Jr. of Industrial Pollution Control 33(2)(2017) pp 1547-1554
www.icontrolpollution.com
Research Article

DYNAMIC RESOURCE ALLOCATION IN CLOUD USING QUEUING MODEL

VETHA S[1*] AND VIMALA DEVI K[2]

[1]Research Scholar, Bharathiar University, Coimbatore, Tamilnadu, India

[2]Department of Computer Science and Engineering Velammal Engineering College, Chennai, Tamilnadu, India

(Received 07 August, 2017; accepted 20 September, 2017)

Key words: Cloud computing, Global scheduler, Queuing model, Resource allocation

ABSTRACT

Cloud computing system offers on-demand computing services to the users in a pay-per-use basis. The data center involves a large number of resources and it is highly scalable and flexible. As there is an enormous amount of users to access the data center, the resource allocation plays a significant role. In order to utilize the resources in an efficient manner, a resource allocation model is necessary to regulate the Virtual Machines (VMs) available in the data center. A scientific model utilizing E-M/M/1/K queue model based scheduling and allocation of resources is proposed. The E-M/M/1/K queue model ensures the quality of a cloud computing environment by considering the resources accessibility. This enhances the nature of service by increasing the throughput and minimizing the job execution time and waiting time to fulfill the necessities of clients. The resource allocation algorithm is developed, by considering some useful proprieties of queuing theory. Performance analysis results indicate that the proposed queuing model increases the utilization rate of global scheduler and yields high throughput and minimum waiting time.

INTRODUCTION

Cloud computing system offers reliable services to the users based on their service requests. The users should pay for the amount of resources availed by them. The main goal of the cloud computing system is to minimize the capital and maintenance cost by eliminating the process of the owning the resources. It supports the users by providing the equipment and programming resources required for them. The cloud data centers comprise a group of configurable resources including networks, applications and servers. The data center of the cloud computing system should satisfy the demands of the user to gain reputation and profit. A Service Level Agreement (SLA) should be signed between the cloud users and service providers to enter into an agreed upon business environment. The requirements of the users will be clearly specified in the SLA, which must be strictly followed (Ghosh, 2012). The SLA is necessary to attain the Quality of Service (QoS) and to validate whether the QoS is met. The QoS measures include parameters such as accessibility, throughput, dependability, security, and other performance markers such as response time, likelihood of service promotion, and mean number of errors in the framework, etc. Cloud computing gained its popularity in the recent years due to the offering of services for a short time at an affordable cost. Infrastructure-as-a-Service (IaaS) offers an environment or infrastructure in an on-demand-basis, where there is no need to buy the required number of servers or networks for scaling up the computing resources. This reduces the capital cost, maintenance cost and administration cost of the owner. The Platform as-a-Service (PaaS) offers a platform to build, execute and run theapplications without the need of installing a licensed operating system. It is helpful for the web developers and software developers, who can utilize the upgraded versions of a new platform. The Software-as-a-Service (SaaS) offers the application software for editing and saving the files and documents online. It reduces the storage management cost and security risks of the

***Corresponding authors email: vetha.s@gmail.com**

users. The classification of the cloud deployment models, including public cloud, private cloud, hybrid cloud and community cloud (Oumellal, *et al*., 2014). To attain a good provisioning environment, it is necessary to utilize all the resources in an efficient way using the scheduling and resource allocation procedures. Queuing theory is one of the scientific models used to schedule the resources in a queue for better allocation. The users of the requested services should wait in the queue to get their need satisfied. The fundamental queuing process comprises of clients touching base at a queuing framework to get some service. Once the servers in the data center complete their task of servicing the allocated request, they start serving the requests that wait in the queue. The services are offered to the users waiting in the queue in a certain perspective to make the servicing balanced.

A cloud has a number of servers connected as hubs, where thousands of requests will arrive from the user and they are queued in the system based on its framework.

