The International Journal Of Engineering And Science (IJES)
|| Volume || 3 || Issue || 10 || Pages || 36-43 || 2014 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

An Efficient Queuing Model for Resource Sharing in Cloud Computing

Bashir Yusuf Bichi*, Tuncay Ercan*
*Yasar University, Department of Computer Engineering
Izmir, Turkey

----------ABSTRACT----------
As cloud computing is gaining more recognition as a public utility which gives the client room to focus on his work without focusing on installation and maintenance of other important devices, as they are installed and maintained by the cloud service providers. Cloud computing is meant to be scalable, and enhance the quality of service (QoS), cost effective and also simplified user interface so that the customer can appreciate the idea behind cloud computing. In dealing with resource allocation, the client's request has to be execute through various stages, in case there are queue of requests waiting to be served in each stage. Therefore, queuing methods are required to solve this kind of situation. In this paper, we focused on mathematical formulation using queuing system technique to show how throughput and time delay of a system may varies between a single server system and a multiple server system in a cloud-computing environment.

KEYWORDS— *Queuing model, cloud computing, resource sharing, throughput, delay, utilization rate*

Date of Submission: 25 September 2014 Date of Publication: 10 October 2014

I. INTRODUCTION

The management of resources requires putting a limited access to the pool of shared resources. No matter what kind of resources you are dealing with, it also controls the status of current resource consumption. Resources in Information Communications Technologies (ICT) are the fundamental elements like hardware part of the computer systems, data communications and computer networks, operating system and software applications. Since the number of these resources is limited, it is important to restrict access to some of them. So, we can ensure an SLA (Service Level Agreement) between the customers who are requesting resources and providers who are the owners of the systems. Main resource sharing function of a distributed computer system is to assign user requests to the resources in the system such that response time, resource utilization, network throughput are optimized.

Over the decades, available ICT systems used in the development of internet and distributed systems gave computer users an opportunity to access and exploit the different resources in those systems. Recently, we have a new term in the area of computing, namely cloud computing which is a technological adaptation of distributed computing and internet. The main idea behind cloud computing is to allow customer access to computing resources through the web services in an efficient way. Cloud based network services are provided by virtual hardware which do not physically exist, and thus scale up and down according to the incoming user requests. Cloud computing provides different types of services like software as a service (SaaS), infrastructure as a service (IaaS) and platform as a service (PaaS). However, it presents a number of management challenges, because customers of these cloud services should have to integrate with the architecture defined by the cloud provider, using its specific parameters for working with cloud components. As the clients in the cloud ecosystem are increasing, it's good to find an efficient way to handle the clients' demand by maximizing the throughput and minimizing the response time for a given system.

Increase in demand of computing resource for a system that uses a single server system can result to overload of the system [1], the main benefits of having multiple servers in a system is to efficiently increase the performance of the system by reducing overloads so that a system can handle request and allocate resources to those request effectively. If single server is used in a system then the services are provided by means of batch processing while for a system with multiples servers the services are provided by using either parallel system or sequential system [2]. In this paper we will show the variation in throughput and time delay when using a single server and multiple servers.

The other part of this paper is organized as follows: In section 2, we introduce some previous studies about resource sharing and cloud computing resource management. In Section 3, we give background information about cloud computing resource sharing and emphasize why we focus on the throughput and delay of single and multiple server models. We explain the mathematics of Queuing Theory in Section 4. In section 5, we provide simulation results and analysis. Concluding remarks are given in Section 6.

