Asian Journal of Engineering and Applied Technology
ISSN: 2249-068X (P) Vol. 4 No. 1, 2015, pp.8-11

DOI: https://doi.org/10.51983/ajeat-2015.4.1.751

Performance Measures of Queuing Models Using Cloud Computing

K.Ruth Evangelin and V.Vidhya
Saveetha University, Chennai, Tamil Nadu, India
E-mail: ruthkavati@yahoo.co.in, vidya_zero@yahoo.co.in
(Received 17 January 2015; Accepted 28 February 2015; Available online 8 March 2015)

***Abstract* - Cloud computing is an emerging technology to provide cost effective and to deliver the business application services in an adaptable way. In cloud computing, multi resources such as processing, bandwidth and storage, need to be allocated simultaneously to multiple users. It is becoming a development trend. The process of entering into the cloud is generally in the form of queue, so that each user need to wait until the current user is being served. In the system, each Cloud Computing User (CCU) requests Cloud computing Service Provider (CCSP) to use the resources, if CCU finds that the server is busy, CCU's needs to enter into the waiting line until CCSP completes its service to the previous CCU . So this may lead to bottleneck in the network.. So to solve this problem, it is the work of CCSP's to provide service to users with less waiting time , otherwise there is a chance that the user might be leaving from queue. CCSP's can use multiple servers for reducing queue length and waiting This paper proposes a (M/M/C):(∞/FIFO) Queuing model which is applied at multiple servers in order to reduce waiting time, queue length, the network performance and QOS effectively in cloud computing environment.**
***Keywords:* Cloud Computing, , Queue length, Waiting time, Queuing Model, QOS.**

I.INTRODUCTION

Cloud computing [13] [14] [15] often referred as a cloud has been an emerging technology for provisioning computing resource and providing infrastructure of web applications [1] in recent years. Meanwhile, leading IT companies have established public commercial clouds as a new kind of investment. For example, Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers [2]. Google App Engine enables enterprises to build and host web applications on the same systems that power Google applications. App Engine offers fast development and deployment; simple administration, with no need to worry about hardware, patches or backups; and effortless scalability [3]. IBM also provides cloud options. Whether you choose to build private clouds, use the IBM cloud, or create a hybrid cloud that includes both, these secure workload solutions provide superior service management and new choices for deployment [4]. We even can establish a private cloud with Ubuntu Enterprise Cloud to offer immediacy and elasticity in the infrastructure of web applications [1]. In summary, both of the numbers of cloud applications and providers have kept gradually increasing for a couple of years. As a result, computing resource scheduling and performance managing have been ones of the most important aspects of clouding computing. Among the top 10 obstacles of cloud which the report [1] proposes, the obstacle 8, Scaling Quickly, is our focus. When a number of web applications are deployed into a cloud environment, dynamical allocating the computing resource to web applications on demand has a positive effect not only on the performance of web applications, but also on the energy saving. The solution to eliminate this obstacle is to automatically scale quickly up and down in response to load in order to save money for web applications providers by optimizing the requesting of computing resource, but without violating service level agreements [1]. Meanwhile, the cloud providers also can save money by optimizing the allocation of computing resource and saving energy, since the cloud providers needn't to provide excessive active computing resources. To achieve the aim of dynamic scaling, we need proper tools and models to diagnose the runtime requirements of web applications. Since there is not any standard model has been widely accepted by industry yet, scaling up and down is an open issue for researchers. The cloud providers, such as Amazon, IBM, and Google have their own mechanisms which are commercial ones and inherited from their existing proprietary technology. The researchers from universities and institutes also have proposed some models and methods. For example, in [6], the author introduces many outcomes on predicting system performance based on machine learning obtained in RAD lab of University of California at Berkeley. The existing solutions to scaling up and down are designed via various techniques, such as statistical methods, machine learning, and queueing theory. Aware of the advantages and disadvantages of these solutions, we propose a queueing-based model for performance management on cloud. In this model, the web applications are modeled as queues and the virtual machines are modeled as service centers. We apply the queueing theory onto how to dynamically create and remove virtual machines in order to implement scaling up and down. The remainder of the paper is structured as follows.

II. CLOUD COMPUTING

Resource allocation in a cloud computing environment can be modeled as allocating the required amount of multiple types of resource simultaneously from a common resource pool for a certain period of time for each request

There are three kinds of cloud services model, namely, Software as a Service (SaaS), Platform as a Service (PaaS) and Cloud Infrastructure as a Service (laaS) [4].

Software-as-a-Service **(SaaS)** is a software distribution model in which applications are accessible through a single interface, like a web browser over the Internet. Users do not have to consider the underlying cloud infrastructure including servers, storage, platforms, etc.

Platform-as-a-Service **(PaaS)** provides a high level of integrated applications that control of distributed applications and their hosting environment configurations. In general, developers accept all instructions on the type of software that can be written to change built-in scalability.

