AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Modeling the Cloud e-Marketplaces for Cost Minimization Using Queuing Model

[1]A.O. Akingbesote, [1]M.O. Adigun, [1]J. Oladosu, [1]E. Jembere, [2]I. Kaseeram

[1]*Department of Computer Science, University of ZuluL, South Africa*
[2]*Department of Economics, University of ZuluLand Kwlangenzua, South Africa*

ARTICLE INFO

Article history:
Received 20 November 2013
Received in revised form 24 January 2014
Accepted 29 January 2014
Available online 25 February 2014

Keywords:
e-Marketplaces; Optimal Service Level; Server Machines; Cost Model; Queue model

ABSTRACT

The cloud e-Marketplaces are becoming perfect competitive markets. In these markets, providers must decide what level of service to offer. A low level of service may be inexpensive, at least in the short run, but may incur high costs of consumer's dissatisfaction, such as loss of future business and actual processing costs of complaints. A high level of service will cost more to e-cloud provider and will result in lower dissatisfaction costs. The optimal level of service to be provided by a cloud provider management that will minimize cost at the same time satisfy the consumers in terms of waiting time is a challenge. The goal of this paper is to determine the optimal service level that will minimize costs as well as minimize the consumer's waiting time. To achieve this, we used the queuing system to get our performance measure and we then designed our cost model using the performance result. Our optimal result was achieved at the point where the total expected cost has the minimum value.

To Cite This Article: A.O. Akingbesote, M.O. Adigun, J. Oladosu, E. Jembere, I. Kaseeram., Modeling the Cloud e-Marketplaces for Cost Minimization Using Queuing Model. ***Aust. J. Basic & Appl. Sci.,*** *8(4): 59-67, 2014*

INTRODUCTION

The cloud e-marketplaces can be referred to as the virtual environment for buying and selling of services. It is virtual because it is not physical like the traditional markets but share the same simple idea of exchanging goods for services. We assume goods are the costs, in terms of waiting time to access server paid by the consumers or clients. These competitive markets consist of two major participants, the cloud e-market consumers and the cloud e-market provider.

In a typical cloud market, response time is of interest to every consumer and is also a key source of competitive advantage for any cloud e-market provider (C.C. Kim, L. Smith, H. Thorne and R.W. Hilton, 2008). For example, a cloud provider must meet the service level agreement (SLA) and if possible, reduce the service response time to attract more consumers. In addition, profit maximization is important to e-cloud providers. One methods of balancing this is to look at various sources of costs and see how one can minimize cost to maximize profit.

The work of (W. Deng, F. Liu, H. Jin, and C. Wu, 2013) mentioned three problems being faced by cloud e-market providers. These are Skyrocketing power consumption and electricity bills, serious environmental impact, and unexpected power outages. With the current server consolidation, the report in ("IDC," 2013) showed that about $45 billion was spent on server management and administrative costs in 2012. Therefore, effective management of these Server Machines (SMs) in terms of cost minimization is imperative.

As a result of changing nature of cloud e-market, environments, and due to diversity of users' requests, and time dependency of load, providing agreed Quality of Service (QoS) while avoiding over-provisioning of server machine is a difficult task (K. Xiong and H. Perros, 2009; G. Rahul, S.T Kishor, K.N. Vijay and S.K. Dong, 2010; K. Hamzeh, M. Jelena and B.M. Vojislav, 2011). One major challenge in this completive market and which is our goal in this paper is on how to determine the optimal level of service (Server machines) to be provided by cloud e-market provider that will minimize cost at the same time satisfies the consumer waiting time.

To achieve our goal, we modeled the cloud e-marketplace as networks of queues with M/M/1/ k and M/M/c/k where our arrival and service times are markovian while c and k represent the number of server machines and the buffer capacity respectively. The result of this queuing model was used to formulate our cost model in terms of expected cost. The justification for using the queuing model was based on the work of (C. Hao-peng and L. Shao-chong, 2010; G. Donald and M H. Carl, 1985) . Our assumption in this paper is in line with that of (K. Hamzeh, M. Jelena and B. M. Vojislav, 2013; P. Ehsan and P. Massoud, 2009); that the cloud

Corresponding Author: A.O. Akingbesote, Department of computer science, University of ZuluL, South Africa.

center consists of a number of SMs that are allocated to consumers in the order of arrival of requests where the servers are physical servers. Furthermore, each task is assigned to only one server, and each server can only run at most one task at a time. Our service provisioning is SaaS where the consumers are web applications.