Due to the dynamic nature of cloud situations, assorted qualities of user demands and time dependency, the cloud focuses on the provision of expected nature of service (Khazaei, *et al*., 2012).

Efforts have been attempted to determine the proficient path to process the requests of the clients in a fast and productive way. In order to acquire the most effective system, the two sort of frameworks such as single server framework M/M/1 and multi-server framework M/M/C are evaluated using the waiting time of the client (Sowjanya, *et al*., 2011; Mohanty, *et al*., 2014). A resource allocation model is proposed to evaluate the performance of cloud system at heterogeneous situations. In this model the requests are placed in the queue by following the Poisson process. The performance parameters including utilization rate, throughput, delay and number of job requests are taken into account (Satyanarayana, *et al*., 2013). A queuing model, namely, M/G/1:∞/ GD Model is introduced to allocate the resources to the clients in the cloud computing environment. This model enhanced the productivity of the cloud environment by satisfying the user's need (Bharathi, *et al*., 2012).

The waiting queue is modified to incorporate priority based resource allocation in the cloud. The utilization rate is increased using several SLA parameters including memory, transfer speed and CPU utilization time. The preemptive execution improved the resource assignment strategy (Pawar and Wagh, 2013). This work is an extension of the work done by Lugun (Li, 2009) to improve the resource allocation process. The QoS parameters required by ever client are considered for the non-preemptive allocation of resources. This method also improves the benefits of the service suppliers from the offered resources. In order to ensure the QoS parameters, an element provisioning strategy is suggested to adjust the various workload changes in the cloud environment. The performance is analyzed to observe the utilization of resources for satisfying the demands of the cloud users. The workload data is reduced to serve all the requests of the clients (Bheda and Lakhani, 2012; Jin, *et al*., 2013). Numerous virtual machines (VMs) with different working frameworks are employed to resolve the resource allocation problems, thereby ensuring the QoS parameters in virtualized situations like Cloud and Grid systems. The resource allocation is made efficient by concentrating on the CPU, memory and system allocation. In the cloud computing environment, a group of virtualized computational resources is provided as a service to the users through the Internet. The issues in the VMs and the resource allocation algorithms are addressed to enhance the QoS in the cloud system. A Stochastic Hill climbing calculation is proposed for the allocation servers or VMs to process the user requests (Mondal, *et al*., 2012). The rest of the paper is organized as follows. Proposed E/M/M/1/K queuing model is explained in section II. Section III presents the performance analysis of the proposed queuing model. Conclusion of the proposed work is discussed in section IV.

PROPOSED E/M/M/1/K QUEUING MODEL

Queuing Model

In the cloud computing system, there is a ton of clients who access the cloud service. The general cloud computing architecture is shown in (Fig. 1). Generally, a cloud computing environment comprises a huge number of cloud users and a set of cloud providers to service the requests. The data center consists of a group of computing resources to fulfill the requirements of the users. It acquires different types of service requests from various clients. Then, it offers the services by charging a certain amount depending on the service time and utilization of resources.

The cloud computing model can be mapped as a queuing model. It is assumedthat there are 'n' number of requests and 'm' number of demands in a cloud infrastructure. Since the continuous arriving

requests might be sent from two distinct clients, the inter-arrival time is an arbitrary variable, which can be demonstrated as an exponential irregular variable in cloud computing. In this manner, the entries of the requests take after a Poisson Process with entry rateλ_i. Demands in the scheduler's queue are dispersed to various computing servers and the scheduling rate relies on upon the scheduler. Assume that there are *m* computing servers, signified as Service 1, Service 2, Service *i*, and Service *m* in the server center.

The service rate is μ_i. So, the average arrival rate is $\lambda\sum_{i=1}^{n}\lambda_i$ and the average service rate is $\mu=\sum_{j=1}^{m}\mu_j$. Here, it is demonstrated that the framework is steady, when $\lambda/\mu< 1$. The rate of service necessity takes after the Poisson Process; it is the same as the client arriving rate. Along these lines, the E-M/M/1/k queuing model is fit for the cloud computing model.