II. RELATED WORK

Effort have been put in trying to find an efficient way in which cloud users can be able to use cloud resources in a way that is very quick and efficiently. Reference [1] is based on two type of systems which are single servers system M/M/1 and multiple server system M/M/n where by the waiting time by a client in the queue of each system is analyse in order to obtained the most efficient system. Satyanarayana et al. focused on a model for allocating resources to a job that arrived into the cloud using queuing model where the performance measures such as the mean number of request, throughput, utilization and mean delay in the system are analysed [3]. Authors in [4] made another effort in handling performance evaluation of cloud data center. All these works have something in common which is improving efficiency in the cloud environment. Pawar and Wagh improved resource utilisation through multiple SLA parameters (memory, network bandwidth and CPU time) and resource allocation by what is known as pre-emptive mechanism for high priority task execution [5]. Their work was based on another study made by Lugun in [6] which considers that job scheduling to be analysed with different QoS parameters required by each user and then builds a non-pre-emptive priority model for the jobs and also considers how the service providers can gain more profit from the offered resources. Bheda and Lakhani presented a dynamic provisioning technique that adapts itself to different workload changes and offers a guaranteed QoS. They modelled the behaviour and performance of applications and Cloud-based IT resources to adaptively serve user requests by using the queuing network system model and workload information for the physical infrastructure [7].

Reference [8] puts an emphasis on running multiple virtual machines (VMs)-with multiple operating systems and applications in order to address the resource allocation issues to guarantee QoS in virtualized environments like Cloud and Grid networks. The authors focused on the disk resource allocation studies rather than CPU, memory and network allocations. Cloud computing as a pool of virtualized computer resources spanned across the network is provided as a service over the Internet. User requests for web services arrive to these virtual servers and software based load balancing approaches are optimized for QoSs. The authors in [9] propose Stochastic Hill climbing algorithm for the allocation of incoming jobs to the servers or virtual machines. They compared the results with First Come First Serve (FCFS) algorithms.

III. CLOUD COMPUTING RESOURCE SHARING

Cloud computing fall in parallel and distributed computing, which is a collection of computers that are interconnected and virtualized as one computing resources and the client get access to the resources following agreement between the provider and the client otherwise known as SLA. As mentioned earlier, cloud computing offers software, platform, and infrastructure as a service respectively as shown in Fig. 1. The software as a service includes providing software such as Mail (e.g. Gmail, Yahoo mail), social network sites, Google drive, and so on, to the customers or clients. The infrastructure as a service deals with VM, storage, network, load balancer and so on as a service to the client and lastly the platform as a service deals with database like sql, oracle, web services, runtime (e.g. java) and so on as a service to the client. The clients get access to these services through various devices as shown in the figure below [10] [11].

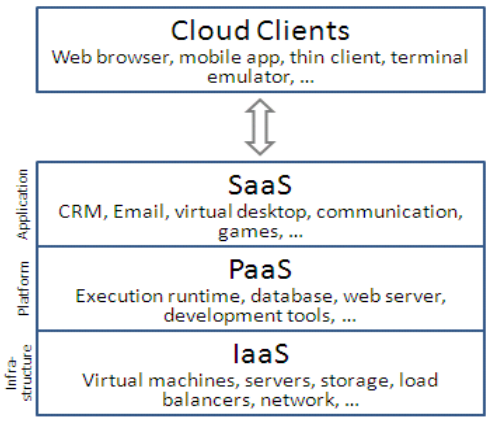

Fig. 1 Service types in Cloud Computing [http://en.wikipedia.org/wiki/Cloud_computing]

As stated in previous paragraphs [1], having multiple servers in a cloud computing environment increases the performance of the system effectively through the reduction of the mean queue length and the waiting time or delay when compared to the system with a single server. In this paper we follow the footsteps of some of the previous papers, but we will put our focus on the throughput and delay of the two models employed in our work to see which among the two models will be more efficient in terms of handling request for resource sharing. The utilization (occupancy) rate will further give us clue on how intense a system will be when dealing with the request in both models employed using the famous queuing system theory.

IV. QUEUING MODEL/KENDALL NOTATIONS

Queuing theory is a study of waiting line; it enables mathematical analysis of related process which includes arrival, waiting in the queue and being served by the server [12]. To understand the queuing system some notations where suggested by D. G. Kendall, the notations give standards to describe and classify the queuing system. A typical Kendall notation is given as A/S/C, where;

- A = arrival time for requests
- S = service time
- C = number of servers

There are other three notations that represent the number of buffers (available places in the system) as (K), calling population size as (N) and services discipline as (SD) which all are considered as infinite queue, population and FCFS (First-come-First-Served) service discipline in [13]. The arrival and service time in our work (A, S) follows Markovian process (M) whereby the arrivals follows exponential or Poisson distribution, the two notations used in our work includes, M/M/1 and M/M/c.