Two major things are needed to get our optimal service level.

- Accurate performance measure (Consumer Waiting Time).
- Good cost structure.

Researchers have used the queuing models to get some performance measures. For example, some have used the single server (K. Xiong and H. Perros, 2009), Multiple server (P. Ehsan and P. Massoud, 2009) or series network model (K. Hamzeh, M. Jelena and B.M. Vojislav, 2011; Xiaoming, H. Yifeng and G. Ling, 2011). All these have never accounted for the significant amount of time spent by cloud e-provider to update, scale up or down as part of consumer waiting time. Our contribution is the re-engineering of a typical e-cloud model as feed back networks of queues to get accurate response time which is then use to formulate our cost model to achieve our goal. In our model we set up the database server as a feed back service station.

The remainder of this paper is organized as follows. Section II discuses the related work. Section III introduces our proposed model. In Section IV, we have our simulation results and discussion. We have the conclusion in Section IV.

2. Literature Review:

Researchers have done much in the area of cloud e-market, however most works have been on implementation while the performance related ones have received less attention (P. Ehsan and P. Massoud, 2009). Some authors used a generalized model, see (K. Xiong and H. Perros, 2009). In (N. Xiaoming, H. Yifeng and G. Ling, 2011) the author modeled the cloud as series of queues with each service station as M/M/1 for optimal resource allocation. In this work, the authors modeled a typical cloud e-market as three concatenated queuing systems, which are schedule queue, computation queue and transmission queue. They then theoretically analyzed the relationship between the service response time and the allocated resources in each queuing system. The work of (P. Ehsan and P. Massoud, 2009) used the M/G/c to evaluate a cloud server firm with the assumption that the number of server machines are not restricted. The result of the author demonstrated the manner in which request response time and number of task in the system may be assessed with sufficient accuracy. The extension of (K. Hamzeh, M. Jelena and B.M. Vojislav, 2011) was carried out by (K. Hamzeh, 2013). This author viewed the cloud as queue in series where the author modeled the dispatcher queue as M/G/1/k and the others as M/G/c/k. The performance evaluation of this model using the response time as performance indicator gave a reasonable result that works in favor of cloud real response time

All these authorsn (K. Xiong and H. Perros, 2009; K. Hamzeh, M. Jelena and B.M. Vojislav, 2011; P. Ehsan and P. Massoud, 2009; N. Xiaoming, H. Yifeng and G. Ling, 2011; K. Hamzeh, 2013) have made sufficient contributions and these have given us the opportunity to make our own contribution as a result of the few observations we noticed. For example, the generalized modeling as discussed by (K. Xiong and H. Perros, 2009) may not reflect a real cloud e-marketplaces. Furthermore, modeling the cloud e-marketplaces as queue with a single server by (N. Xiaoming, H. Yifeng and G. Ling, 2011) in each station may not be real. In addition, the idea of an open buffer capacity by most authors (K. Xiong and H. Perros, 2009; K. Hamzeh, M. Jelena and B.M. Vojislav, 2011; P. Ehsan and P. Massoud, 2009; N. Xiaoming, H. Yifeng and G. Ling, 2011) may not be real. In addition, the significant amount of time spent by cloud e–market provider to update and collect statistical record for scaling up or down was never accounted for by these authors. This has made us to re-engineer a typical e-cloud model as feed back networks of queues and formulate our cost structure for the determination of optimal service level.

3. Proposed Model:

The proposed model of our cloud e-market is shown in Fig. 1. This model consists of service stations that are networked together. These are dispatcher queue, Web queue and the database queue.

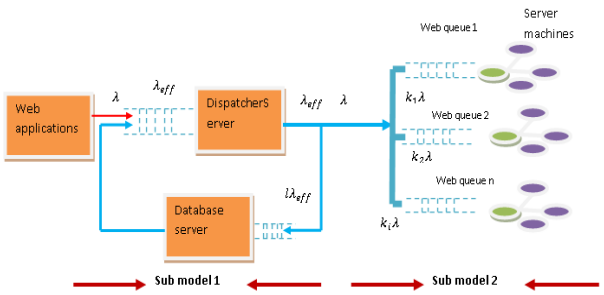

Fig. 1: Proposed Model of e-cloud market.

