vacation queueing model. The service in the service station resumes immediately, after repairment and vacation starts. The matrix-form expression is generated for various system performances. Wu et al. [28] elegantly proposed an M/G/1 model with an exhaustive search and multiple vacations. They defined the model in which the resources work in various service time in random variables, and are distributed based on both service times and the server in vacation.

From the above analysis, the performance evaluation is carried out using queueing model in cloud computing. There are only a few works, which addressed performance issues, using a mathematical model in the cloud . Among that , There is no work which addresses the load balancing issues based on the vacation queueing model with QOS metrics like energy efficiency, cost and throughput. The proposed work moderates energy by sending the virtual machines to multiple vacations, based on the threshold policy using M/G/1 queuing model.

3. Problem Formation:

In this section, we present our proposed model, using M/G/1 queueing model with two vacation types. Our ultimate goal is to distribute the workload among virtual machines evenly and switch over the virtual machines to vacation state, if the load on the particular virtual machine is less than the threshold value. Thereby reduces the energy consumption and costs consequently. The data center in the cloud consist of k physical machines indexed by the set $P=\{p1, p2, ...p_k\}$ hosting of m virtual machines represented by the set $VM= \{v1, v2,\ldots\ldots vm\}$ and d tasks $\{t1,t2,\ldots\ldots t\ d\ \}$. The broker in the data center receives a number of tasks to be executed; the broker in the data center assigns the tasks to each virtual machine, as per the scheduling procedure.

Our proposed approach considers an M/G/1 queueing model where each virtual machine in the data center follows a two vacation types with threshold policies. Let the user task arrive the server system based on Poisson distribution with the λ rate. The service times of each task in the system are independent , in which it is identically distributed with random variables and indicated by S. The proposed approach considers two kinds of vacations for the virtual machine. The Type1 vacation (vacation Type 2) time are exponentially distributed for the random variable Y_1 (Y_2). The Type2 vacation

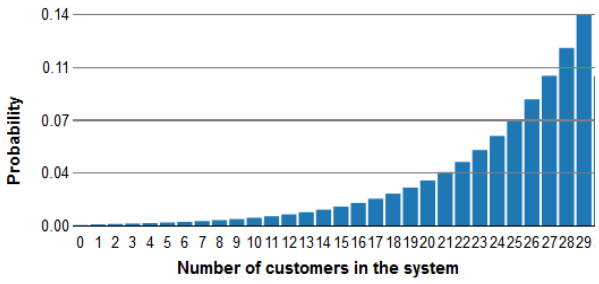

Figure. 1. Steady State Distribution

for the virtual machine is stochastically lesser than Type2. The arrival process, service time and the vacation duration are mutually independent. The virtual machine executes all the tasks till the system gets empty, then the virtual machine initially takes vacation type1. After completing vacation type1 the virtual machine verifies the queue, if the number of tasks is from 0 to n-1, then it meets the lower threshold value. So the virtual machine takes another type 1 vacations. If queue length is greater than or equal to m and the number of tasks is less than m+c, then the virtual machine takes type2 vacation. If the number of tasks is m+c or more that is an upper threshold, then the virtual machines resume from the vacation and execute the tasks.

The vacation strategy confirms that if there is any virtual machine idle, then the virtual machine can go for vacation so that the number of under-utilized virtual machines can be reduced and it reduces the energy consumption and increases the performance. In our approach, the variable c is a decision variable which fully controls the virtual machines to take at least one vacation type 2 after every busy state. The variable c in the proposed model is fixed and evidenced the convexity of the cost function (Zhang et.al. [29]).

We represent the two threshold policy by (m,c). For a random variable N, $F_M(m)=P(M\leq m)$ specifies its probability distribution $\bar{M}= E(M)$ and $M^{(2)}=E(M^2)$ represent its first and second moments. Let $\rho=\lambda \bar{S}$ and $\rho<1$ as an assumption for the stability of the data center. Where $\bar{S} = 1/\mu$. Figure. 1 illustrates the steady state distribution for the number of customers in the system. The proposed system VQM-LB (Vacation Queueing Model – Load Balancing) has one server with n number of virtual machines depicted in figure 2.