Queuing Theory

In the cloud computing environment, the requests are sent by the user to the cloud application which is facilitated in DC. These requests are handled by a heap balancer or the occupation scheduler to disperse these requests among VMs. The main intent is to utilize the servers at a maximum level by the efficient sharing of resources, thereby reducing the waiting time and cost of purchasing the additional resources. Once the service requests are received from the client, the response time for the requests is assessed using the queuing theory. When compared to the Machine Learning (ML) and Artificial Intelligence (AI) techniques, the queuing theory provides more precise and accurate results. The formula and steps of the AI techniques require high computational overhead than the formulas for the queuing theory.

E/M/M/1/K Queue-Analytical Model

The queuing framework model, illustrated in (Fig. 2

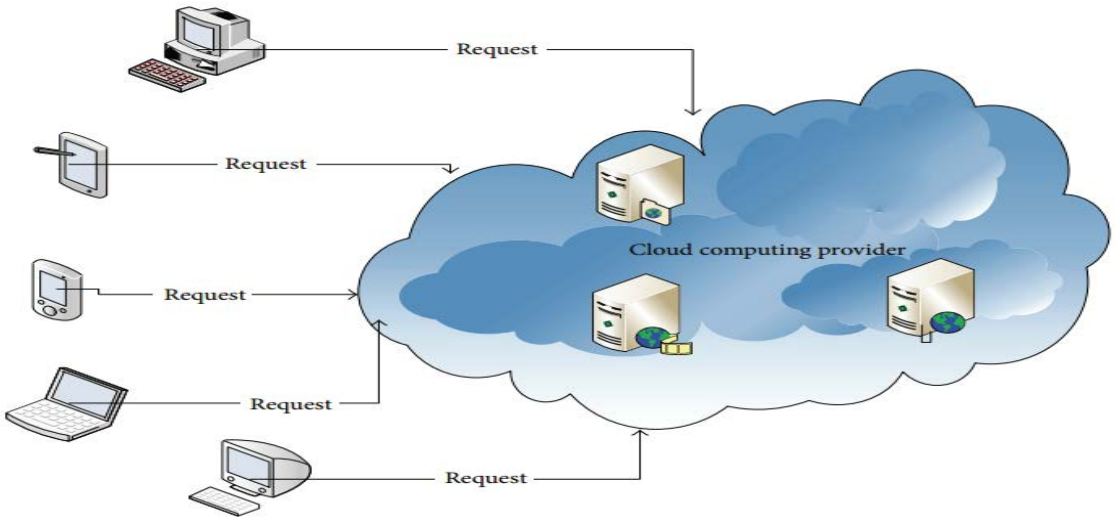

Fig.1 Cloud computing service model.