The dispatcher receives all in coming requests from both the consumers (λ_{eff}) and the database feedback ($l\lambda_{eff}$) and then schedules the requests to web queue servers. The web queue servers act as the real processors that provide the service based on First Come First Served (FCFS). In this model, each of the web queue stations has c (c= 1,2,…c) identical parallel servers with equal probability distribution of requests to each of the web queue stations that is $k_1 = k_2 = k_i$. As earlier said, the arrival and the service rate of the requests follow a Poisson process. One other assumption in this network is in line with (N. Xiaoming, H. Yifeng and G. Ling, 2011), that the latency of internal communication between the master server, database server and the web queue service stations is insignificant. We modeled the dispatcher and the database servers as M/M/1 queue respectively and that of web queue stations as M/M/c/k queue. To get our performance measure, we followed the six steps stated in (Prof. M. Bharathi, Prof. K.P. Sandeed and Prof. G.V. Poornim, 2012) and the law of conservation of flow (K. Leonard, 1975).

The server utilization ρ_1(for dispatcher) and ρ_2(database) of the two servers are given as:

$\rho_1 = \frac{\lambda_{eff}}{\mu_1}$ and $\rho_2 = \frac{\lambda_{eff}}{\mu_1}$ (1)

Unlike the work of some authors, for example (N. Xiaoming, H. Yifeng and G. Ling, 2011) where $\lambda_{eff} < \mu_1$ was assumed for steady state condition, our model need not to assume that but,

- $\lambda_{eff} \leq \mu_1$ and $l\lambda_{eff} \leq \mu_2$where μ_1 *and* μ_2 represent the service rate of dispatcher and database.
- Expected rate of flow into a state = Expected rate of flow out of that state. For the dispatcher it is given as

$(\lambda_{eff} + \mu_1)\ P_n = \lambda_{eff} P_{n-1} + \mu_1\ P_{n+1}$ (2)

and for database server, it is given as

$(\lambda_{eff} + \mu_2)\ P_n = \lambda_{eff} P_{n-1} + \mu_2\ P_{n+1}$ (3)

Therefore the probabilities of having one or more than one (n) web application(s) in the dispatcher P(disp) and database P(dbase) servers are:

$\mathrm{P(disp)}_1 = \frac{\lambda_{eff}}{\mu_1}\mathrm{P}_0$ and $\mathrm{P(dbase)}_1 = \frac{\lambda_{eff}}{\mu_2} P_0$ (4) $P(\mathrm{disp})_n = \left(\frac{\lambda_{eff}}{\mu_1}\right)^n P_0$ and $\mathrm{P(dbase)}_n = \left(\frac{\lambda_{eff}}{\mu_1}\right)^n P_0$ (5)

$P(\mathrm{disp})_n = \rho_2{}^n P_0$ and $\mathrm{P(dbase)}_n = \rho_2{}^n P_0$ (6)

Since the total probability = 1, then

$\sum_{i=0}^{N} P_i = 1 = \sum_{i=0}^{N} \rho_1{}^n P_0 = 1 = \rho_1{}^n \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1} = 1$ (7) and for the database server it is given as

$P_0 = \rho_2{}^n \left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right]^{-1}$ (8)

because the queue cannot build up unbounded, where ρ_1 and ρ_2 are = 1

$P(disp)_0 = \lim_{\rho_1 \to 1} \rho_1{}^n \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1}$ (9)

Using the L'Hospital rule [15], it follows that

$P(disp)_0 = \lim_{\rho_1 \to 1} \rho_1{}^n \left[\frac{-(N+1)\rho_1{}^N}{1-\rho_1}\right]^{-1} = \left[\frac{N+1}{1}\right]^{-1}$ (10)

$P(dbase)_0 = \lim_{\rho_2 \to 1} \rho_2{}^n \left[\frac{-(N+1)\rho_1{}^N}{1-\rho_1}\right]^{-1} = \left[\frac{N+1}{1}\right]^{-1}$ (11)

Combining the situation where ρ_1 and ρ_2 are = 1 for the dispatcher and dbase queues then we have

$$P(disp)_0 = \begin{cases} \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right] & if\ \rho_1 < 1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_1 = 1 \end{cases} \quad (12)$$

and

$$P(dbase)_0 = \begin{cases} \left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right] if \rho_2 < 1 \\ \left[\frac{N+1}{1}\right]^{-1} if\ \rho_2 = 1 \end{cases} \quad (13)$$

