Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 8 No. 2, June, 2017 - www.cisdijournal.net

Balancing Service Level and Performance Measures in Cloud E-Marketplaces Using Aspiration Model

Akingbesote, A.O., Adigun, M.O. & Xulu, S.
Department of Computer Science
University of ZuluLand,
P. O. Box X1001
KwaDlangezwa, South Africa
Email:oluwamodimu2012 @gmail.com

Kaseeram, I
Department of Economics
University of Zulu land, X1001,
KwaDlangezwa, 3886, South Africa
KaseeramI@unizulu.ac.za

ABSTRACT

As consumers drift to the Cloud E-marketplaces for services, the service level and the performance measures are important considerations in cloud E-market places from a cost efficiency perspective. In this regard, the consumers' waiting time and Idle percentage rate of servers are critical measures to consider. Formulating a decision model that strikes a balance between these two is a challenge and our goal. Most literature has been presenting their decision model based on cost model but the viability of this model depends on how well the cost parameters can be estimated which indeed are difficult. We extend the body of knowledge in the context of Cloud E-marketplaces by using the aspiration model. We first use the queuing theory to formulate our mathematical model to obtain our conflicting measure of performance. This is then use to determine an acceptable range for the service level through the use of our aspiration model. We then simulate to demonstrate the real life scenario. Our results reveal an acceptable range for the service level by specifying reasonable limits the e-market decision maker wishes to reach on the conflicting measure of performance. Our contribution is in the provisioning of regulatory mechanism that gives an acceptable range of server resources, which is an essential component of Cloud resource management.

Keywords — Waiting time, percentage of idleness, provider, consumer;

CISDI Journal Reference Format
Akingbesote, A.O., Adigun, M.O., Xulu, S. & Kaseeram, I. (2017): Balancing Service Level and Performance Measures in Cloud E-Marketplaces Using Aspiration Model. Computing, Information Systems, Development Informatics & Allied Research Journal. Vol 8 No 2. Pp 61-74. Available online at www.cisdijournal.net

1. INTRODUCTION

The cloud E-market place is a virtual environment for buying and selling of services (Akingbesote A.O et al. 2013). The service level in the cloud e-marketplace is a function of the service rate and the number of the parallel servers. There are two major participants in cloud e-marketplaces, the clients or consumers and the cloud E-market provider that does the service provisioning. On the service provider side, the issue of profit maximization is crucial and is a function of cost of the service incurred in service provisioning or delivery. A reduction in cost will increase profit. Two major parameters are required to reduce costs; the first is the number of server machines employed by service provider during service provisioning. The second is the performance measures which are used to determine the effectiveness of the cloud e-marketplaces. One important of these measures is the consumer waiting time. This is the time spent to respond to consumers by these servers for products or requests. This waiting time, which is a key source of competitive advantage, is so important that traditional marketplaces like MCDonalds in 1990s offered consumers their meal free of charge if the order was not served within 2 minutes (Carlos et al. 2013).

In this competitive market, every provider wants to meet the SLA (Service Level Agreement) and if possible, perform better than the minimum requirements stipulated by the SLA especially in terms of waiting time and other measure of effectiveness like the percentage of Idle machines to attract more consumers. A high level of service will raise the cost to the E-cloud provider and will result in lower dissatisfaction costs. But a low level of service may be inexpensive, at least in the short run, but may incur high consumer dissatisfaction costs, such as loss of future business and actual processing costs of complaints. Therefore, striking a balance between this performance measures, for example, consumer waiting time and the service level is a challenge (Mustafa & McCluskey 2009) and require a good decision model to approach this challenge.

In order to tackle this challenge, researchers have put forward different solutions. For example, (Nan et al. 2011)(Goudarzi & Pedram 2011) (Nagendram 2011) use the optimization approach to solve this problem, but the constraints look too difficult to understand. The use of resource over provisioning policy is sometime adopted using the worst case scenario as the benchmark which can lead to a largely suboptimal utilization of the hosting environment resources (Ferretti et al. 2010)(Goudarzi & Pedram 2011)(Vinothina et al. 2012) . Other researchers concentrate on the use of cost model both in the field of cloud and others to formulate their decision model (Akingbesote A.O et al. 2013) (Akingbesote et al. n.d.) (Kembe 2012). But literature for example, (Taha n.d.) reveals that the viability of this model depends on how well the cost parameters can be estimated which indeed are also difficult. By viability, we mean the quality of having a reasonable chance of success. For example, to determine the cost of waiting for x web applications (clients or consumers) will require so many assumptions by a human decision maker as a result of the dynamic change and uncertainty of the waiting time.

This and other fundamental obstacles make it difficult to apply them to many real world problems. Even previous research in experimental economics and cognitive psychology has shown that human decision makers often do not adhere to fully rational behavior (Rosenfeld & Kraus n.d.). The work of (Kahneman & Tversky 2007) also shown that individuals often deviate from optimal behavior as prescribed by Expected Utility Theory. In addition, decision makers do not know the quantitative structure of the environment in which they act as a result of lack of complete information. Even when people act rationally, they cannot always compute the optimal solution for a given problem, as they lack the required facts to arrive at this decision. Therefore expecting optimality based solutions seems to be difficult and sometimes unrealistic.

In addition, all these works have concentrated on consumer waiting time without much attention to other conflicting measures of performance like the percentage of server idleness. We extend existing and widely adopted aspiration theory to Cloud e-marketplaces. The idea is not to solve for optimal solution but to reveal an acceptable range for the service level by specifying reasonable limits the provider wishes to reach on the conflicting measure of performance. We first model the cloud E-marketplace as networks of queues with a feedback. The queuing theory is used to formulate our mathematical model to obtain our conflicting measure of performance. The first part is similar to our previous work (Akingbesote A.O et al. 2013). This is then use to determine an acceptable range for the service level which is the crux of this research. The simulation is done to demonstrate the real life scenario. Two conflicting performance measures are used in this research. These are the consumers' waiting time and the percentage of server idleness. The remainder of this paper is organized as follows. Section II discusses the related work. Section III introduces our mathematical model description with the numerical and simulation set up. In Section IV, we present our results and discussions. We provide a conclusion in Section V.