Infrastructure-as-a-Service **(IaaS)** provides users with computation processing, storage, networks and computing resources. IaaS users can implement an arbitrary application which is able to grow up and down dynamically. Also, IaaS sends programs and related data, while the cloud provider does the computation processing and returns the result [4]. The main aim of the cloud service providers is to administer the system , monitor traffic flow to ensure maximum usage of the resources in minimum waiting time. When Multiple users enters into the Cloud for sharing the resources or data at a time, when server is busy, the CCU forms queue or may enter into reneging state which degrades the network performance . Therefore, in this paper, the (M/M/C): (∞/FIFO) Queuing model is applied at multiple servers to reduce mean waiting time which results in decrease in queue length and improving QoS in cloud computing environment.

III. RELATED WORK

Queuing theory has been applied to develop analytical methods for evaluating Cloud service performance. Xiong and Perros[8] modeled a Cloud computing system as an open queue network consisting of two tandem servers with finite buffer space, where both interracial and service times are assumed to have exponential distribution. T.Sai Sowjanya et al.,[7] have shown M/M/S model for two servers which increases the performance over using one server by reducing the queue length and waiting time.. In order to study resource allocation for meeting performance requirements of clients with different priority levels. modeled a Cloud centre as an M/M/C/C queuing system, which has C = 1, C =2, C = 3 servers with no buffer space and Markov processes for both arrival and departure. Yang et al [9] developed a queuing model for Cloud data centres . model both arrival and service times are assumed to be exponentially distributed and service response time is broken into three independent parts: waiting, service, execution periods. In [10] they employ the queuing model to investigate resource allocation problems I service case and multiple-class service case. Furthermore, they optimize the resource allocation to minimize the mean response time.

IV. QUEUING THEORY

Queuing Theory [11] is a collection of mathematical models of various systems of queues. It is widely used to analyze the arrival rate and service time. Formation of queues arises when demand for a service exceeds the limited capacity of the system. To analyse the arrival rate & service rate and to deliver the packet to the destination a Queuing model[12] which is a Mathematical, Probalistic and Markovian model is applied at routing stages. Queuing system is characterized by the components namely:

1. Arrival rate: describes the way the population arrives either static or dynamically..
2. Service rate: describes how many customers can be served when the service is available .
3. No of service channels: Service channel contains single ormultiple. Customers enter one of the parallel service channels and is served by the customer d) Queue discipline: describes the manner in which customers choose for the service like First in Firstout(FIFO), Last in First Out(FIFO).

Customer behaviour generally be in four states. They are:

a. Balking : when the Queue is too long customer decides to enter or not in the queue .
b. Reneging: The customer leaves from the queue if he has impatience to wait.
c. Jockeying :when there are two or more parallel queues the customer move from one queue to other.

KENDELL'S NOTATION

A Queuing system can be described based on their notations:
A/B/C/D/E/F where
A : probability distribution of the arrival rate
B : service time distribution
C : number of servers
D : system capacity
E: population size
F : service discipline
Key notations:
λ: Mean arrival rate
μ:Mean Service rate
$\rho = \lambda / \mu$: server utilization
Steady state distribution: the system is in steady state when the behaviour of the system becomes independent of time.

V. (M/M/C):(∞/FIFO) QUEUING MODEL

It is assumed that, if CCU arrives at an average rate λ and server has service mean rate μ and finds the server in busy state then CCU has to wait till the server completes its job or CCU may enter into Balking or Reneging state .This

results increasing in waiting time and queue length . Therefore in order to overcome this problem (M/M/C): (∞/FIFO). Queuing model is applied when there are multiple servers , C and each server has an independent identical exponential service time distribution n. The arrival process assumed to be poisson .and ∞ indicates CCU.

The mean service = rate will be Cμ . The steady state probabilities are:

Measures of effectiveness :

The probability of zero customers in the system is $P_0 = \frac{1}{\frac{(\frac{\lambda}{\mu})^n}{n!}}$

The probability of n customers in the system is $P_n = \frac{(\frac{\lambda}{\mu})^n e^{-(\frac{\lambda}{\mu})}}{n!}$

Average number of customers in the system is $[E_s] = (\frac{\lambda}{\mu-\lambda})$

Average waiting time at the system is $[C_s] = (\frac{1}{\mu-\lambda})$

if there are c = 2 servers or c = 3 servers, the generalized formula is given as

$[E_q] = \frac{\rho^{c+1}}{(c-1)!(c-\rho)^2} P_0$

$[E_s] = [E_q] + \rho$

$[C_q] = \frac{[Eq]}{\lambda}$

$[C_s] = [Cq] + (\frac{1}{\mu})$

Where ,

$[E_q]$: Expected. number of customers in queue.

$[E_s]$: Expected number of customers in the system.

[Cq] : Expected waiting time per customers in the queue

$[C_s]$: Expected waiting time per customers in the system

expected waiting time per customers in the system

If the no. of servers is c = 1, then it turns out to be a single server.(i.e) an M/M/1 queuing model

Where we can calculate the performance measures such as

VI. NUMERICAL EXAMPLE

Consider two Queueing models (M/M/1): (∞/FIFO), (M/M/C): (∞/FIFO) (with arrival rate $\lambda = 5$ and service rate $\mu = 7$). We have found out the performance measures such as the waiting time at the system $[C_s]$ and the number of customers at the system $[E_s]$ numerically.