This implies that for all value of n, n = 0,1,2,3,….,N

$$P(disp)_n = \begin{cases} \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]\rho^n \quad if \rho_1 < 1 \\ \left[\frac{N+1}{1}\right]^{-1} \quad if\ \rho_1 = 1 \end{cases} \quad (14)$$

and

$$P(dbase)_n = \begin{cases} \left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right]\rho^n \quad if \rho_2 < 1 \\ \left[\frac{N+1}{1}\right]^{-1} if\ \rho_2 = 1 \end{cases} \quad (15)$$

In this experiment, ρ_1, ρ_2 ad ρ_2 are less than 1. Therefore, the expected number of web applications in dispatcher and database system are:

$E(web_{disp}) = \sum_{n=0}^{N} nP_n = \sum_{n=0}^{N} n\rho_1{}^n P_0$

$= \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1} \rho_1 \left[\frac{(1+\rho_1{}^{N+1})-(N+1)\rho_1{}^N(1-\rho_1)}{[1-\rho_1]^2}\right]$ (16)

$E(webdbase) =$

$\left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right]^{-1} \rho \left[\frac{(1+\rho_2{}^{N+1})-(N+1)\rho_2{}^N(1-\rho_2)}{[1-\rho_2]^2}\right]$ (17)

Two things we have done in our re-engineering process. The first is the modification of the little's formulae to determine the expected number of web applications in the dispatcher and database queues. This is because the expected number of the web applications in dispatcher queue for example

$E(\text{disp. queue}) = \sum_{n=0}^{N}(n-1)P_n$

$= \sum_{n=0}^{N} nP_n - \sum_{n=0}^{N} P_n = E(web_{disp})-(1-P_0)$ (18)

but using Little formulae in our model we have $E(\text{dsisp. queue}) = E(web_{disp}) - \frac{\lambda_{eff}}{\mu_2}$

This is only true when the mean arrival rate is λ_{eff}. However, from Eq. 18, $1-P_0 < \frac{\lambda_{eff}}{\mu_1}$ because the mean arrival rate is λ_{eff} as long as there is vacancy in the queue and it is zero when the system is full. This gives us the motivation to define our real effective arrival rate as λ_{eff}'. Therefore applying Eq. 18 and the little's formulae as $\frac{\lambda_{eff}'}{\mu_1} = 1-P_0$ or $\lambda_{eff}' = \mu_1(1-P_0)$.

Thus, we can rewrite Eq. 18 as

$E(dispqueue) = E(web_{disp}) - \frac{\lambda_{eff}'}{\mu_1}$ (19)

This apply to database queue which is then written as

$E(dbasequeue) = E(web_{disp}) - \frac{\lambda_{eff}'}{\mu_2}$ (20)

The second which is the re-engineering process is the calculation of the average waiting time in both the queue and system of the dispatcher and the database where most authors like (N. Xiaoming, H. Yifeng and G. Ling, 2011)multiply $\lambda_{eff}'^{-1}$ *by* $E(webdisp)$ as the waiting time in the dispatcher system or $\lambda_{eff}'^{-1}$ *by* $E(\text{disp. queue})$ as the waiting time in the dispatcher queue.

We based our waiting time both in dispatcher system (Ws_{disp}) and the queue (Wq_{disp}) as

$$Ws_{disp} = \frac{E(LS_{disp})}{\lambda_{eff}'} * Ex_{visitdisp} \quad (21)$$

$$Wq_{ddisp} = \left(Ws_{disp} - \frac{1}{\mu_1}\right) * Ex_{visitdisp} \quad (22)$$

$$Wq_{ddbase} = \left(Ws_{disp} - \frac{1}{\mu_2}\right) * Ex_{visitdbase} \quad (23)$$

$$Ws_{dbase} = \frac{E(LS_{disp})}{\lambda_{eff}'} Ex_{visitdbase} \quad (24)$$

Where $Ex_{visitdisp}$ represents the number of visit(s) to the dispatcher which is given as

$Ex_{visit\ disp} = \frac{1}{1-\lambda_{eff}'}$ and that of the database as

$$\mathrm{Ex}_{\text{visit dbase}} = \frac{1}{1-\lambda_{\text{eff}}}$$

Each of our web queue service center is modeled as M/M/c/k with equal service distribution $k_i\lambda_{\text{eff}}$ as shown in our model of Fig. 1 where i= 1,2,3,….j represent the number of web queue service station and each station has equal or identity servers (c) with the same service rate μ. For example, web queue service station 1 whose arrival rate is $k_i\lambda_{\text{eff}}$ may have 3 servers where the total service rate in that station is 3μ. Therefore, for each web queue service station the mean arrival rate is given by