A typical queuing system consists of input, which are the requests seeking to be process, arrivals i.e. units that entered the system seeking resources, queue that houses the request seeking resources, service facilities that served the request and departure i.e. the units of request that have complete service and leave the system [13].

The queuing discipline is an important characteristics of queuing system where request are selected for service when queue is formed, the discipline can be; FCFS, random service selection (RSS) or priority system, where some requests are given higher priority. Number of service channels is another important characteristic of queuing system where the system can be either single or multiple servers [12].

Since the cloud computing environment can have either a single server system or multiple server system, this gives us the ability to use the queuing theory to mathematically show some relationships in terms of efficiency between the two systems. Below are some notations we used in our analysis;

P_o = Probability the system is empty
R_s = expected number of request in the service facility
R_q = expected number of request in the queue
R= expected number of unit in the system
T_s = expected time in the service facility
T_q = expected time in the system
λ = arrival rate of request
μ = number of request completion
Th= Throughput

The queue is said to be stable if the service rate μ is greater than the mean arrival rate λ i.e. "$\mu>\lambda$", so that the system will not keep growing forever, and whenever the system is busy, it will eventually reach a state where the system will be idle [13].

A. Single Server System (M/M/1)

For an M/M/1 server system shown in Fig. 2, it means it is an exponential distribution that consist of

- Exponentially distributed inter-arrival times.
- Exponentially distributed service time.
- Infinite population of potential request.

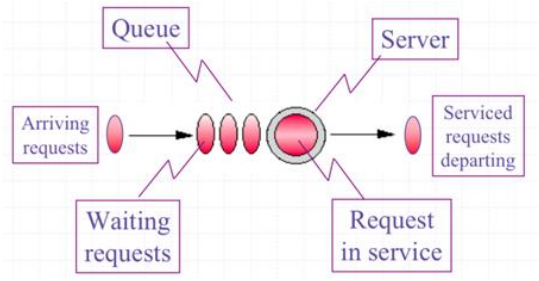

Fig. 2 M/M/1 Service types in Cloud Computing

ρ<1 is needed to assure the system is in equilibrium n= number of units (request) in the system (n≥ 0), for a system with one server and infinite request, using a derived reduced equations and the notation in (1) , the probability of n units in the system

Traffic intensity for a single server system is given as;

$\rho = \lambda/\mu$ (1)

$P_o = (1-\rho)$ (2)

$P_n = \rho^n(1-\rho)\, n \geq 0$ (3)

Expected request in the system facility is given by;

$R_s = \rho$ (4)

Expected request in the queue

$R_q = \rho^2/(1-\rho)$ (5)

Expected number of request in the system is the summation of (4) and (5) which gives;

$R = \rho/(1-\rho)$ (6)

The expected time in the service (T_s), is obtained by dividing (4) by λ

$T_s = 1/\mu$ (7)

The expected time (delay) in the queue (T_q) for a single server system is given by;

$T_q = R_q/\lambda = \rho/[\mu(1-\rho)]$ (8)

The throughput for M/M/1 systems is given by

$\therefore$ Th = λ (9)

B. Multiple Server System (M/M/c)

In multiple server system shown in Fig. 3, the request join a single queue, where by the request will be served by single server in the system that is idle. The servers are identical and any request are be served by any server.