An (m,c) cycle is defined as θ_{mc}. It is the random time interlude between two task completion moments in which the system befits to an empty state. The θ_{mc} can be split into three parts. The accumulation period is represented by T_M during the period where M task arrive, R represents forward

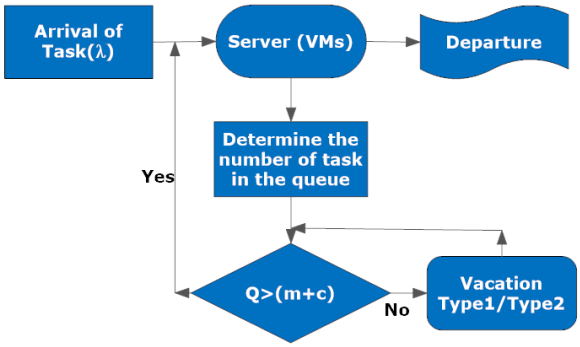

Figure.2 System model

Notation	Meaning
λ	Arrival rate
$\bar{S}$	Mean service time of the virtual machine
$S^{(2)}$	Second moment of the virtual machine.
$\bar{V}_m$	Mean Vacation period where m=1,2
$\rho=\lambda\bar{S}$	Average Server utilisation
RT	The last vacation's residual time at the arrival rate of M^{th} task.
RT_{RT}	Residual life of RT.
hc	cost (holding) of one task in the queue
r_0	Period of vacation start-up
r_m	Reward rate of type of vacation m=1,2
m	Threshold value (Lower)
$M=m+c$	Threshold value (Upper)
c	Difference between the low and upper Threshold value $c=M-m$
g_{mc}	Average cost of long run

recurrence period of the virtual machine's last vacation.

Here A denotes the attending period. Let $\overline{VT_i}$ denotes the average time taken for vacation during an (m,c) cycle, $i = 1, 2$. Let P^1_{mc} and it is the probability for RT being a vacation Type1. The distribution function we have

$$F_{RT}(t) = P^1_{mc}F_{V1}(t) + (1-P^1_{mc})F_{V2}(t) \quad (1)$$

The exponential distribution for Memoryless Property is

$$P^1_{mc} = q^c \quad (2)$$

Where $q = \lambda\bar{V}_1 / (1+\lambda\bar{V})$

$$\overline{RT}_1 = \bar{V}_1, \overline{RT} = q^c\bar{V}_1 + (1-q^c)\bar{V}_2$$

$$\overline{VT}_1 = \frac{m}{\lambda} + \bar{V}_1,$$

$$\overline{VT}_2 = \frac{c}{\lambda} + \overline{RT} - \bar{V}_1 = \frac{c}{\lambda} - (1-q^c)\bar{V}_1 + (1-q^c)\bar{V}_2,$$

$$\bar{\theta}_{mc} = \frac{1}{1-\rho}\left\{\frac{m+c}{\lambda} + q^c\bar{V}_1 + (1-q^c)\bar{V}_2\right\}, \tag{3}$$

The quantity $\overline{RT}_{RT}$ for exponential vacations is given by

$$\overline{RT}_{RT} = \frac{q^c\bar{V}_1}{q^c\bar{V}_1 + (1-q^c)\bar{V}_2}\bar{V}_1 + \frac{(1-q^c)\bar{V}_2}{q^c\bar{V}_1 + (1-q^c)\bar{V}_2}\bar{V}_2 \tag{4}$$

Theorem 1: If *r1≥r2* then the average cost is [30].

$$g_{mc} = \frac{\lambda\rho}{1-\rho}\frac{S^{(2)}}{2\bar{S}}hc + \rho hc + \frac{(m+c)((m+c-1)/(2\lambda) + \overline{RT}) + \lambda RTRT_{RT}}{(m+c)/\lambda + \overline{RT}} + h\frac{r_0(1-\rho)}{(m+c)/\lambda + \overline{RT}} - \frac{r_1(1-\rho)(n/\lambda + V_1)}{(m+c)/\lambda + \overline{RT}} - \frac{r_2(1-\rho)(c/\lambda + \overline{RT} - V_1)}{(m+c)/\lambda + \overline{RT}} \tag{5}$$

It is convex for a fixed variable c in m

Proof:

Using the equation (5), we can calculate the optimal, the nearby integer of

$$m' = -(c + \lambda\overline{RT}) + \sqrt{Y}, \tag{6}$$

Where

$$Y = \frac{2(1-\rho)}{hc}(\lambda r_0 + (r_1 - r_2)(c + \lambda\overline{RT} - \lambda V_1)) + \lambda\overline{RT} + 2\lambda^2\overline{RTRT}_{RT} - \lambda^2\overline{RT}^2$$

Our proposed approach uses this convexity property for a procedure, that determines the threshold policy for controlling loads among virtual machine and balances the system.