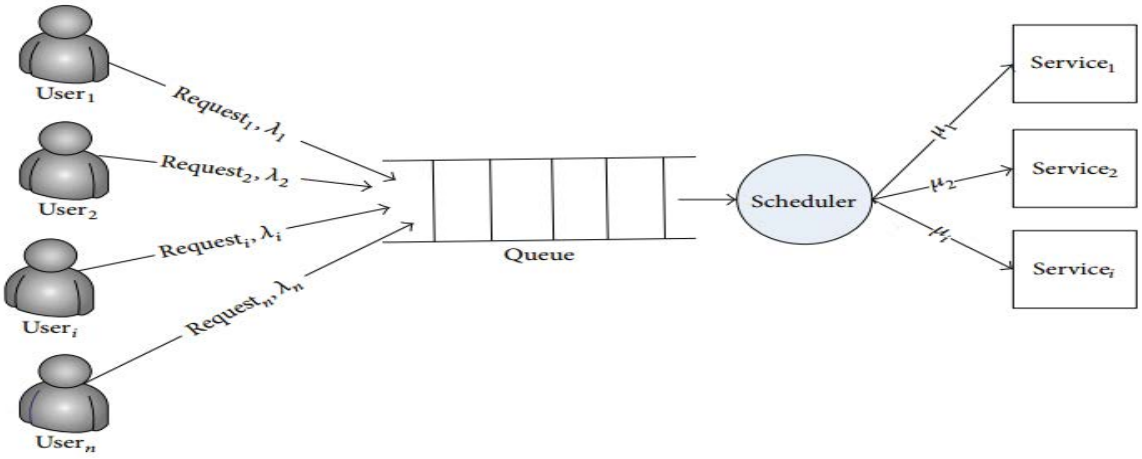

Fig.2 Queuing performance mode in cloud computing.

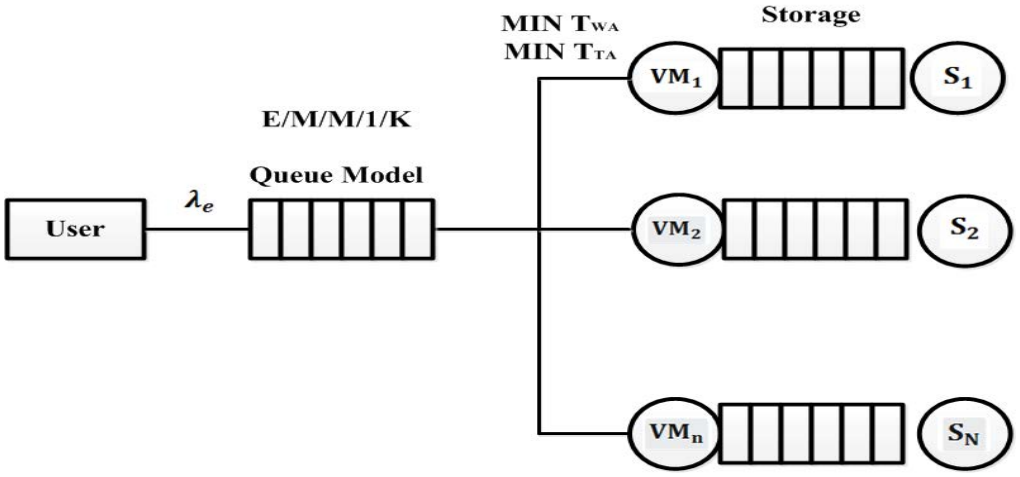

Fig.3 Queuing system model.

and 3), outlines the E-M/M/1/K queuing framework. The (Fig. 2 and 3) demonstrates a Resource Allocator (RA) and VMs that are distributed in a specific cloud application. Each VM in the proposed queuing model offers service to the clients. The queuing framework incorporates RA queue model and VM queue models. The requests in the RA queue are passed along the queues of the VMs one by one in a consecutive manner. The time spent by the request in the RA queues is termed as RRA, whereas the time taken to service the request is termed as RVM.

E-M/M/1/K model properties are (Al Qayedi, *et al*., 2015):

- Poisson entry for approaching request (λ).
- Inter-arrival time of requests is exponential with the rate μ. Number of services is Poisson Process with rate μ.
- The size of the queue is finite. So, the incoming request into a complete queue (buffer) is rejected.
- First-In-First-Out (FIFO) for the entry request to the queue.

μ_{RA}- Mean service rate of RA (requests/seconds).

μ_{VMS} - Mean service rate for load server (requests/seconds).

λ_{RA}- Request arrival rate for RA.

λ_{VM} - Request arrival rate for load server.

ρ_{RA}- Traffic rate of resource allocation.

ρ_{VMS} - Traffic rate of load server

St_{RA} - Service time of RA.

St_{VMS} - Service time of server.

R - Mean response time (seconds).

D_{total}- Total delay.

D_{in} - Ingress latency.

D_{out} - Egress latency.

Throughput-Mean number of requests during the timely request.

