International Journal of Information and Computation Technology.
ISSN 0974-2239 Volume 4, Number 5 (2014), pp. 469-474

http://www. irphouse.com /ijict.htm

A Comparative Approach to Reduce the Waiting Time Using Queuing Theory in Cloud Computing Environment

Suneeta Mohanty, Prasant Kumar Pattnaik and Ganga Bishnu Mund

School of Computer Engineering, KIIT University, INDIA.

Abstract

Cloud computing is a model to provide different type of services like SaaS, PaaS and IaaS over the internet. Cloud is a collection of virtualized and interconnected computers dynamically presented as one or more unified computing resources based on Service Level Agreement. Job scheduling is the very important issue in Cloud Computing Environment because cloud resources are provisioned according to the need of the users. So the users can get the cloud resources efficiently within short period of time. In a Cloud Environment, service required by the cloud user, and the service times of service provider are random variables. If the servers are doing some task at the time user request came or there are limited numbers of servers then users need to wait.In this point queuing theory is applied to reduce the waiting time to improve the quality of the cloud service provider. In this paper we have done different numerical analysis and comparison using M/M/C-Model & Erlang-C-Model to reduce waiting time by varying the number of servers.

Keywords: Cloud Computing, QoS, queuing theory, average waiting time.

Introduction

Cloud Computing is a model that delivers resources as services over the Internet on demand based on service-level agreement [1][2][3].The main aim of the cloud service providers is to ensure maximum usage of the resources with minimal waiting time[4]. Scheduling criteria would be in such a way that the waiting time can be minimized and depending upon the servers. Here we can use multiple numbers of servers which can be efficiently shared among the users [5].To increase the system performance,

maximize the usage of resources and minimize the waiting time multiple servers are used in Cloud Computing . Efficient queuing theory is highly required to provide the Quality of Service (QoS) to the cloud users. Therefore, we have used the concept of Queuing theory for providing QoS. In this paper, we have presented a comparative study between M/M/C Model and Erlang-C Model to reduce the waiting time by increasing number of servers.

The paper is organized as follows:

In the section 2, we have discussed about the Cloud Computing, Section 3 describes the Queuing theory, and Section 4 provides the study of numerical analysis. Lastly, section 5 concludes the work.

Brief About Cloud Computing

Cloud computing [8] is used for providing different resources as service on demand to cloud user [7][6] with less cost and time. Quality of Service (QoS) is referred to as the resource reservation control mechanisms in place to guarantee a certain level of performance and availability of a service.ServiceLevelAgreements provide a way for mutual agreement over QoS between an End-User and Service Provider .

Queueing Theory

3.1. Need for Queuing Theory in Cloud Computing Environment

In concern to Cloud services, there are numbers of cloud users who are very keen to getting the services from the Cloud Service Providers. So they made the requests to the provider and the job scheduler of the cloud provider handles all the requests. Efficient usage of servers may be capable of maximized sharing of the system and computational resources, minimizing the cost complexity, and reducing the waiting time. In a Cloud Environment, service required by the cloud user, and the service times of service provider are random variables. If the servers are doing some task at the time user request came or there are limited numbers of servers then users need to wait. In this point queuing theory is applied to reduce the waiting time. Cloud customers are arriving to the cloud provider, and due to lack of servers or limited capacity of resources, they have to wait in line (in a queue). In this paper, we have used different number of servers for reducing waiting time. Queuing theory is the learning of the waiting line phenomenon. Waiting lines or queues arise when the number of requests exceeds the ability of that service. The basics of queuing process are described as below [9]:

3.2 The arrival Process

- **The source size**

The source size denotes the total number of arriving populations. If the population is infinite, we can assume that the number of present population won't affect the arrival process. And if the population is finite, the number of population will affect the arrival process.

- **Pattern of arriving populations**

Cloud customers may arrive for getting service at a queuing system either in regular basis or randomly.

- **Behavior of arrival**

Arriving customers may behave in a different way when the all the servers are busy or waiting room is full due to limited waiting queue.

3.3 The Service process

- **Distribution pattern of Service Time**

Service time is the time period from the start of service to the end of service. Erlang and exponential distributions are the most appropriate one.

- **Capacity of the system**

This refers to the system capacity that means the ability to accommodate the maximum number of users in the system.

- **The number of servers:**

If the number of server is one, then service is provided using batch processing and if the there are multiple servers, service is provided using either parallel system or sequential system.

3.4 The Queuing discipline

Queuing discipline indicates the service discipline in which the customers would stand and the sequence in which they getting served.