4. Procedure Workload Control Policy for Virtual Machines

Load balancing mechanism is applied to distribute the tasks (workload) equally to all the virtual machines in the data center. A good load balancing approach should accomplish resource utilisation and greater user satisfaction ratio [31]. An apt load balancing method aids to minimise resource conception, overhead and maximise scalability.

Our proposed model can work in the situation where the cloud service provider wishes to control the resource utilisation level by allocating some additional load during the idle time. In our approach, the type1 vacation duration is long, compared to type2 vacation. The probability that the system follows at least one vacation type,i.e., type2 can be controlled by c.

The proportion of time spent on the virtual machine [32] on type2 vacation can be denoted by *φ(c, m)* is

$$\varphi(c,m) = \frac{\overline{VT_2}}{\theta_{mc}} = (1-\rho)\left(\frac{c}{\lambda} - (1-q^c)\bar{V}_1 + (1-q^c)\bar{V}_2\right) \times \left(\frac{m+c}{\lambda} + q^c\bar{V}_1 + (1-q^c)\bar{V}_2\right)^{-1} \tag{7}$$

The following result in the behaviour of φ(c,m).

Proposition: (i) The increase in *φ(c, m)* in decision variable c with the upper limit of *(1-ρ)* for m. (ii) The decrease in *φ(c, m)* in decision variable c for m.
Proof: Let us take the partial derivative of *φ(c, m)* with variable c, we have

$$\frac{\partial\varphi(c,m)}{\partial c} = (1-\rho)\frac{[1/\lambda + q^c(Inq)(\bar{V}_1 - \bar{V}_2)][m/\lambda + \bar{V}_1]}{\left\{(m+c)/\lambda + q^c\bar{V}_1 + (1-q^c)\bar{V}_2\right\}^2} \tag{8}$$

can prove the numerator as positive,

$$1 + q^c(Inq)(\lambda\bar{V}_1 - \lambda\bar{V}_2) > 0$$

or equivalently,

$$q^c(Inq)(\lambda\bar{V}_1 - \lambda\bar{V}_2) > -1. \tag{9}$$

Note that

$$q = \lambda\bar{V}_1/(1+\lambda\bar{V}_1) < 1, Inq < 0$$

with c ≥1, we have

$$q^c(Inq)(\lambda\bar{V}_1 - \lambda\bar{V}_2) \geq q(Inq)(\lambda\bar{V}_1 - \lambda\bar{V}_2) \geq q(Inq)\lambda\bar{V}_1 = \left(1 - \frac{1}{1+\lambda\bar{V}_1}\right)(\lambda\bar{V}_1)In\left(1 - \frac{1}{1+\lambda\bar{V}}\right) \tag{10}$$

Taylor series is used for expansion

$$\text{In}\left(1 - \frac{1}{1+\lambda\bar{V}_1}\right) = \sum_{m=0}^{\infty}\frac{(-1)^m}{m+1}\left(\frac{-1}{1+\lambda\bar{V}}\right)^{m+1}$$

$$= -\frac{1}{\lambda \bar{V}_1} \quad (11)$$

From the equation (10) and (11),

$$q^c(Inq)(\lambda\bar{V}_1 - \lambda\bar{V}_2) \geq -\left(1 - \frac{1}{1+\lambda\bar{V}_1}\right) \geq -1.$$

If the time interval of type2 vacation is a non-productive one, then *1-φ* is a very effective virtual machine utilisation level. Thus, the variable c controls the utilisation level of the virtual machine. Based on the two thresholds *(c, m)* policy, at least one type 1 vacation can be taken by the virtual machine. The expected number of type1 vacations taken by the virtual machine during each *(m, n)* cycle is symbolised by

$$E(M_1) = \frac{1}{1-\rho}\left(\frac{m}{\lambda\bar{V}_1} + 1\right) \quad (12)$$

The Resource Manager in cloud computing data center usually designs a service policy based on many constraints at min average cost. The following procedure is used for searching the best optimal solution that satisfies two constraints. The expected number of the additional tasks is to be executed after every busy period, which is more compared to the minimum number χ, and the virtual machine utilisation is not below the minimum δ level.