The response time for the queuing system is calculated as follows:

$$R = \frac{N}{Throughput} \tag{1}$$

Where $\overline{N}$ denotes mean number of requests in the system. It is defined as

$$\overline{N} = \begin{cases} \dfrac{p[1-(k+1)p^k + Kp^{K+1}}{(1-p)(1-p^{K+1})} & ,if\ \lambda = \mu \\ \dfrac{K}{2}, \text{if } \lambda \neq \mu \end{cases} \tag{2}$$

Where $\mu = \dfrac{1}{St}$ (3)

And $\rho = \dfrac{\lambda}{\mu}$ (4)

Where K is a buffer size based on the Poisson process rate

$$\text{Throughput} = \frac{\lambda}{1-Pn} \tag{5}$$

Where N is thenumber of resources and Pn is the probability of 'n'number of service requests. So,

$$P_n = \begin{cases} \dfrac{(1-\rho)\rho^n}{1-\rho)^{K+1}} & ,if\ \lambda = \mu \\ \dfrac{1}{K+1} if\ \lambda \neq \mu \end{cases}$$

Resource Allocation

The main objective of the resource allocation process is to minimize the waiting time at the cloud service

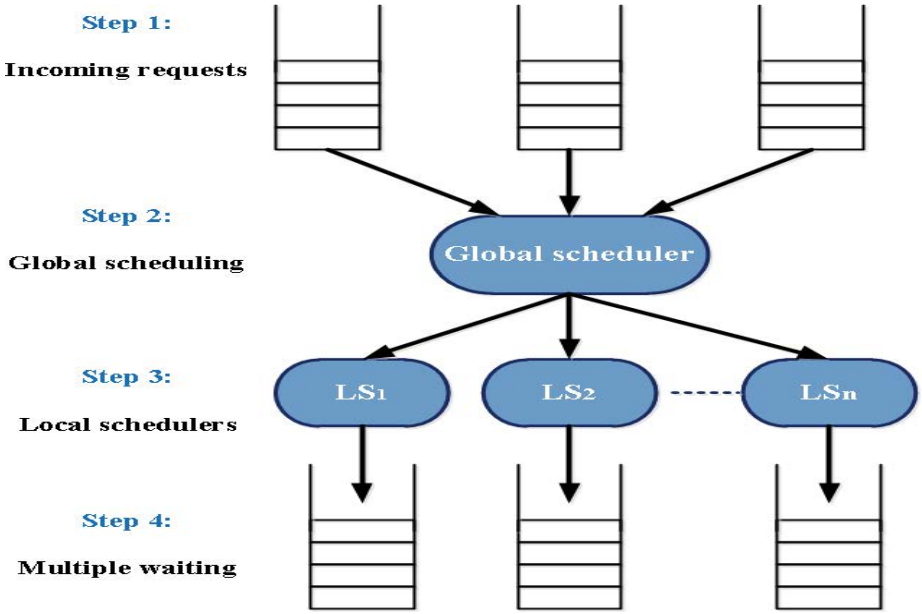

Fig.4 Proposed flow.

provider by decreasing turnaround time and waiting time for every queue as exhibited in (Fig. 4). The proposed resource allocation algorithm achieves the objective by including a number of waiting queues, which in turn services the request in a quick manner. Hence, the turnaround time and the waiting time of all the requests are minimized. The resource allocation algorithm starts with examining unique waiting queues into sets and it places the requests in the next neighbourhood scheduler to reduce the turnaround time and waiting time. This turnaround time and waiting time figured from all base nearby scheduler will be joined in another choice framework with low conflicting. In the initial step of the algorithm, each one of this waiting queue is assigned to one nearby scheduler that worksbased on the queuing technique. In the next step, the lower and upper waiting time for every queue are processed. In the final step, the waiting job is mapped to the suitable queue.