$$k_i\lambda_{\text{eff}\,n} = \begin{cases} \mathrm{k_i}\lambda_{\text{eff}} & \text{for } n = 0,1,2 \ldots . N-1 \\ 0 & \text{for } n = N, N+1, \ldots \ldots \end{cases} \quad (25)$$

and

$$\mu_n = \begin{cases} n\mu & \text{for } n = 0,1,2 \ldots . c-1 \\ c\mu & \text{for } n = c, c+1, \ldots \ldots \end{cases} \quad (26)$$

where $1 < c < N$

From difference equation, giving steady- state probabilities P_n and P_0, we have

$$P_n = \frac{\mathrm{k_i}\lambda_{\text{eff}\ 0}\ \mathrm{k_i}\lambda_{\text{eff}\ 1}, \ldots \ldots, \mathrm{k_i}\lambda_{\text{eff}\ n-1}}{\mu_1\mu_2 \ldots \ldots \ldots \ldots \ldots \mu_n} P_0 \quad (27)$$

$$P_0{}^{-1} = 1 + \sum_{n=1}^{\infty}\left[\frac{\mathrm{k_i}\lambda_{\text{eff}\ 0}\ \mathrm{k_i}\lambda_{\text{eff}\ 1} \ldots \mathrm{k_i}\lambda_{\text{eff}\ n-1}}{\mu_1\mu_2 \ldots \ldots \ldots \mu_n}\right] \quad (28)$$

substituting the value $k_i\lambda_{\text{eff}\,n}$ and μ_n

$$P_0 = \left[\sum_{n=0}^{c-1}\frac{1}{n!}\left(\frac{\mathrm{k_i}\lambda_{\text{eff}}}{\mu}\right)^n + \frac{1}{c!}\left(\frac{\mathrm{k_i}\lambda_{\text{eff}}}{\mu}\right)^c \sum_{n=c}^{k}\left(\frac{\mathrm{k_i}\lambda_{\text{eff}}}{\mu}\right)^{n-c}\right]^{-1)} \quad (29)$$

and

$$\mathrm{P_n} = \begin{cases} \frac{1}{n!}\left(\frac{\mathrm{k_i}\lambda_{\text{eff}}}{\mu}\right)^n P_0 & \text{for } \mathrm{n} \le \mathrm{c} \\ \frac{1}{\mathrm{c!c^{n-c}}}\left(\frac{\mathrm{k_i}\lambda_{\text{eff}}}{\mu}\right)^n P_0 & \text{for } \mathrm{c} < n \le \mathrm{k} \\ 0 & \text{for } \mathrm{n} \le \mathrm{k} \end{cases} \quad (30)$$

Therefore the expected number of web application in the queue of each service station *i* is

$$E\ (web\ queue_i) = \sum_{n-c}^{N} n - c\,\frac{1}{\mathrm{c!c^{n-c}}}\left(\frac{\mathrm{k_i}\lambda_{\text{eff}}}{\mu}\right)^n P_0 \quad (31)$$

but the server utilization in each web queue service station *i* is

$\rho_i = \frac{\mathrm{k_i}\lambda_{eff}}{c\mu_i}$ substituting ρ_i in Eq. 31 and differentiating

$\frac{d}{d\rho_i}\left[\frac{1-\rho_i{}^{N-c+1}}{1-\rho_i}\right]$ we have

$$E\ (web\ queue_i) = P_0\ \frac{\mathrm{k_i}\lambda_{eff}}{\mu}\frac{\rho_i}{c!(1-\rho_i)^2}[1 - \rho_i{}^{N-c} - (N-c)(1-\rho_i)\rho_i{}^{N-c}] \quad (32)$$

The expected number of web applications in the system is given as

$$E\ (\text{web system}_i) = \sum_{n=0}^{c-1} nP_n + \sum_{n=c}^{N} nP_n \quad (33)$$

Therefore, our modified little's formulae then is

$$E\ (web\ system_i) = E\left(web\ queue_i\right) + \frac{\mathrm{k_i}\lambda_{eff}{}^{i}}{\mu} \quad (34)$$

Where $\mathrm{k_i}\lambda_{eff}{}^{i}$ is the real effective arrival rate given as