2. LITERATURE REVIEW

Research in cloud e-marketplace has received much attention in so many sub-domains including security (Alvi et al. 2012), energy (Wang n.d.) and privacy (Sun et al. 2013) but little has been done in the area of optimization for resource management with regard to cloud performance(Guo et al. 2014). On the cloud performance issue, for example, in (Xiong & Perros 2009), the authors use the M/M/1 model to address three things; these are the level of Quality of Service (QoS) that can be guaranteed given service resources, the number of service resources that are required to ensure that customer services can be guaranteed in term of the percentile and the number of customers to be supported to ensure that customer services can be guaranteed in term of the percentile of response time. The work is further extended in (Nan et al. 2011) as a series of queues with each service station modelled as M/M/1 for optimal resource allocation. The authors model a typical cloud e-market as three concatenated queuing systems, consisting of a 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 (Pakbaznia & Pedram 2009) uses the M/G/c to evaluate a cloud server firm with the assumption that the numbers of server machines are not restricted. The result of the author demonstrates the manner in which request response time and number of task in the system may be assessed with sufficient accuracy. The work of (Wu et al. 2011) focuses on scheduling customer requests for Software as a Service (SaaS) providers with the explicit aim of cost minimization with dynamic demands handling. The author uses the mapping and scheduling mechanism to solve the customer side dynamics demand and resource level heterogeneity problem. While this work is in the area of Enterprise Application, that of (Goudarzi & Pedram 2011) considers the Service Level Agreement (SLA) based allocation using a distributed solution approach for managing data storage, communication and processing resources. The simulation results of the work produce a solution very close to the optimum. The work of (Nan et al. 2011) proposes a dynamic Optimization model in order to optimize the performance of multiple request and services in cloud computing. This is done using M/M/m queuing theory with the assumption of infinite buffer capacity. The authors use synthesis optimization, function and strategy to achieve their objective. The simulation results show a better performance over the classical optimization methods in terms of the number of customers' average waiting time and average queuing length.

All these authors have made sufficient contribution towards providing acceptable optimum solutions. However, the few observations we noticed in these works have given us the opportunity to make our contribution. For example, the generalized approached (M/M/1) use by (Xiong & Perros 2009) may not reflect today's Cloud computing due to large numbers of consumers using the cloud. Also the work of (Nan et al. 2011) which is similar to our model assumes an infinite buffer capacity by contrast we base our work on a finite capacity because, Our opinion is in line with (Chiang & Ouyang 2014), that there is always limitations to what the server can contain. Although this work discusses the optimization of the performance measures but it fails to discuss or offer a solution in the formulating of a model to optimize the service level, which our study addresses. Additionally, our work is further differentiated with the introduction of dedicated database server that collates the statistical information as to when the system is to be scaled up or down. The use of the aspiration level approach differentiates our work from all these authors which to the best of our knowledge has not appeared in the literature in the context of E-cloud marketplaces. Furthermore, the maximization problem proposed in (Goudarzi & Pedram 2011) is a hard problem in the sense that the constraints set up in this work looks very difficult to understand and a simple problem solver may not be able to arrive at a solution.

In (Akingbesote A.O et al. 2013) (Akingbesote et al. n.d.) (Kembe 2012), the authors use the queuing theory to obtain their performance measures. This is then used as part of their decision mechanism to formulate their cost model which estimates the expected cost of waiting and the expected operational costs incurred. The optimal value is derived at the point where the service level obtains the minimum value. While this result seems to have produced the optimal service level, literature [10] (Taha n.d.) reveals that the viability of this model depends on how well the cost parameters can be estimated which indeed are difficult and therefore making it almost impossible to get an accurate optimal value. We extend this work by proposing the use of the aspiration model which reveals an acceptable range for the service level. This is done by specifying reasonable limits the provider wishes to reach on the conflicting measure of performance (i.e., minimum waiting time and maximum server usage) which to the best our knowledge has not reflected in any literature. We approach the problem by modeling the cloud as network of queues with a feedback using the queuing theory. The contribution of this work is the providing of regular information in predicting the outcome of an event given the aspiration constraints. This will be a useful model by providing the feasible acceptable range of server resources to be used, which is an essential component of Cloud resource management**.**

3. PROPOSED MODEL

The proposed model of our cloud e-market is shown in Figure 1. This model consists of two sub models which are the sub model 1 and 2 respectively. The sub model 1 consists of the incoming web or consumer application with the dispatcher and the database feedback. The sub model 2 consists of three service stations that are networked together. The processing of the applications takes place at these service stations.

The dispatcher receives all in coming requests λ_d from both the consumers (λ) and the database feedback $(l\lambda_{eff})$ which is then scheduled to the web queue servers. Though the number of requests arriving the dispatcher is $\lambda + l\lambda_{eff} = \lambda_d$, due to the fact that not all consumers can join the dispatcher as a result of the limited server capacity (M/M/1/k) of our model. Therefore, this gives us the motivation to define our real effective arrival rate as λ_{eff} $(\text{where } \lambda_{\text{eff}} < \lambda_d)$ as shown in the proposed model. Therefore, $\lambda + l\lambda_{eff} = \lambda_{\text{eff}}$. We first use the λ_d for our mathematical formulation and later revert to our λ_{eff} value. The word consumer in this paper is referred to as an application requesting service from the provider (Goudarzi & Pedram 2011).

The web queue servers act as the real processors that provide the service on a First Come First Served (FCFS) basis. Each of the web queue stations has c (c = 1,2,…c) identical parallel servers with equal probability distribution $k_i\lambda$ of requests to each web queue station. Where, $k_1 = k_2 = k_t$. Moreover, arrival and the service rate of the requests follow a Poisson process. One other assumption in this network is in line with (Nan et al. 2011) , that the latency of internal communication between the dispatcher server, database server and the web queue service stations is insignificant. Our general idea is to derive $P(i)_n$ as a function of $\lambda(i)_n$, $\rho(i)$ and $\mu(i)_n$. Where *i = 1,2,3*= disp, dbase, and the each of the service stations respectively. For example, $P(disp)_n$ represents the steady state probability of n consumers in the dispatcher queue while $\lambda(\text{web queue } 1)_n$ represents the number of consumers arriving to the web queue station 1. Also $\mu(disp)_n$ represents the departure or service rate in the dispatcher server given n numbers of consumers in the system and $\rho(i)$ is the server utilization in i^{th} server machine.