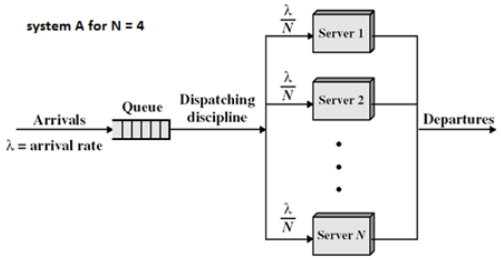

Fig. 3 M/M/c Service types in Cloud Computing
[http://math.stackexchange.com]

For a system with many servers and infinity request the probabilities are given as;

Utilization of single server is given by

$\rho = \lambda/n\mu$ (10)

Utilization of the system is given by $r = \lambda/\mu$

p_o = probability when there is no request in the system

$$p_o = 1/\{\sum_{n=0}^{k-1} r^n/n! + r^k/((k-1)!(k-\rho))\} \quad (11)$$

Where n is the number of request in the system (n≥0) and k is the number of service facilities (servers)
The probability of n request in the system is given by

$$p_n = \begin{cases} r^n/n! * p_o = (0, k-1) \\ r^n/(k!\,k^{n-k}) * p_o \; n \geq k \end{cases} \quad (12)$$

Expected request in the queue (R_q)

$$R_q = p_0 \; r^k \rho/k!\,(1-\rho)^2 \quad (13)$$

The expected time (delay) in the queue for the multiple server system is given by;

$$T_q = R_q/\lambda = (p_0 * r^k/k!\,(k\mu)\,(1-\rho)^2 \quad (14)$$

To obtain the throughput of the system we first find the throughput of a completed service in a given time which is obtained as;

$$Th = k\rho\mu$$
$$Th = k\lambda \quad (15)$$

It is obvious that a multiple server system will be more efficient in terms of performance; however it is important to put these facts into analysis, as it will help researchers to easily visualize the differences when employing such systems. Even in multiple server systems, those with more servers will perform better that those with less servers. Virtualization of these servers will make the systems more efficient when allocating resources to different user request.

V. SIMULATION AND ANALYSIS OF RESULTS

In our simulation we let "n=10", and for M/M/c, the number of service facility in the system "k=5". The table below shows the result of our simulations for both single server system and multiple server system. Our aim is to compare the throughput of each system to have a clear view of how using multiple server system is more efficient and time saving than using the single server system. The throughput is the number of completed request per unit time while the delay is the time taken by a request in the queue until it's been executed. The values we used in our simulation are randomly picked so as to conform with the pattern in which request for resources enters a given system.

TABLE I
ARRIVAL RATE, NUMBER OF REQUESTS, THROUGHPUT, DELAY

λ	μ	Throughput(Th)		Delay (T_q)	
		M/M/1	M/M/c	M/M/1	M/M/c
15	25	15	75	0.0600	2.68e-05
30	50	30	150	0.0300	1.34e-05
45	78	45	225	0.0161	6.67e-06
60	95	60	325	0.02280	1.12e-05
85	110	85	425	0.0309	1.47e-05