$k_i\lambda_{eff}{}^{i} = \mu[c - \sum_{n=0}^{c-1}(c-n)P_n]$

Our web system and queue waiting time are

$$Wsystem_i = \left[k_i\lambda_{eff}{}^{i}\right]^{-1} * E(\text{web system}_i) \quad (35)$$

$$Wqueue_i = \left[k_i\lambda_{eff}{}^{i}\right]^{-1} * E(\text{web queue}_i) \quad (36)$$

The average mean waiting time in the queue and system of all the web queue service stations are given as

$$Wqueue_{ave} = \frac{1}{j}\sum_{i=0}^{j} Wqueu_i \quad (36)$$

$$Wsystem_{ave} = \frac{1}{j}\sum_{i=0}^{j} Wsystem_i \quad (37)$$

In this research, the performance measure that was used as part of the determinant factor in our cost structure was the $Wqueue_{ave}$. Therefore, the total queue waiting time in all the service stations is given as

$$Wq_{total} = Wq_{ddisp} + Wq_{dbase} + Wqueue_{ave} \quad (38)$$

A. Cost Model:

We based our cost structure on (A.T. Hamdy, 2011; R. Harsharger, 2013) where we attempt to balance two conflicting issues.

- Cost of offering the service
- Cost of delay in offering the service

The allocated resources considered in this cost model are all the servers in all service stations.
Letting c represent the service level, then the expected total cost is formulated as

$$ETC(x) = EOC(x) = EWC(x) \quad (39)$$

Where

ETC(x) = Expected total cost per unit time

EoC(x) = Expected costof operating the facility per unit time

ETC(x) = Expected cost waiting per unit time

$$ETC(x) = k * Wq_{total} + \left(\frac{c}{10} + 1\right) \quad (40)$$

k = Cost value of waiting in the queue

4. Simulation:

We run the simulation to evaluate the performance of our model using Arena software. We set the dispatcher arrival rate to 50 web applications per minutes and the probability distribution of each of the web queue service station was set to 0.43 respectively. The database arrival was given a small probability distribution of 0.14. Each of the web queue service stations has a buffer capacity of 200. Due to the scheduling service distribution to the service stations, the dispatcher buffer capacity was 400. We created 2 web queue service stations with each station having 3 server machines with identical service time of 0.07 per minutes. The base time and the time unit were set to minutes with 10 replications and also with average of 299,5000 minutes spent on each experiment. At the end of each experiment, the Wq_{total} is recorded and used in our linear cost model. We then increase the server machine in each web queue station by one. The results obtained are shown in Tables 1- 4 and Figures 2-5.

RESULTS AND DISCUSSION

Our results in Table 1 and Fig.2 show the arrival and the departure of the web applications in and out of each service stations. A total of 15,000 of web applications were received by the dispatcher and these were distributed to each service station. An average of 2107.6 requests was generated automatically for the statistical update of the database which has significant influence on the consumer waiting time. This is in line with our re-engineering process as this number of visit to the database has a significant role to play in the total waiting time of web application in the queue and in the system. Our re-engineering process is not to have a decrease in waiting time when compared to the work of other authors (N. Xiaoming, H. Yifeng and G. Ling, 2011) but towards an accurate waiting time to reflect a real typical cloud e-market. A total of 15,000 web applications was serviced by the 3 service stations with identical web application requests in each station. This is in line with our model of Fig. 1where total number of web applications that went into the system (λ) is also the same that went to the web queue stations. That is $\lambda = \sum_{i=0}^{j} k_i\lambda_{eff}$ where j = 2 in this experiment.

The results in Table 2 and Fig. 3 represent the data collected at the end of every experiment where the average queue waiting time is derived using our Eq. 36. In this table and figure, as the number of server machine increases, the average queue waiting time decreases in each service station. Though at the point of using 16- 20

server machines, the waiting time remain constant. The reason was due to constant service rate of our dispatcher and database throughout the experiment. At those points, the web applications have zero waiting time in both web queue1 and 2 respectively but the dispatcher and database have some web applications on queue. The average queue waiting time was then used in Table 3 and Fig. 4 to get our total cost based on Eq. 40. In this table, our optimal service level is at the point where we have the $Min(ETC_i)$ which is 10 in this experiment. At this point, one critical question one need to ask is, using these optimal sever machines , can we meet the Service level agreement (SLA)?. Since SLA is being decided by cloud e-market provider, for the purpose of this experiment, we formulated our SLA as $\frac{\sum_{i=0}^{N} Wq_{total_i}}{N}$ which is shown in Table 3.