The tabular column is listed below.

Model	Number of servers	$[E_s]$	$[C_s]$
M/M/1	C = 1	2.5	0.5
M/M/C	C = 2	0.934	0.187
	C = 3	0.764	0.15

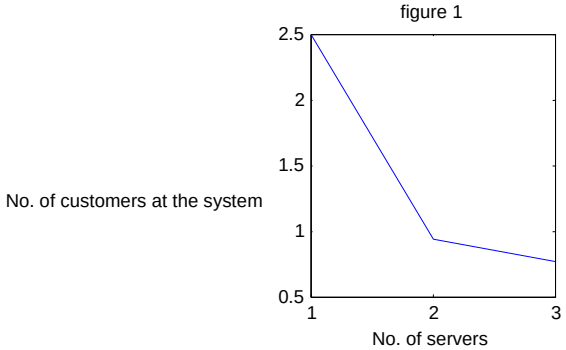
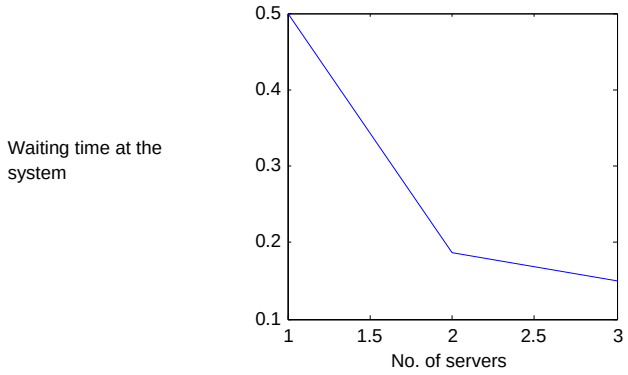


figure 2



VII. CONCLUSION

From the numerical values, we are able to analyse that when we have more cloud computing servers, our waiting time will be definitely reduced. We are able to conclude that the dynamic behaviour of the servers gives a good enhancement. With good selection of the number of servers, we can reduce the queue length and reduce the waiting time.

REFERENCES

[1] Michael Armbrust, Armando Fox, et.al, Above the Clouds: A Berkeley View of Cloud Computing, http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf

[2] Amazon, Amazon Elastic Compute Cloud (Amazon EC2), available at:http://aws.amazon.com/ec2/, 2010

[3] Google, *Google App Engine*, available at :http://code.google.com /intl/en/appengine/, 2010

[4] IBM, IBM Smart Business Cloud Computing, available at:http://www.ibm.com/ibm/cloud/, 2010

[5] Ubuntu, *Private cloud: Ubuntu Enterprise Cloud*, available t:http:// www.ubuntu.com/cloud/private, 2010

[6] Archana Sulochana Ganapath, Predicting and Optimizing SystemUtilization and Performance via statistical Machine Learning, Technical Report No. UCB/EECS-2009-181, available at:http:// www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009- 181.html, December 17, 2009

[7] T.Sai Sowjanya et al, "The Queueing Theory in loud Computing to Reduce the Waiting time", International Journal of Computer Science and Engineering Technology, April 2011, Vol 1, Issue 3, pp 110-112

[8] K.Xiong and H. Perros, "Service performance and analysis in cloud computing", in Proceedings of the 5th World Congress on Services, Los Angeles, California, USA, July 2009, pp 693-700.

[9] Yang, F.Tan, Y.S. Dai and S. guo, "Performance evalution of cloud service considering fault recovery" in Proceedings of the I international Conference on Cloud Computing, Beijing, China,December 2009, pp 571-576

[10] X.M. Nan, Y.F. He, L.Guan, " Optimal Resource Allocation for Multimedia Cloud Based on Queueing Model", Multimedia Signal Processing, 2011 IEEE 13th International Workshop on, 2011, pp 1-6.

[11] N. Ani Brown Mary and K.Saravanan, "' Performance factors of Cloud Computing Data Centers using {(M/G/1) : (_/gdmodel)} Queuing systems" International journal of grid computing & Applications vol 4, no.1, march 2013.

[12] C. Knessl, B.Matkowsky, Z.Schuss and C.Tier, "Asymptotic behavior of a state dependent M/G/1 ueueing system", SIAM journal of Applied Math.46(1986) pp 483-505

[13] Suneeta Mohanty, Prasant Kumar Pattnaik and Ganga Bishnu Mund "A Comparative Approach to Reduce the Waiting Time Using Queuing Theory in Cloud Computing Environment" International Journal of Information and Computation Technology. ISSN 0974-2239 Volume 4, Number 5 (2014), pp. 469-474.

[14] Kusaka, T, Okuda, T, Ideguchi, T, Xuejun,Tian " Queuing theoretic approach to server allocation problem in timedelay cloud computing systems " Teletraffic congress (ITC), 2011, 23rd International publications , 2011 pp:310-311.

[15] Souvik Pal and P. K. Pattnaik, "Efficient architectural Framework of Cloud Computing", in "International Journal of Cloud Computing and Services Science (IJCLOSER)", Vol.1, No.2,June 2012, pp. 66-73.