Algorithm: Resource Allocation

Input: Multiple Waiting Queue

Output: Server selection based on queue model

// Initialization

1. For k=1:N do

2. Create set Qi by sampling Q/N //Number of files

3. Calculate arrival time

// λe = λ.Pr {an arrival enters the system}

4. Calculate burst time

$$L_s = \sum_{n=0}^{k} n.P_n$$

$$//\, L_s = \lambda_e.W_s \Leftrightarrow W_s = \frac{L_s}{\lambda_e}$$

5. Calculate completion time

$//T_c = \lambda_e + W_s$

6. Calculate turnaround time

$//T_{TA} = T_c - \lambda_e$

7. Calculate waiting time

$//Tw = T_{TA} - W_s$

8. Server selection

$//If\ min\ (T_{WA} \&\& T_{TA})$

//End if

9. Average time

//Average waiting time

$$T_{WA} = \sum_{WAJ=1}^{m} T_{WAJ}$$

//Average turnaround time

$$T_{TA} = \sum_{TAJ=1}^{m} T_{TAJ}$$

10. Calculate utilization

U = Pr{ n > 0 } = 1 − P0

Where P0 = P1 + P2 + P3 + … + Pk;

11. End for

EXPERIMENT AND RESULT

In the simulation process, it is assumed that "n=5", for queuing model and the quantity of service facility in the system "k=3". Table 1 demonstrates the consequence of the simulations for the single server system, multiple server system and proposed strategy. It shows that the throughput of the proposed system is higher than the throughput of the single server and multi-server systems. The throughput is defined as the time taken to provide the service. The delay is defined as the amount time that a request has to wait in the queue before processing.

Table 1. Performance comparison of delay

Arrival Rate (No. of. Processes)	M/M/1 (ns)	M/M/C (ns)	E-M/M/1/k (ns)
1	0.0600	0.0268	0.0098
2	0.0300	0.0134	0.00481
3	0.0161	0.00667	0.000214
4	0.02280	0.0112	0.00532
5	0.0309	0.0147	0.00787

From (Fig. 5), it can be understood that the waiting time or delay in the E-M/M/1/K is lower than the M/M/1 and M/M/C models. It is observed that the plots of existing models continue expanding, whereas the E-M/M/1/Ki increases after the slight decrement, however the increase is moderate. The throughput comparison of the proposed and the existing models are illustrated in (Fig. 6). Table 2 depicts the performance comparison of throughput. It is demonstrated that the throughput of the E-M/M/1/K model is very higher, when compared to the conventional queuing models. The throughput

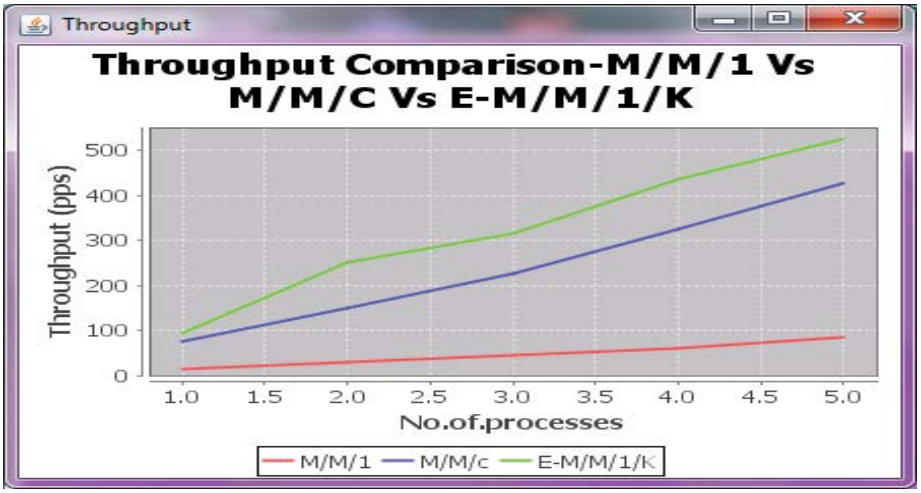

Fig.5 Performance comparison of delay.