Considering the SLA in this experiment, our result revealed that using 10 server machines will not only meet the SLA but will further bring a reduction of 0.47 minutes (0.509 - 0.452) to consumer waiting time as show in Fig. 5.

Table 1: Web Application In and out.

Queue	Average Number In	Average Number out
DatabQ	2107.60	2107.60
Web Q1	7462.30	7462.30
Web Q2	7537.70	7537.70
Web.Appl	15000.00	15000.00

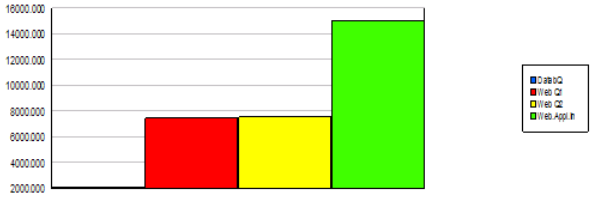

Fig. 2: Web Application Summary.

Table 2: Server machine and Average waiting time.

Server machine	Ave. Wqtotal
4	0.943
6	0.653
8	0.522
10	0.452
12	0.417
14	0.402
16	0.397
18	0.396
20	0.397

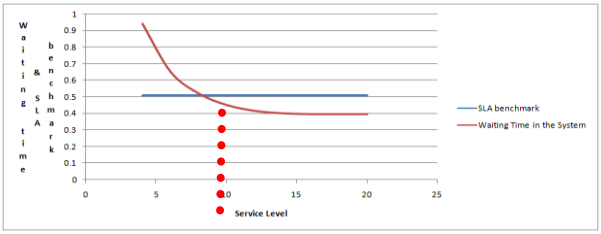

Fig. 3: Waiting Time, SLA and Service level.

Table 3: Total Cost.

Experiment (*i*)	Server machine (Q)	Average waiting Time (P)	Cost of waiting EWC(x)	Operating Cost EOP(x)	Total Cost ETC(x)
1	4	0.943	4.72	1.4	6.12
2	6	0.653	3.27	1.6	4.87
3	8	0.522	2.61	1.8	4.41
4	10	0.452	2.26	2	4.26
5	12	0.417	2.09	2.2	4.29
6	14	0.402	2.01	2.4	4.41
7	16	0.397	1.99	2.6	4.59
8	18	0.396	1.98	2.8	4.78
9	20	0.397	1.99	3	4.99
	SLA	0.509			

The overall analysis is in Table 4. In the Table, we observed that using $4 \leq sms < 10$ will reduce cost but will be at the disadvantage of the consumers as indicated by the negative sign because the waiting time

agreement has been breached (0.509). But if $sms = 10$ as ealier said, there is a gain of 0.47 and also the provider has the minimum cost reduction. At this point, the cloud e-market provider has minimized cost at the same time satisfied the consumers waiting time. Any further increase in the server machine will still reduce cost but at a diminishing rate but further improve consumer waiting time.

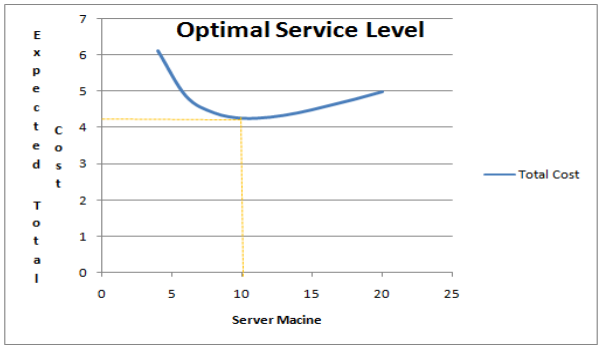

Fig. 4: Optimal Service Level.

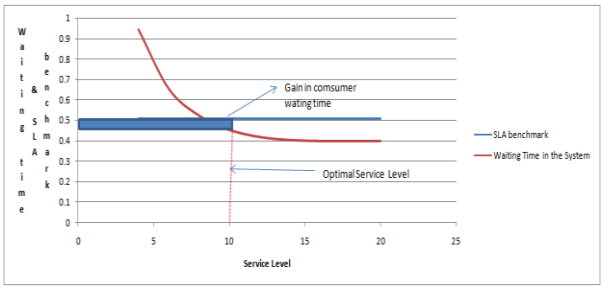

Fig. 5: Comparison of SLA with web application waiting time.

Table 4: Overall Analysis.