Procedure: defining optimal two threshold policy

1. From equation (12) for determining the min $m=m_0$, such that the min expected number of type1 vacation by the virtual machines, after every busy state. After completing the execution of the task (T) in a virtual machine (VM), the virtual machine takes the Type1 vacation.

2. For the static m_0,using the equation (7) to find $c=c_0$, *a* (m_o,c_o) policy provides the effective utilisation of the virtual machine in the cloud data center.

3. To compute the lower threshold m^* at c_0, the equation (6) is used. Based on the convexity property, if $m^* \geq m_0$ then m^* is the optimal threshold. If $m^* \leq m_0$ then the feasible and optimal policy is (m_0, c_0) for c_0.

5. Performance Analysis

The simulation of the queuing system is done using SHARPE tool. The performance evaluation of the proposed model (VQM-LB) is analysed. We have analysed the proposed model, using SHARPE tool, and it is obvious that the M/G/1 with vacation policy (m,c) controls the workload of under-utilised virtual machines. The simulation of this type is analysed to envisage the Quality of Service metrics such as response time, cost and utilisation of the system. The proposed model is compared with traditionally proven algorithms like FCFS (First Come First Serve) and OLB (Opportunistic Load Balancing).

Figure 3 and Figure 4 illustrate the VQM-LB efficiently increases the performance compared to the standard FCFS and OLB algorithm. Figure 3 shows the CPU utilisation time based on the number of jobs, in the cloud data center. The VQM-LB attains less CPU utilisation time for more number of jobs compared to FCFS and OLB.

Figure 4 shows the Response time based on the number of jobs in the data center. The response time increases, when the arrival rate of the task increases in the data center. Our proposed approach stabilises the workload to reduce the response time accordingly. Thus, the comparison result proves that VQM-LB is more efficient and performs well, when the number of jobs in the data center increases. Figure 5 depicts how the cost of the data center is reduced based on the service time distribution.