These probabilities are then used to determine the conflicting measures of performance which are the average queue length, average waiting time and the average utilization of the facility. We model the dispatcher and the database servers as M/M/1/k queue respectively and that of web queue stations as M/M/c/k queue. Our derived conflicting measure of performance is derived based using the six steps stated in (Sundarapandian 2009) and the law of conservation of flow (Gross & John 1985)[14]. We first model the dispatcher and the database and later model that of the web queue stations

A. Modelling the Dispatcher and the Database

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

$$\rho_1 = \frac{\lambda_d}{\mu_1} \text{ and } \rho_2 = \frac{\lambda_d}{\mu_2} \qquad (1)$$

Most authors, for example[11] assumed $\lambda_{eff} < \mu_1$ for steady state condition, but in this model, our assumptions are

- $\lambda_d \leq \mu_1$ and $l\lambda_d \leq \mu_2$ where μ_1 and μ_2 are the dispatcher and database service rate of respectively.
- Expected rate of flow into a state = Expected rate of flow out of that state.

We first model the dispatcher queue as

$$(\lambda_d + \mu_1) P_n = \lambda_d P_{n-1} + \mu_1 P_{n+1} \qquad (2)$$

and for the database server, it is given as

$$l\lambda_d + \mu_2) P_n = l\lambda_d P_{n-1} + \mu_2 P_{n+1} \qquad (3)$$

Therefore the probability of having one (n=1) consumers in the dispatcher P(disp) and database P(dbase) servers are:

$$\mathbf{P(disp)_1} = \frac{\lambda_d}{\mu_1} \mathrm{P_0} \text{ and } \mathrm{P(dbase)_1} = \frac{l\lambda_d}{\mu_2} P_0 \quad (4)$$

and

$$P(\mathrm{disp})_n = \left(\frac{\lambda_d}{\mu_1}\right)^n P_0 \text{ and } \mathrm{P(dbase)}_n = \left(\frac{l\lambda_d}{\mu_1}\right)^n P_0 \qquad (5)$$

$$P(\mathrm{disp})_n = \rho_1{}^n P_0 \text{ and } \mathrm{P(dbase)}_n = \rho_2{}^n P_0 \quad (6)$$

Since the total probability = 1, then

$$\sum_{i=0}^{N} P_i = \sum_{i=0}^{N} \rho_1{}^n P_0 = \rho_1{}^n \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1} = 1 \qquad (7)$$

and for the database server it is given as

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

Eq. 7 and 8 hold when $\lambda_d = \mu_1$ and $l\lambda_d = \mu_2$. But when $\lambda_d \neq \mu_1$ and $l\lambda_d \neq \mu_2$ then

$$P(disp)_0 = \lim_{\rho_2 \to 1} \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1} \qquad (9)$$

Using the L'Hospital rule (Sundarapandian 2009) it follows that

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

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

Combining the two situations when, $\lambda_d = \mu_1$, $l\,\lambda_d = \mu_2$ and when $\lambda_d \neq \mu_1$ and $l\,\lambda_d \neq \mu_2$ for the dispatcher and the database servers then

$$P(disp)_0 = \begin{cases}\left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1} & if\ \rho_1 < 1\ or\ \ \lambda_d \neq \mu_1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_1 = 1\ or\ \ \lambda_d = \mu_1\end{cases} \qquad (12)$$

and

$$P(dbase)_0 = \begin{cases}\left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right]^{-1} & if\ \rho_2 < 1\ or\ \ l\lambda_d \neq \mu_1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_2 = 1\ or\ \ l\lambda_d = \mu_1\end{cases} \qquad (13)$$

and

$$P(disp)_n = \begin{cases}\left[\frac{1-\rho_1{}^{N+1}}{\rho_1{}^n(1-\rho_1)}\right]^{-1} & if\ \rho_1 < 1\ or\ \ \lambda_d \neq \mu_1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_1 = 1\ or\ \ \lambda_d = \mu_1\end{cases} \qquad (14)$$

and

$$P(dbase)_n = \begin{cases}\left[\frac{1-\rho_2{}^{N+1}}{\rho_2{}^n(1-\rho_2)}\right] & if\ \rho_2 < 1\ or\ \ l\lambda_d \neq \mu_1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_2 = 1\ or\ \ l\lambda_d = \mu_1\end{cases} \qquad (15)$$

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

$$P(disp)_n = \begin{cases}\left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]\rho^n & if\ \rho_1 < 1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_1 = 1\end{cases} \qquad (16)$$

and

$$P(dbase)_n = \begin{cases}\left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right]\rho^n & if\ \rho_2 < 1 \\ \left[\frac{N+1}{1}\right]^{-1} & if\ \rho_2 = 1\end{cases} \qquad (17)$$

In this experiment, ρ_1 and ρ_2 is less than 1. Therefore, the expected number of consumers in dispatcher E(web_{disp}) and database $\mathbf{E(webdbase)}$ system are:

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

$$= \left[\frac{1-\rho_1{}^{N+1}}{1-\rho_1}\right]^{-1} \rho_1 \left[\frac{(1+\rho_1{}^{N+1}) - (N+1)\rho_1{}^N (1-\rho_1)}{[1-\rho_1]^2}\right] \quad (18)$$

$$\mathbf{E(webdbase)} = \left[\frac{1-\rho_2{}^{N+1}}{1-\rho_2}\right]^{-1} \rho \left[\frac{(1+\rho_2{}^{N+1}) - (N+1)\rho_2{}^N (1-\rho_2)}{[1-\rho_2]^2}\right] \quad (19)$$