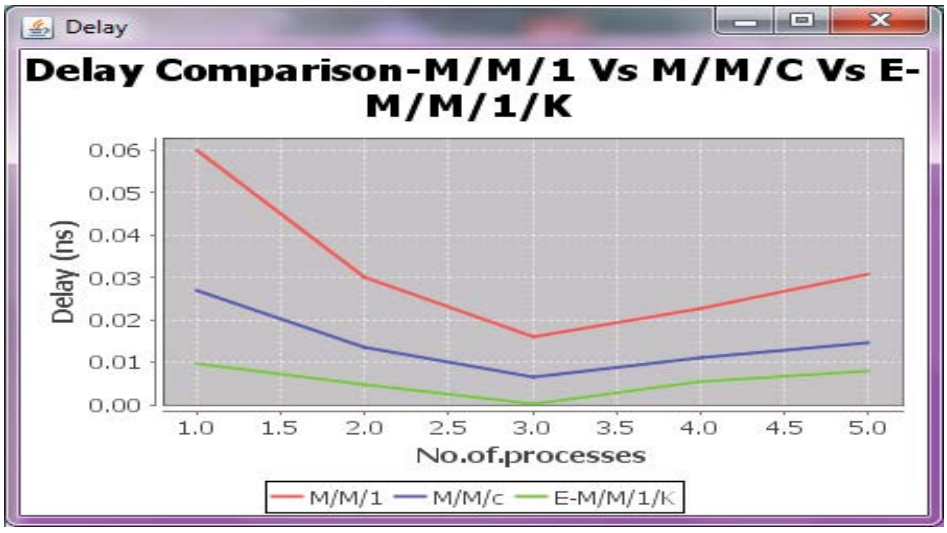

Fig.6 Performance comparison of throughput.

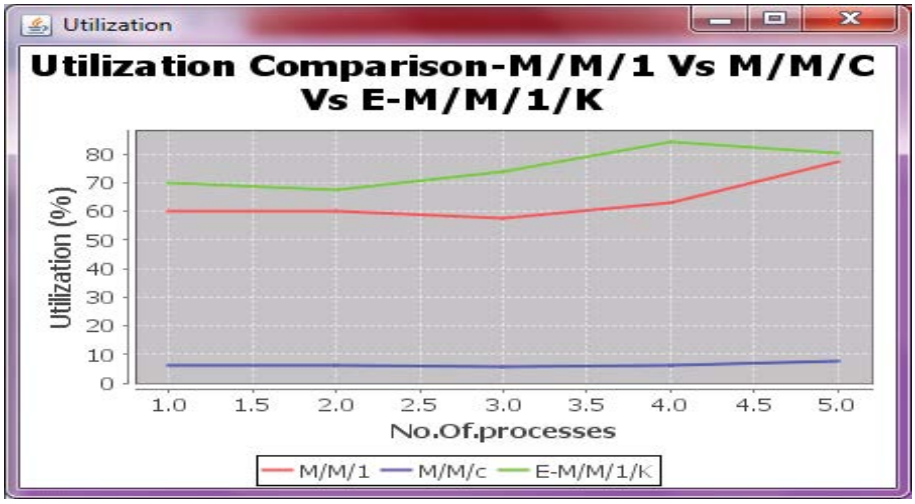

Fig.7 Performance comparison of resource utilization rate.