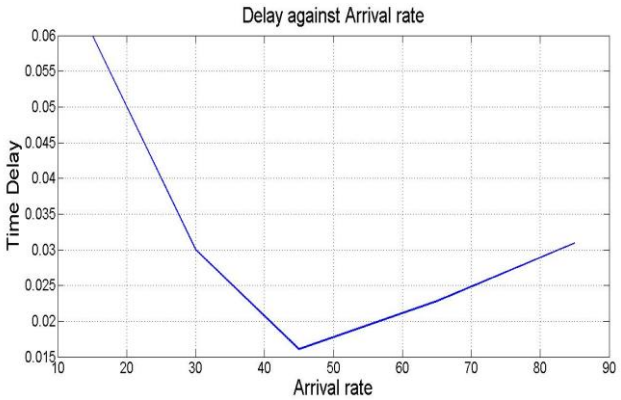

Fig. 4 Delay against the arrival time for M/M/1 system

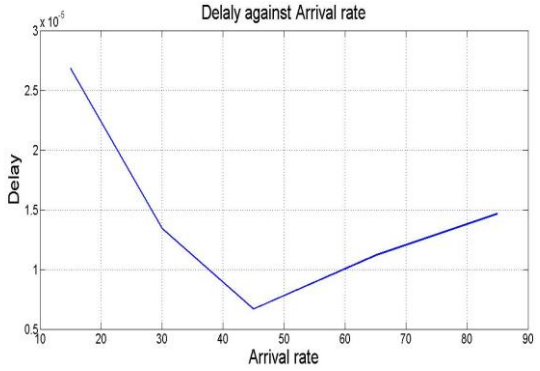

Fig. 5 Delay against the arrival rate for M/M/c system

From the graphs (Fig.5 and Fig.6), we can see the waiting time or delay in the M/M/c crashed down to values which are less when compared to that of M/M/1 and we can see the M/M/1 graph keeps increasing. The M/M/c graph started to increase after it crashed down, however the rate at which it increases is very slow.

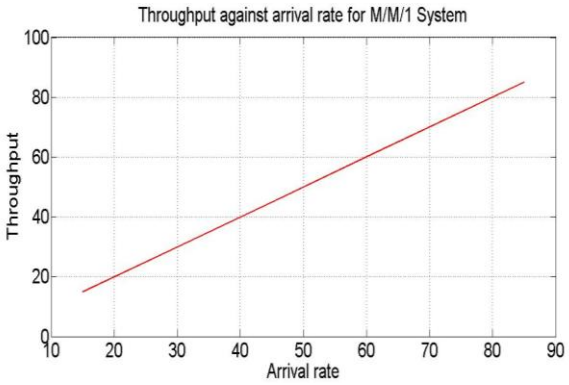

Fig. 6 Graph of throughput against the arrival rate for M/M/1 system

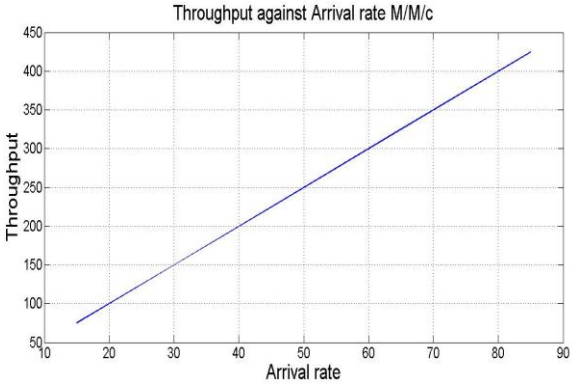

Fig. 7 Graph of Throughput against the arrival rate for M/M/c

Considering Fig.6 and Fig.7, each shows the throughput of our respective models and we can visibly see the differences in terms of the request that can be served in each model, and from these figures we conclude that M/M/c is more efficient for the resource allocation.

TABLE II
ARRIVAL RATE, NUMBER OF REQUESTS, UTILIZATION RATE (%)

Λ	μ	M/M/1 (ρ %)	M/M/c (ρ %)
15	25	60	6
30	50	60	6
45	78	57.69231	5.769231
60	95	63.15789	6.315789
85	110	77.27273	7.727273