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

$$= \sum_{n=0}^{N} nP_n - \sum_{n=0}^{N} P_n = E(web_{disp}) - (1-P_0) \quad (20)$$

but using Little formulae in our model we have $\mathbf{E(disp.queue)} = E(web_{disp}) - \frac{\lambda_d}{\mu_1}$. This is only true when the mean arrival rate is λ_d as assumed by (Nan et al. 2011)(Xiong & Perros 2009). However, $1-P_0 < \frac{\lambda_{eff}}{\mu_1}$ because the mean arrival rate is λ_d when there is vacancy in the queue and 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

$$\mathbf{E(disp.queue)} = \mathbf{E(webdisp)} - \frac{\lambda_{eff}}{\mu_1} \quad (21)$$

This also apply to database queue which is then written as

$$\mathbf{E(dbase.queue)} = E(web_{dbase}) - \frac{1\lambda_{eff}}{\mu_2} \quad (22)$$

The second issue is the re-engineering process which is the calculation of the average waiting time in both the queue and system of the dispatcher and the database where most authors like [11] multiply $\lambda_{eff}{}^{-1}$ by $\mathbf{E(webdisp)}$ as the waiting time in the dispatcher system or $\lambda_{eff}{}^{r-1}$ by $\mathbf{E(disp.queue)}$ as the waiting time in the dispatcher queue.
We base our waiting time both in dispatcher system $(\mathbf{Ws_{disp}})$ and the queue $(\mathbf{Wq_{disp}})$ as

$$\mathbf{Ws_{disp}} = \frac{\mathbf{E(LS_{disp})}}{\lambda_{eff}} * \mathbf{Ex_{visitdisp}} \quad (23)$$

$$\mathbf{Wq_{ddisp}} = \left(\mathbf{Ws_{disp}} - \frac{1}{\mu_1}\right) * \mathbf{Ex_{visitdisp}} \quad (24)$$

$$\mathbf{Wq_{dbase}} = \left(\mathbf{Ws_{dibase}} - \frac{1}{\mu_2}\right) * \mathbf{Ex_{visitdbase}} \quad (25)$$

$$\mathbf{Ws_{dbase}} = \frac{\mathbf{E(LS_{disp})}}{\lambda_{eff}} \mathbf{Ex_{visitdbase}} \quad (26)$$

Where $\mathbf{Ex_{visitdisp}}$ represents the number of visit(s) to the dispatcher which is given as $\mathbf{Ex_{visit\ disp}} = \frac{1}{1-\lambda_{eff}}$ and that of the database as $\mathbf{Ex_{visit\ dbase}} = \frac{1}{1-\lambda_{eff}}$

B. Modelling the web service stations

As noted earlier, we modeled our each web queue station as M/M/c/k with equal service distribution $k_i\lambda_d$ as shown in 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 $k_i\ \lambda_d$ with c servers have total service rate of 3μ. Therefore, for each web queue service station the mean arrival rate is given by

$$k_i\lambda_{\text{eff}_n} = \begin{cases} \mathrm{k_i}\ \lambda_d & \text{for } n = 0,1,2\ldots.,N-1 \\ 0 & \text{for } n = N, N+1,\ldots\ldots \end{cases} \quad (27)$$

and

$$\mu_n = \begin{cases} n\mu & \text{for } n = 0,1,2\ldots.,c-1 \\ c\mu & \text{for } n = c, c+1,\ldots\ldots \end{cases} \quad (28)$$

where $1 < c < N$. From the difference equation, given the steady- state probabilities P_n and P_0, we have

$$P_n = \frac{\mathrm{k_i}\ \lambda_{d_0}\,\mathrm{k_i}\ \lambda_{d_1},\ldots\ldots,\mathrm{k_i}\ \lambda_{d_{n-1}}}{\mu_1\mu_2\ldots\ldots\ldots\ldots\ldots\mu_n}P_0 \quad (29)$$

$$P_0^{-1} = 1 + \sum_{n=1}^{\infty}\left[\frac{\mathrm{k_i}\ \lambda_{d_0}\ \lambda_{d_1}\ldots\mathrm{k_i}\ \lambda_{d_{n-1}}}{\mu_1\mu_2\ldots\ldots\ldots\mu_n}\right] \quad (30)$$

substituting the value $k_i\lambda_d$ and μ_n

$$P_0 = \left[\sum_{n=0}^{c-1}\frac{1}{n!}\left(\frac{\mathrm{k_i}\ \lambda_d}{\mu}\right)^n + \frac{1}{c!}\left(\frac{\mathrm{k_i}\ \lambda_d}{\mu}\right)^c\sum_{n=c}^{k}\left(\frac{\mathrm{k_i}\ \lambda_d}{\mu}\right)^{n-c}\right]^{-1} \quad (31)$$

and

$$P_n = \begin{cases} \dfrac{1}{n!}\left(\dfrac{\mathrm{k_i}\ \lambda_d}{\mu}\right)^n P_0 & \text{for } n \le c \\ \dfrac{1}{c!\,c^{n-c}}\left(\dfrac{\mathrm{k_i}\ \lambda_d}{\mu}\right)^n P_0 & \text{for } c < n \le k \\ 0 & \text{for } n \le k \end{cases} \quad (32)$$

Therefore the expected number of consumers in the queue of each service station *i* is given by

$$E\,(web\ queue_i) = \sum_{n-c}^{N} n - c\,\frac{1}{c!\,c^{n-c}}\left(\frac{\mathrm{k_i}\ \lambda_d}{\mu}\right)^n P_0 \quad (33)$$

but the server utilization in each web queue service station *i* is $\rho_i = \frac{\mathrm{k_i}\lambda_d}{c\mu_1}$ 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

$$E\,(web\ queue_i) = P_0\,\frac{\mathrm{k_i}\lambda_d}{\mu}\,\frac{\rho_i}{c!\,(1-\rho_i)^2}\left[1-\rho_i^{N-c}-(N-c)(1-\rho_i)\rho_i^{N-c}\right] \quad (34)$$

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

$$E\ (web\ system_i) = E\ (web\ queue_i) + \frac{k_i\lambda_{eff}}{\mu} \qquad (35)$$