Table 2. Performance comparison of throughput

Arrival Rate (No. of. Processes)	M/M/1 (pps)	M/M/C (pps)	E-M/M/1/k (pps)
1	15	75	95
2	30	150	250
3	45	225	315
4	60	325	435
5	85	425	525

Table 3. Performance comparison of resource utilization rate

Arrival Rate (No. of. Process)	M/M/1 (%)	M/M/C (%)	E-M/M/1/k (%)
1	60	6	70
2	60	6	67.32
3	57.69231	5.769231	74
4	63.15789	6.315789	84.32
5	77.27273	7.727273	80.23

of M/M/1 is very low and the M/M/C achieved an intermediate throughput. The resource utilization of the proposed and existing models is illustrated in (Fig. 7). Table 3 shows the performance comparison analysis of resource utilization rate. The proposed model utilized more than 80% of the resources in an efficient manner. But, the existing models such as M/M/1 and M/M/C utilized only 60% and below 10% of the resources.

CONCLUSION

In this paper, the E-M/M/1/K queuing model is proposed to improve the resource allocation and scheduling efficiency in the cloud computing system. The resource allocation algorithm included a number of waiting queues to reduce the turnaround time and waiting time. The performance of the proposed E-M/M/1/K queuing model is compared with the existing models. From the analysis, it is understood that the throughput and utilization of the proposed system are very higher than the conventional models. The proposed model achieved a minimum delay, when compared to the traditional models. Hence, the resource allocation is more efficient, thereby reducing the waiting and the turnaround time of the requests.

REFERENCES

Al-Qayedi, F., Salah, K. and Zemerly, M.J. (2015). Queuing theory algorithm to find the minimal number of VMs to satisfy SLO response time. *International Conference on Information and Communication Technology Research (ICTRC)*. 64-67.

Bharathi, M., Sandeep Kumar, P. and Poornima, G. (2012). Performance factors of cloud computing data centers using M/G/m/m+ r queuing systems. *IOSR J. Eng*. 2 : 06-10.

Bheda, H.A. and Lakhani, J. (2012). QoS and performance optimization with VM provisioning approach in Cloud computing environment. *Nirma University International Conference on Engineering (NUiCONE)*. 2012 : 1-5.

Ghosh, J.K. (2012). Introduction to modeling and analysis of stochastic systems, by VG Kulkarni. International Statistical Review. 80 : 487-487.

Jin, H., Ling, X., Ibrahim, S., Cao, W., Wu, S. and Antoniu, G. (2013). Flubber: Two-level disk scheduling in virtualized environment. *Future Generation Computer Systems*. 29 : 2222-2238.

Khazaei, H., Misic, J. and Misic, V.B. (2012). Performance analysis of cloud computing centers using m/g/m/m+ r queuing systems. *IEEE Transactions on parallel and distributed systems*. 23 : 936-943.

Li, L. (2009). An optimistic differentiated service job

scheduling system for cloud computing service users and providers. *Third International Conference on Multimedia and Ubiquitous Engineering*. MUE'09. 295-299.

Mohanty, S., Pattnaik, P.K. and Mund, G.B. (2014). A Comparative Approach to Reduce the Waiting Time Using Queuing Theory in Cloud Computing Environment. *International Journal of Information and Computation Technology*. 4 : 469-474.

Mondal, B., Dasgupta, K. and Dutta, P. (2012). Load balancing in cloud computing using stochastic hill climbing-a soft computing approach. *Procedia Technology*. 4 : 783-789.

Oumellal, F., Hanini, M. and Haqiq, A. (2014). MMPP/G/m/m+ r Queuing System Model to Analytically Evaluate Cloud Computing Center Performances. *British Journal of Mathematics & Computer Science*. 4 : 1301.

Pawar, C.S. and Wagh, R.B. (2013). Priority based dynamic resource allocation in Cloud computing with modified waiting queue. *International Conference on Intelligent Systems and Signal Processing (ISSP)*. 311-316.

Sowjanya, T.S., Praveen, D., Satish, K. and Rahiman, A. (2011). The queueing theory in cloud computing to reduce the waiting time. *IJCSET*. 1 : 110-112.

Satyanarayana, A., Varma, P.S., Sundari, M.R. and Varma, P.S. (2013). Performance analysis of cloud computing under non-homogeneous conditions. *International Journal of Advanced Research in Computer Science and Software Engineering*. 3.