Server Machine	Consumer Time (Loss/Gain in mins.)	Consumer - SLA	Cloud e-Provider
4	-0.434	Breach of SLA	Cost Reduction
6	-0.144	Breach of SLA	Cost Reduction
8	-0.013	Breach of SLA	Cost Reduction
10	0.057	Gain in Service Time	Optimal Cost
12	0.092	Gain in Service Time	Increase in Cost
14	0.107	Gain in Service Time	Increase in Cost
16	0.112	No effect	Increase in Cost
18	0.113	No effect	Increase in Cost
20	0.112	No effect	Increase in Cost

Conclusion:

With the migration of consumers to the cloud e-marketplaces for good qaulity of service based on pay-per-go, the cloud e-market providers are concerned mainly with two major issues. These are profit maximization through cost minimization and consumer's satisfaction. Balancing the trade off is what this paper has addressed. This was done by first re-engineering a typical cloud market as feed back proceess in order to achieve dramatic improvement in critical area of performance such as cost and accurate waiting time. Secondly, simulating this model with data to achieve our optimal service level. We hope to buid our future work on the performance of pre-emptive nested scheduling in a typical e-cloud marketplaces.

REFERENCES

Deng, W., F. Liu, H. Jin and C. Wu, 2013. "Smart DPSS: Cost-Minimizing Multi-source Power Supply for Datacenters with Arbitrary Demand," In Proc. Of ICDCS, Berkeley, California, USA, May 21-24.

Donald, G. and M.H. Carl, 1985. "Fundamental of Queue Theory". Second edition. Canada: John Wiley and Sons Inc, 2-8.

Ehsan, P. and P. Massoud, 2009. "Minimizing Data Center Cooling and Server Power Costs," in Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design, New York, USA, 145-150.

Hamdy, A.T., 2011. "Operation Research: An Introduction to Operation Research". Ninth edition. Upper Saddle River, New Jersey: Pearson Education Inc., 662-673.

Hamzeh, K., 2013. "Performance Modeling of Cloud Computing Centers," A Ph.D thesis. Manitoba, Canada: University of Manitoba Winnipeg.

Hamzeh, K., M. Jelena and B.M. Vojislav, 2011. "Modeling of Cloud Computing Centers Using M/G/m Queue" in Proceedings of1st International Conference on Distributed Computing Systems Workshops, Minneapolis, MN, 87-92.

Hamzeh, K., M. Jelena and B.M. Vojislav, 2013. "Performance Evaluation of Cloud Data Centers with Batch Task Arrivals," In T.M Hussein, K. Burak, editors, Cmmunication Infrastructures for Cloud Computing, 199-223.

Hao-peng, C. and L. Shao-chong, 2010. "A queueing-based model for performance management on cloud," in proceedings of 6th International conference on Advanced Information Management and Service (IMS), Seoul, 83-88.

Harsharger, R., 2013. "Mathemathetical Application for the management, Life and Social Sciences". Tenth Edition. USA: Richard Stratton, 668-676.

IDC. The Economics of Virtualization: Moving toward an application-based cost model. [document on the internet], c2013[cited 2013 Aug. 03]. Available from: http://www.Vmware.com/files/pdf/ irtualization-application-based-cost-model-WP-EN.pdf.

Kim, C.C., L. Smith, H. Thorne and R.W. Hilton, 2008. "Management Accounting: Information for managing and creating value. Maidenhead", Berkshire: McGraw-Hill Higher Education, 726-728.

Leonard, K., 1975. "Queuing System", Canada: John Wiley @ son, 3(1): 3-7.

Prof. M. Bharathi, Prof. K.P. Sandeed and Prof. G.V. Poornim, 2012. "Performance factors of cloud computing data centers using M/G/m/m+r Queuing system," IOSR Journal of Engineering (IOSRJEN), 2(9): 06-10.

Rahul, G., S.T. Kishor, K.N. Vijay and S.K. Dong, 2010. "End-to-End Performability Analysis for Infrastructure-as-a-Service Cloud: An Interacting Stochastic Models Approach," in proceeding of the 16th ieee PRDC, 125-132.

Sundarapandian, V., 2009. "Probability, stattistics and quueuing theory". New Delhi, India: PHI Learning Private Limited, 732-735.

Xiaoming, N., H. Yifeng and G. Ling, 2011. "Optimal resource allocation for multimedia cloud based on queuing model," presented at the Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on Hangzhou.

Xiong, K. and H. Perros, 2009. "Service performance and analysis in cloud computing," presented in ICWS, International Workshop on Cloud Computing; Los Angeles, CA, July 6-10.