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

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

Our web system and queue waiting time are

$$Wsystem_i = \lfloor k_i\lambda_{eff} \rfloor^{-1} * E(web\ system_i) \qquad (36)$$

$$Wqueue_i = \lfloor k_i\lambda_{eff} \rfloor^{-1} * E\ (web\ queue_i) \qquad (37)$$

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

$$Wqueue_{ave} = \frac{1}{j}\sum_{i=0}^{j} Wqueu_i \qquad (38)$$

$$Wsystem_{ave} = \frac{1}{j}\sum_{i=0}^{j} Wsystem_i \qquad (39)$$

Therefore, the total queue waiting time in all the service stations is given as

$$Wt = Wq_{ddisp} + Wq_{dbase} + Wqueue_{ave} \qquad (40)$$

4. ASPIRATION LEVEL MODEL

In the work of some authors, for example (Nan et al. 2011)[Xioming, Hadi], the derived performance measures are minimized subject to various constraints. In some other work (Akingbesote A.O et al. 2013) (Akingbesote et al. n.d.), (Kembe 2012), the decision model is analysed using mathematical formulae linking costs or inputs with waiting time and operating metrics to produce an estimated output. This is formulated as $ETC(x) = EWC(x) + EOC(x)$ where ETC(x) is the Expected total cost per unit time, EOC(x) is the expected cost of operating the cloud e-market servers per unit time and EWC(x) is expected cost of waiting by cloud consumers per unit time. In this research we use the aspiration model based on the consumers' average waiting time in the system (wt) and the percentage of the servers' idleness (S_Idle) in both dispatcher and web queue stations. These two conflicting measures of performance serve as our control mechanism that is used to regulate the service level. This mechanism can be extended depending on the voice of the decision makers. The idea is not to solve for optimal solution but for efficient solution that can be best attained.

These are

$$Wt = Wq_{ddisp} + Wq_{dbase} + Wqueue_{ave} \qquad (40)$$

and

$$S\_Idle = \frac{\sum_{i=1}^{j}\frac{(Wsystem_i - E\ (web\ queue_i))}{c}}{j} * 100$$

$$= \frac{\sum_{i=1}^{j}\left(1 - \frac{k_i\lambda_{eff}}{cu}\right) * 100}{j} \qquad (41)$$

Where j is the number of web queue station in the model and c is the total number of server machines per web queue station. Our problem is therefore to determine the service level of servers (c)

subject to

$$Wt \leq \alpha \quad (42)$$

and

$$S\_Idle\ (Idleserver\ period\ (\%)) \leq \beta \quad (43)$$

Where α could be the acceptable SLA waiting time and β the acceptable percentage of the servers' idleness. This is done by plotting wt and S_idle as a function of the number of used server (c).

A. Numerical Validation and Simulation

First, we validate our mathematical solution with the simulation to ascertain the degree of correction. This is done by setting both the simulation and the analytical parameter to the same value. That is $c = 4, 6, 8, 10, \ldots, 20$ respectively in each of the service stations and λ to a constant value. We use the Wolfram Mathematical 9.0 as the mathematical tool for our validation results and arena 14.5 as the simulator. This simulation was run with replication length of 1000 in 24 hours per day with base time in hours and it was replicated 5 times. The service rate was set to 0.001 for the dispatcher-In and 0.0005 for each of the servers in the web Queue stations and the dispatcher-Out. We use server of low service rate of .0002 for the database server because of its randomness. The result is shown in Figure 2 and the explanation is given under the results and discussion section in section IV.

On the major experiment, we started with 2 server machines in each of the web queue stations and at the end of every experiment we increase the server machine by one and a total of three web queue stations are used. The arrival rate is set to 0.1 sec and the service time is set to 0.1sec in each of the servers used. There is no bulking because we started with six server machines. Each experiment is repeated ten times with a replication length of 100000 for 24 hours per day. The results are analyzed in section IV.

5. RESULTS AND DISCUSION

Table 1, Fig. 2 and 3 provide the results of our percentage of idleness and waiting time distributions based on the number of server used (c). Fig. 2 is the function of idleness percentage over the used server machines. In Table 3 and Fig. 2, the Idle period (in %) increases as the number of used server machines increases. Also in Fig 3 the waiting time reduces as the server machine increases. The result in Fig. 4 reveals the acceptable Aspiration Level 1 and 2 respectively where the constraints α and β are the two level of aspirations specified by the decision makers as shown in equation 3.44 and 3.55. In this context, the decision makers may involve the cloud E-market provider and the consumer. In Fig. 4, it was observed that the condition was set to determine the feasible server range to be used when the SLA on waiting time ($\alpha 1$) is 0.00002693 and that of the server idle period ($\beta 1$) is set to 88%. In this figure, it was observed that the Aspiration service level 1 (ASL1) ranges from 9 to-15 servers. This serves as the feasible region if the SLA is to be obeyed.

Also, the Acceptable Service Level (ASL 2) changes in the range of 12 to 17 server machines when the two conflicting measures were changed from $\alpha 1$ *to* $\alpha 2$, and that of $\beta 1$ *to* $\beta 2$, that is from 0.00002693 to 0.00002 (sec) and from 88% to 91.80% as depicted in Fig. 5.. One noticeable change in these two figures was that the percentage of idleness increases while that of waiting time reduces. The reverse could be the case depending on the given constraints. Fig. depicts the combined graph of Fig. 4 and 5 This explains the gain and loss of the model. In this Figure, it was observed that as the waiting time goes down, the number of server machines increases which will inevitably increase costs. For example, in Figure 3.12, as the waiting time dropped from 0.00002693 to 0,00002, we observed a drop of 0,00000693 (K1) which brings an increase of 2 additional server machines (i.e from 15 to 17).

Also, as the percentage of the idle period increases, the server machine also increases which also increase cost. However, a drop in the percentage of Idle period would reduce the number of servers (k2) which will be at the expense of high cost of consumer's dissatisfaction, such as loss of future business and actual processing costs of complaints. Therefore, striking this balance depends on the given conflicting measures. One major advantage of this model in Cloud E-marketplaces is that, it allows the decision makers to predict the feasible outcome of the event based on the given constraint (α, β). For intance, a provider with 20 server machines having the percentage of idleness of 86% in this context will not accept consumer's request with an SLA of waiting time that is less than 0.000015 ($W_t \leq 0.000015$) because it is outside the feasible region and such provider does not have servers that will meet up.