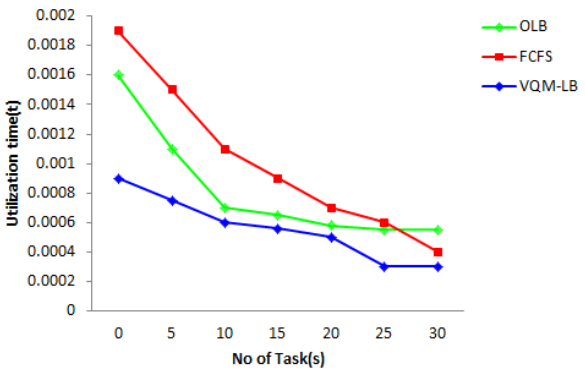

Figure.3 Utilisation time

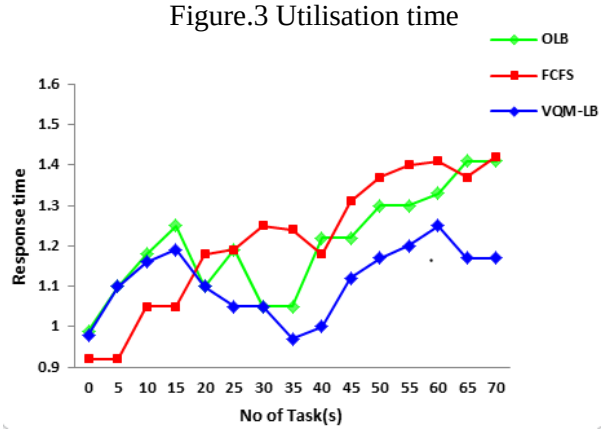

Figure .4 Response time

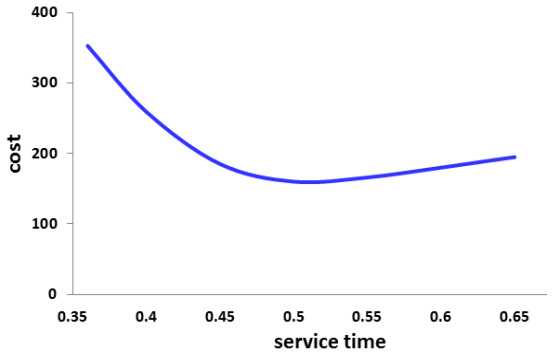

Figure.5 Effect of cost based on service time

6. CONCLUSION

In this paper, we have proposed a model for load balancing the virtual machines in the cloud data center. Our proposed approach takes advantage of M/G/1 with vacation policy to control the workloads of under-utilised virtual machines and reduce the overall cost of the data center. The model uses two types of vacations with threshold policy. We simulated our proposed model, using SHARPE tool. The result analysed by VQM-LB shows enhancement in the performance of the cloud queueing system. The performance is measured by using utilisation, cost and response time. Simulation result shows the fact that utilisation rate has increased by 15% compared to the existing model, and increases the performance effectively. In future, we intend to extend our work with the other QOS factors such as fault tolerance and network traffic information in a cloud environment with multiple vacations.

References

[1] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg and I. Brandic, "Cloud computing and emerging it platforms : vision , hype, and reality for delivering computing as the 5th utility", *Future Generation Computer Systems* , pp. 599 – 616, 2009.

[2] I. T. Foster, Y. Zhao, I. Raicu and S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared " , *In: Grid Computing Environments Workshop*, pp. 1–10, 2008.

[3] B. Balusamy, J. Sridhar, D. Dhamodaran and P. Venkata Krishna, "Bio-inspired algorithms for cloud computing: a review", *International Journal of Innovative Computing and Applications*, Vol. 6, No. 3-4, pp. 181-192, 2015.

[4] R. A. Haidri, C. P. Katti, and P. C. Saxena, "A load balancing strategy for Cloud Computing environment ", In: *Proc. Of International Conf. on Signal Propagation and Computer Technology*, IEEE, pp. 636-641, 2014.

[5] B . Balusamy, "Extensive survey on usage of attribute based encryption in the cloud ", *Journal of Emerging Technologies in Web Intelligence*, Vol. 6 , No. 3 , pp. 263-272, 2014.

[6] B. Balusamy, and P. V. Krishna, "Collective advancements on access control scheme for the multi-authority cloud storage system", *International Journal of Grid and Utility Computing* , Vol. 6, No. 3-4 , pp. 133-142 , 2015.

[7] B. Sotomayor, S. R. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds ", IEEE *Internet computing*, Vol. 13, no. 5, pp. 14-22, 2009.

[8] A.S Milani, and N.J. Navimipour, " Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends ", *Journal of Network and Computer Applications*, pp. 25-35, 2016.

[9] K. Li, "Optimal load distribution for multiple heterogeneous blade servers in a cloud computing environment ", *Journal of Grid Computing* , Vol. 11. No. 1, pp. 27- 46, 2013.

[10] Y. Liu, C. Zhang, B. Li, and J. Niu , " DeMS : A hybrid scheme of task scheduling and load balancing in computing clusters ", *Journal of Network and Computer Applications*, 2015, http:// dx.doi.org /10.1016/ j.jnca.2015.04.017.

[11] K. Karthikeyan, B. Balamurugan, and A. K. Sangaiah, "Ant colony based load balancing and fault recovery for cloud computing environment", *International Journal of Advanced Intelligence Paradigms* (In production)(2016).

[12] K. Xiong and H. Perros, "Service performance and analysis in cloud computing", In : *Congress on Services-I*, IEEE, pp. 693-700, 2009.

[13] K. Al. Nuaimi, N. Mohamed, M. Al. Nuaimi , and J. Al. Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms", In *Network Cloud Computing and Applications (NCCA), Second Symposium on*, IEEE, pp. 137-142, 2012.

[14] H. Khazaei, J. Misic and V. B. Misic ", Performance analysis of cloud computing centers using m/g/m/m+r queuing systems ", *IEEE Transactions on parallel and distributed systems*, pp. 936-943, 2012.