- *FCFS (First Come First Serve):* This is the most popular pattern and customers are served on the basis of the order they arrive.
- *LCFS (Last Come First Serve):* The customers are served on the basis of opposite order they arrive
- *SIRO (Service in Random Order):* Service is provided in a random fashion.
- *GD (General queue Discipline):* Service is provided in general queue system.
- *Priority discipline:* Service is provided based on their priorities.

3.5 The waiting room

It is of two types:

- *Finite:* When waiting room is full, a newly arrived customer can't find a place to accommodate, he leaves never to return.
- *Infinite:* When the customer-arrival is relatively extremely large, then it is assumed that waiting room is infinite.

4. Numerical Analysis

In the M/M/C model [10][5],

n = number of servers
λ = Arrival rate
μ = Service rate
The traffic intensity for n servers is $\rho_s = \rho/S$ 1
Mean staying time is: $W = Wq + 1/\mu$ 2
Mean waiting time is $Wq = Lq / \lambda$ 3
Mean queue length is $Lq = P_0 \rho^s \rho_s / s!(1- \rho_s)^2$ 4
In t In Erlang- C queuing model,we assume a **FIFO** queuing discipline. ***x*** = Traffic intensity , ***C*** = Probability of delay & n= number of servers.
AWA = **A**verage **W**ait for **A**ll customers
The Erlang C function ***C***(***n***,***x***) is defined as
$C(n,x) = n*B(n,x) / (n - x*(1-B(n,x)))$ 5
Where ***B*** (***n***,***x***) is the Erlang B function. Then
$\lambda/\mu < n$ and $x = \lambda/\mu$ 6
C = ***C***(***n***,***x***) 7
$AWA = C / (\mu*(n-x))$ 8

Considering the following set of λ(20,40,60) and μ(70,120,122) we have the following results shown in the tables.

Table 1: M/M/C-Model

λ	Wq(n=1)	Wq(n=2)	Wq(n=3)
20	0.025	0.0003	0.0001
60	0.08	0.001	0.004
120	0.49	0.1	0.009

Table 2: Erlang-C-Model.

λ	AWA(n=1)	AWA(n=2)	AWA(n=3)
20	0.024	0.001	0.0001
60	0.079	0.001	0.0003
120	0.25	0.001	0.0003

5. Conclusion

In Cloud Computing Environment, different services are provided to the customer as per their request. Since the arrival time of request and service time to serve are random variables, it is important to reduce the waiting time to improve QoS of Cloud Computing Environment. Hence in this paper through above numerical analysis we have shown that how M/M/C model and Erlang-C models are used to reduce the waiting time by increasing number of servers. Also we found that Erlang-C model is giving better result to reduce waiting time by increasing number of servers than M/M/C model.

References

[1] M. Armburst et al., "Above the Clouds: A Berkeley View of Cloud Computing", Tech. report, Univ. of California, Berkeley, 2009.

[2] RajkumarBuyyaa, Chee Shin Yeoa, , SrikumarVenugopala, James Broberga, and IvonaBrandicc, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility", Future Generation Computer Systems, Volume 25, Issue 6, June 2009, Pages 599-616.

[3] V. Sarathy, P. Narayan, RaoMikkilineni, "Next generation cloud computing architecture -enabling real-time dynamism for shared distributed physical infrastructure", 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10), Larissa, Greece, 28-30 June 2010, pp. 48-53.

[4] Luquan Li, "An Optimistic Differentiated Service Job Scheduling System for Cloud computing Service Users and Providers", Third International Conference on multimedia and Ubiquitous Engineering, 2009, pp. 295~299.

[5] T. sai Sowjanya et al, "The Queuing Theory in cloud Computing to Reduce the Waiting Time", International Journal of Computer Science and Engineering Technology, April 2011, Vol. 1, Issue 3, pp. 110~112.

[6] V. Sarathy et al, "Next generation cloud computing architecture -enabling real-time dynamism for shared distributed physical infrastructure", 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE'10), Larissa, Greece, 28-30 June 2010, pp. 48-53.

[7] P. Mell et al, "NIST definition of cloud computing", vol. 15, October 2009.

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

[9] Lotfi Tadj, "Waiting in line", IEEE POTENTIALS, January1996, pp. 11~13.

[10] D. G. kendall, "Some Problems in Theory of Queues", J. Roy. Stat. Soc., Series B, Vol. 13, No. 2, 1951.

[11] Hall, J.A.& Liedtka, L.L. "The Sarbanes-Oxley Act: implications for Large-scale IT outsourcing", Communications of the ACM, 2007, pp. 95-100.