As ealier mentioned, The research solution is not to determine the optimal solution but an acceptable range for the service level by specifying reasonable limits the provider wishes to reach on the conflicting measure of performance**.** One issue that needs to be discussed even though this did not happen in this experiment is when the two conditions can not be satisfied simoultanously. In that case, one or both must be relaxed before a feasible range can be attained.

6. CONCLUSION

One major variable cost that determines cloud profit is the server machines used in the data centers. A low level of servicer machines reduces cost but increase consumers' waiting time which may lead to high costs of consumer's dissatisfaction, such as loss of future business and actual processing costs of complaints. Also, a higher level of servicers will cost more to an E-cloud provider and will result in lower profit or even a loss. Therefore, striking a balance between the service level and some performance measures is a challenge. One of the decision models to tackle this challenge is the use of cost model approach. The viability of this model depends on how well the cost parameters can be estimated which indeed are difficult. We approach this solution using the aspiration level model to achieve an efficient solution that can determine an acceptable range for the service level. Two conflicting performance measures are used for the experiment. (w_s *and* X) and an acceptable service level is attained based on the given constraints.

It is hope that this model apart from the determination of acceptable range of service level, will be useful to predict the aspiration feasible region based on the given constraints.

ACKNOWLEDGMENT

This work is based on the research supported in part by the National Research Foundation of South Africa-Grant UID: TP11062500001 (2012-2014). The authors also acknowledge funds received from industry partners: Telkom SA Ltd, Huawei Technologies SA (Pty) Ltd and Dynatech Information Systems, South Africa in support of this research.

REFERENCES

1. Akingbesote, A.. et al., Determination of optimal service level in cloud e-marketplaces based on service offering delay. , pp.283–288. Available at: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=6914190 [Accessed October 7, 2014].
2. Akingbesote A.O et al., 2013. The Trade-off between consumer's satisfaction and resource service level by e-market providers in e-market places. *in International Conference on Information Technology (ICIT)*. Available at: http://connection.ebscohost.com/c/articles/95511261/modeling-cloud-e-marketplaces-cost-minimization-using-queuing-model.
3. Alvi, F.A., Choudary, B.S. & Jaferry, N., 2012. A review on cloud computing security issues & challenges. In *International conference on Mobility for life, Technology, telecommunication and problem based learning (TTPBL),*.
4. Carlos, C. et al., 2013. Management Accounting: Information for Managing and Creating Value: *9780077116903: Amazon.com: Books*. Available at: http://www.amazon.com/Management-Accounting-Information-Managing-Creating/dp/0077116909 [Accessed July 25, 2014].
5. Chiang, Y. & Ouyang, Y., 2014. Profit Optimization in SLA-Aware Cloud Services with a Finite Capacity Queuing Model. , 2014.
6. Ferretti, S. et al., 2010. QoS;Aware Clouds. *2010 IEEE 3rd International Conference on Cloud Computing*, pp.321–328. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5557978 [Accessed September 9, 2014].
7. Goudarzi, H. & Pedram, M., 2011. Maximizing Profit in Cloud Computing System via Resource Allocation. *2011 31st International Conference on Distributed Computing Systems Workshops*, pp.1–6. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5961382.
8. Gross, D. & John, F., 1985. Fundamentals of Queueing Theory. *Wiley Series in Probability and Mathematical Statistics 2nd Edition*. Available at: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-1118211642.html [Accessed October 2, 2014].
9. Guo, L. et al., 2014. Dynamic Performance Optimization for Cloud Computing Using M/M/m Queueing System. *Journal of Applied Mathematics*, 2014, pp.1–8. Available at: http://www.hindawi.com/journals/jam/2014/756592/.
10. Kahneman, D. & Tversky, A., 2007. Prospect Theory: An Analysis of Decision under Risk. , 47(2), pp.263–292.
11. Kembe, 2012. A Study of Waiting And Service Costs of A Multi- Server Queuing Model In A Specialist Hospital. , 1(8), pp.19–23.
12. Mustafa, F. & McCluskey, T.L., 2009. Dynamic Web Service Composition. In *2009 International Conference on Computer Engineering and Technology*. IEEE, pp. 463–467. Available at: http://dl.acm.org/citation.cfm?id=1510526.1511085 [Accessed July 29, 2014].
13. Nagendram, S., 2011. Efficient Resource Scheduling in Data Centers using MRIS. *(IJCSE)(IJCSE)*, 2(5), pp.764–769.

14. Nan, X., He, Y. & Guan, L., 2011. Optimal resource allocation for multimedia cloud based on queuing model. *2011 IEEE 13th International Workshop on Multimedia Signal Processing*, pp.1–6. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6093813.
15. Pakbaznia, E. & Pedram, M., 2009. Minimizing data center cooling and server power costs. In *Proceedings of the 14th ACM/IEEE international symposium on Low power electronics and design - ISLPED '09*. New York, New York, USA: ACM Press, p. 145. Available at: http://dl.acm.org/citation.cfm?id=1594233.1594268 [Accessed July 29, 2014].
16. Rosenfeld, A. & Kraus, S., Modeling Agents Based on Aspiration Adaptation Theory. *JAAMAS : http://www.icons.umd.edu/*.
17. Sun, W. et al., 2013. Privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. In *Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security - ASIA CCS '13*. New York, New York, USA: ACM Press, p. 71. Available at: http://dl.acm.org/citation.cfm?id=2484313.2484322 [Accessed July 26, 2014].
18. Sundarapandian, 2009. *Probability, Statistics and Queuing Theory*, PHI Learning Pvt. Ltd. Available at: http://books.google.co.za/books/about/Probability_Statistics_and_Queuing_Theor.html?id=9oUS6BBJkCYC&pgis=1 [Accessed October 7, 2014].
19. Taha, A., Operations Research an Introduction: Hamdy A. Taha: 9788120322356: Amazon.com: Books. Available at: http://www.amazon.com/Operations-Research-Introduction-Hamdy-Taha/dp/8120322355/ref=sr_1_4?s=books&ie=UTF8&qid=1410515660&sr=1-4 [Accessed September 12, 2014].
20. Vinothina, V., Lecturer, S. & Sridaran, R., 2012. A Survey on Resource Allocation Strategies in Cloud Computing. , 3(6), pp.97–104.
21. Wang, J., eRAID: A Queueing Model Based Energy Saving Policy. *14th IEEE International Symposium on Modeling, Analysis, and Simulation*, (1), pp.77–86. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1698539.
22. Wu, L., Garg, S.K. & Buyya, R., 2011. SLA-Based Resource Allocation for Software as a Service Provider (SaaS) in Cloud Computing Environments. *2011 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp.195–204. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5948610 [Accessed July 14, 2014].
23. Xiong, K. & Perros, H., 2009. Service Performance and Analysis in Cloud Computing. *2009 Congress on Services - I*, pp.693–700. Available at: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5190711.