[15] M. Miyazawa, "Approximation of the queue-length distribution of an M/GI/s queue by the

basic equations", *Journal of Applied Probability*, pp. 443-458 , 1986.

[16] T. Kimura, " A transform-free approximation for the finite capacity M/G/s queue ", *Operations Research* , Vol. 44, No. 6 , pp. 984-988, 1996.

[17] D.A. Bacigalupo , J. Van Hemert , X. Chen , A. Usmani, A.P. Chester , L. He, D.N. Dillen , G.B. Wills ,L. Gilbert and S.A. Jarvis, "Managing dynamic enterprise and critical workloads on clouds using layered queuing and historical performance models ", *Simulation Modelling Practice and Theory* , Vol. 19 , No. 6 , pp.1479-1495, 2011.

[18] S. Vakilinia, M.M . Ali and D Qiu, " Modeling of the resource allocation in cloud computing centers ", *Computer Networks*, Vol. 91 , pp.453-470 , 2015.

[19] H. Khazaei, J. Misic and V. B Misic, " Performance analysis of cloud computing centers using m/g/m/m+r queuing systems ", *IEEE Transactions on Parallel and distributed systems*, Vol. 23 , No. 5, pp. 936-943, 2012.

[20] B. Bouterse and H. Perros , " Scheduling cloud capacity for time-varying customer demand ", In *Cloud Networking (CLOUDNET), 1st International Conference on*, IEEE, pp. 137-142, 2012.

[21] N. Rathore and I. Chana , " Load balancing and job migration techniques in the grid : a survey of recent trends ", *Wireless Personal Communications* , Vol.79 , No. 3, pp. 2089-2125, 2014.

[22] S. K. Goyal and M . Singh ,. " Adaptive and dynamic load balancing in grid using ant colony optimization", *International Journal of Engineering and Technology* , Vol. 4, No. 9 , pp. 167, 2012.

[23] M. Moradi, M. A. Dezfuli and M .H. Safavi, "A new time optimizing probabilistic load balancing algorithm in grid computing", In *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, IEEE , Vol. 1, pp. V1-232., 2010.

[24] Y. F. Hu, R. J. Blake and D . R. Emerson ," An optimal migration algorithm for dynamic load balancing", *Concurrency-Practice and Experience* , Vol. 10 , No. 6, pp. 467-483, 1998.

[25] A. M Alakeel, "A guide to dynamic load balancing in distributed computer systems ", *International Journal of Computer Science and Information Security* , Vol. 10, No.6, pp. 153-160,2010.

[26] Y. Lu, Q. Xie , G. Kliot , A. Geller, J. R. Larus and A. Greenberg , " Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services " , *Performance Evaluation* , Vol. 68 , No. 11 , pp. 1056-1071, 2011.

[27] M. R. S. Raj and B. Chandrasekar, "Matrix-Geometric Method for Queueing Model with Subject to Breakdown and N-Policy Vacations", Vol. 5, No.5 , pp. 917-926 ,2015.

[28] D. A. Wu and H. Takagi ", M/G/1 queue with multiple working vacations ", *Performance Evaluation* , Vol. 63, No. 7, pp. 654-681, 2006.

[29] Z.G. Zhang, " On the convexity of the two-threshold policy for an M/G/1 queue with vacations", *Operations research letters* , Vol. 34, No. 4, pp. 473-476, 2006.

[30] Z. G. Zhang, "A note on workload control for an under-utilized server of M/G/1 system", *Computers & Industrial Engineering* , Vol. 56, No.1 , pp. 28-32, 2009.

[31] Z. G. Zhang and C. E. Love , " The threshold policy in an M/G/1 queue with an exceptional first vacation" , *INFOR: Information Systems and Operational Research* , Vol.36 ,No.4, pp.193-204, 1998.

[32] Z.G. Zhang, R. G. Vickson and M.J.A van Eenige , " Optimal two-threshold policies in an M/G/1 queue with two vacation types " , *Performance Evaluation* , Vol.29, No.1, pp.63-80, 1997.

[33] K.S. Trivedi and R Sahner, " SHARPE at the age of twenty two", *ACM SIGMETRICS Performance Evaluation Review* , Vol. 36, No. 4, pp. 52-57, 2009.