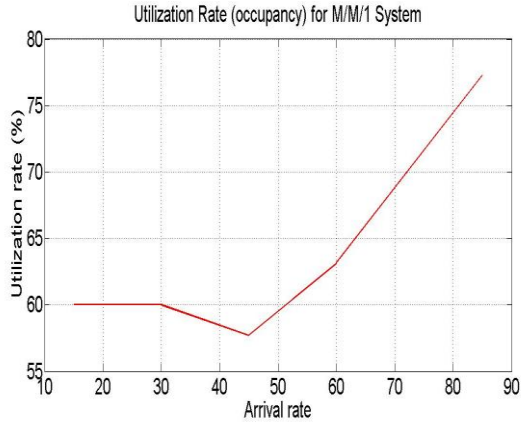

Fig. 8 Utilization rate (%) against the arrival rate for M/M/1 system

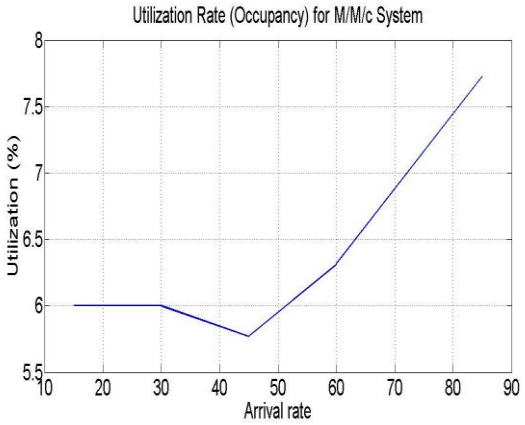

Fig. 9 Utilization rate (%) against the arrival rate for M/M/c system

Fig.8 and Fig.9 shows the level of utilization (based on the values in Table 2) of a given system when handling the same request and depicts that Fig.9 is less occupied than Fig.8.

VI. CONCLUSION

In conclusion we can see clearly that the throughput of system with multiple serving units is above the system that has a single serving unit. And also when we consider the waiting time for a certain request before it can be executed on a single server is greater than when a multiple server system is employed. The utilization rate (occupancy) for the multiple server system is much lower than that of single server system, there for in our opinion, to have an efficient and reliable system in handling request for resources in cloud computing environment it is necessary to have a multiple server system, even for a multiple server, virtualization of each server will help more in increasing the efficiency in handling the systems activities.

REFERENCES

[1] T. Sai Sowjanya, D.Praveen, K.Satish, A.Rahiman, *The Queueing Theory in Cloud Computing to Reduce the Waiting Time*, April 2011.
[2] S. Mohanty, P. K. Pattnaik and G. B. Mund, *A Comparative Approach to Reduce the Waiting Time Using Queuing Theory in Cloud Computing Environment*, 2014.
[3] A. Satyanarayana, P. Suresh Varma, M.V.Rama Sundari, P Sarada Varma, *Performance Analysis of Cloud Computing under Non Homogeneous Conditions*, May 2013
[4] M. Bharathi, P. Sandeep Kumar, G. V. Poornima, *Performance factors of cloud computing data centers using M/G/m/m+r queuing systems*, Sept 2012.
[5] C. S. Pawar, R. B. Wagh. *Priority Based Dynamic Resource Allocation in Cloud Computing with modified waiting Queue*, March 2013.
[6] L. Lugun. *An Optimistic Differentiated Service job Scheduling System for Cloud Computing service users and providers*, 2009.
[7] H. A. Bheda, J. Lakhani, *QoS and Performance Optimization with VM Provisioning Approach in Cloud Computing Environment*, 3rd Nirma University International Conference on Engineering , Ahmedabad, INDIA, 2012.
[8] H. Jin, X. Ling, S. Ibrahim, W. Z. Cao, S. Wu, and G. Antoniu, *Flubber: Two-level disk scheduling in virtualized environment*, Future Generation Computer Systems-the International Journal of Grid Computing and E-science, vol. 29, pp. 2222-2238, Oct 2013.
[9] B. Mondal, K. Dasgupta, and P. Dutta, *Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach*, 2nd International Conference on Computer, Communication, Control and Information Technology (C3it-2012), vol. 4, pp. 783-789, 2012.

[10] B. Sosinsky, *Cloud Computing Bible*, Wiley Publishing. Inc 2011
[11] J. Kris, *Cloud Computing: SaaS, PaaS, IaaS, Virtualization, Business Models, Mobile, Security and More*, 2012 Ed.
[12] K. A. Williams, *Queuing note, Department of computer science*, North Calorina A & T State University, 2012.
[13] N. T. Thomopoulos, *Fundamentals of Queuing Systems*, Stuart School of Business Illinois Institute of Technology Chicago, IL 60661 USA. 2012 Ed.
[14] S. Mohanty, P. K. Pattnaik and G. B. Mund, A Comparative Approach to Reduce the Waiting Time Using Queuing Theory in Cloud Computing Environment, 2014.
[15] M. Zukerman , Introduction to Queuing Theory and Stochastic Teletraffic Models, 2014, p.88.