FIGURES AND TABLES

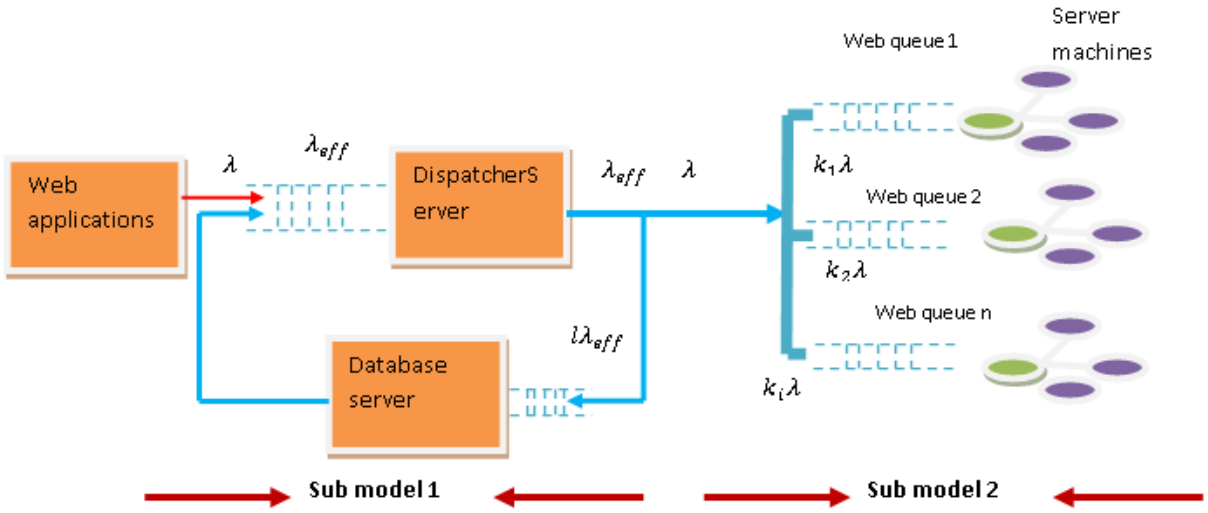

Fig. 1: Proposed Model of e-cloud market

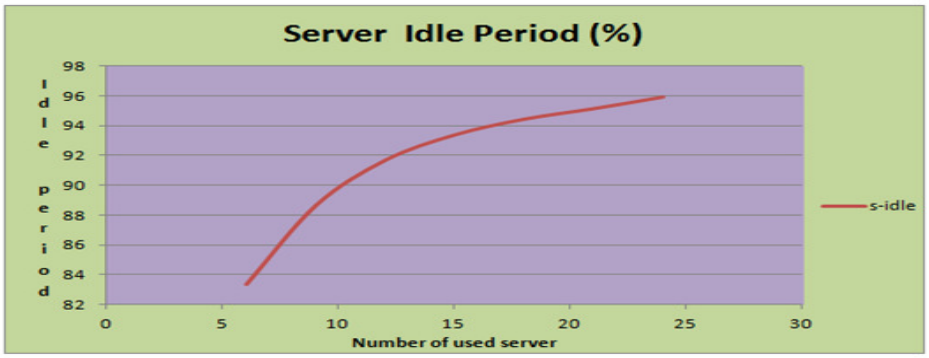

Fig. 2: Idle Period- - Service Level

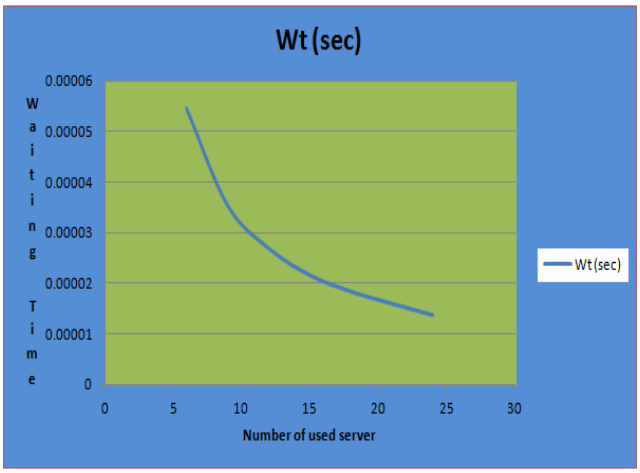

Fig. 3: Waiting time - Service Level

No of used server	Wt (sec)	S-idle (%)	Server Utilization
6	0.00005459	83.31	0.1669
9	0.0000354	88.62	0.1138
12	0.00002693	91.662626	0.0833737
15	0.00002162	93.363636	0.0663636
18	0.00001838	94.422222	0.0557777
21	0.00001598	95.132323	0.0486767
24	0.00001369	95.913131	0.0408686

Table 1: Waiting Time and idle period distributions

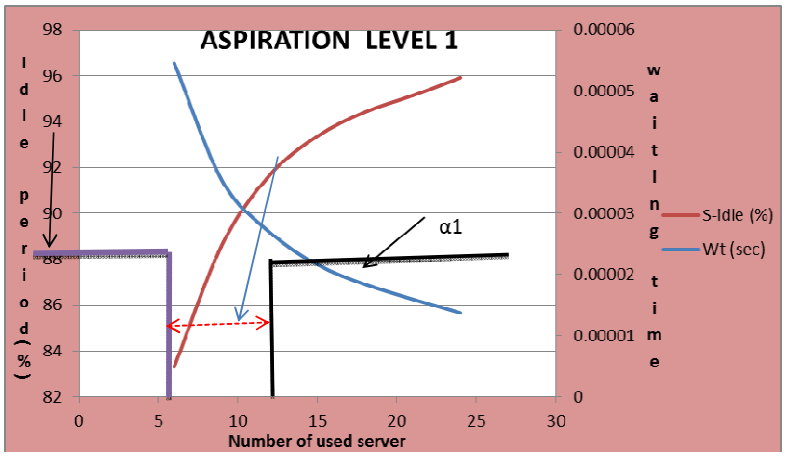

Fig. 4: Aspiration Level 1 (ASL 1)

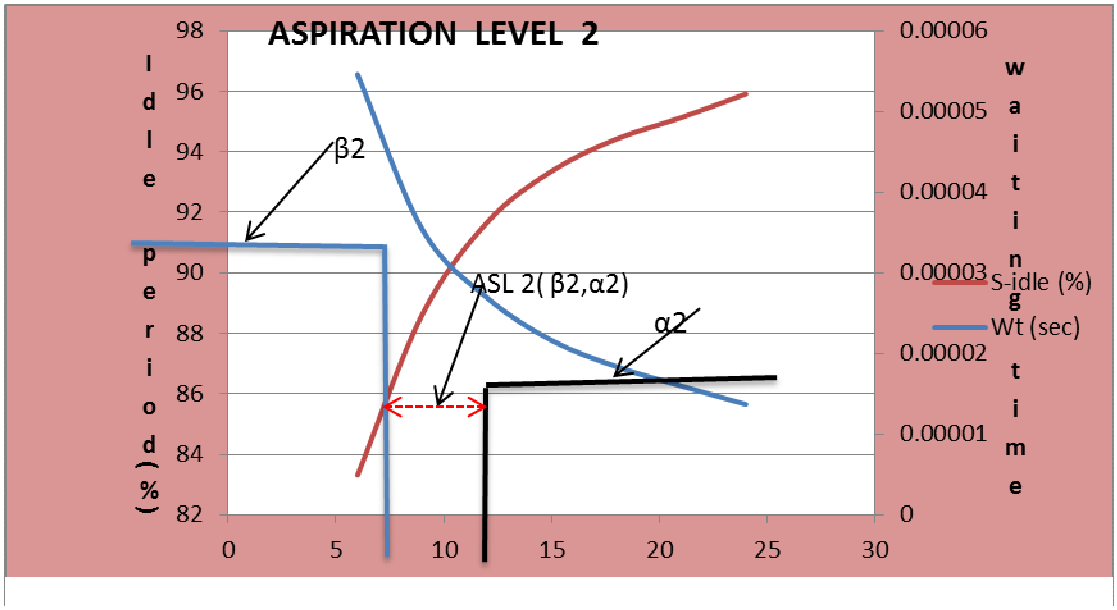

Fig. 5 Aspiration Level 2 (ASL 2)

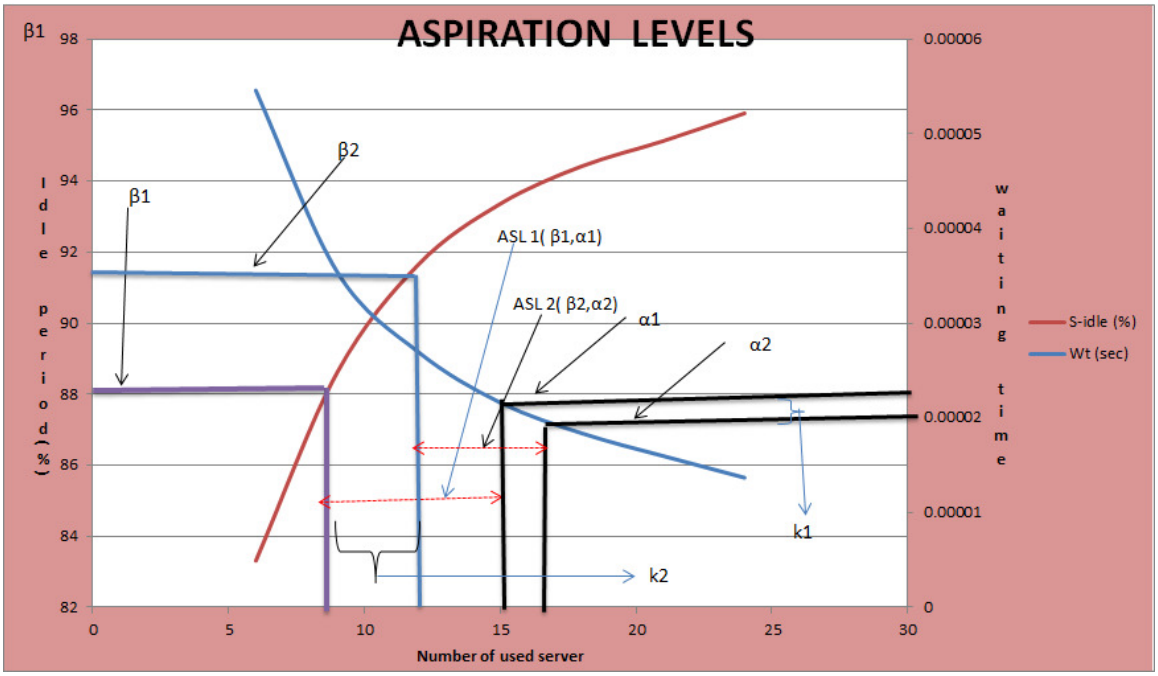

Fig. 6: Combined Aspiration levels