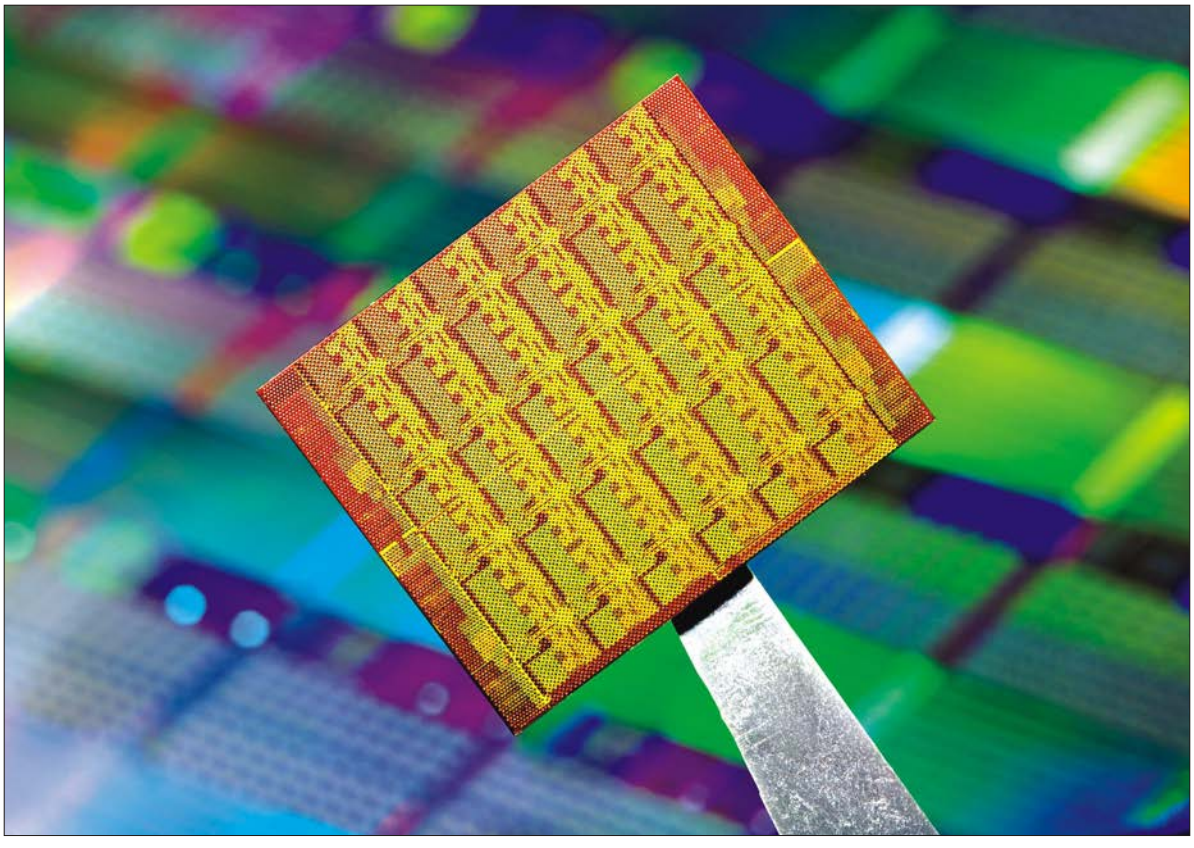

Australasian Journal of

Computer Science

ISSN: 2251-3221

science alert

http://scialert.net/ajcs

OPEN ACCESS Australasian Journal of Computer Science

ISSN 2251-3221
DOI: 10.3923/aujcs.2017.1.16

Research Article
Revenue Maximization Based on Slowdown in Cloud Computing Environments

[1]Michael Okopa, [2]Didas Turatsinze, [3]Tonny Bulega and [1]Jowalie Wampande

[1]Faculty of Science and Technology, Cavendish University Uganda, Kampala, Uganda
[2]Kigali Institute of Education, Kigali, Rwanda
[3]College of Computing and Information Sciences, Makerere University, Kampala, Uganda

Abstract

Background and Objective: Previous pricing mechanisms have been based on response time. The challenge with response time is that it only focuses on the time when a request terminates and does not focus on the size of the request, thus response time tends to be representative of the performance of just a few big requests and not all the requests since they count the most in the mean. On the other hand, slowdown measures the responsiveness of the system with respect to the length of the request that is, requests are completed within the time proportional to request demand. The main objective of this study is to maximize revenue using resource allocation in cloud computing environments based on mean slowdown and instant slowdown customer-oriented pricing mechanisms. **Methodology:** To overcome the challenge of pricing based on response time, two customer-oriented pricing mechanisms Mean Slowdown (MS) and Instant Slowdown (IS) are proposed, in which the customers are charged according to achieved service performance in terms of slowdown. Analytical models of pricing mechanisms based on slowdown are developed for cloud computing under First Come First Served and Processor Sharing scheduling policies. Lagrange multiplier composite functions are then differentiated and equated to zero to determine the number of se`rvers that give maximum revenue. **Results:** The numerical results obtained from the derived models show that revenue generated under slowdown pricing mechanisms are higher than revenue generated under response time pricing mechanisms. It is further observed that processor sharing policy generally generates more revenue than first come first served scheduling policy especially when there are more servers. **Conclusion:** It is concluded that pricing mechanisms based on slowdown can generate more revenue for the service provider than pricing mechanism based on response time.

Key words: Instant slowdown, mean slowdown, mean response time, revenue, processor sharing policy

Citation: Michael Okopa, Didas Turatsinze, Tonny Bulega and Jowalie Wampande, 2017. Revenue maximization based on slowdown in cloud computing environments. Australasian J. Comp. Sci., 4: 1-16.

Corresponding Author: Michael Okopa, Faculty of Science and Technology, Cavendish University Uganda, P.O. Box 33145 Kampala, Uganda

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Cloud computing represents the delivery of computing as a service. In this case, resources such as software, information and devices are provided to end-users as a metered service over the internet. To date, there is no definition that is agreed upon in most quarters. According to National Institute of Standards and Technology (NIST)[1], cloud computing can be defined as "The management of resources, applications and information as services over the cloud (internet) on demand." Cloud computing is a model for enabling convenient and on demand network access to a shared group of computing resources that can be rapidly released with minimal management effort or service provider interaction.

The cloud makes it possible for one to access information from anywhere at any time[2,3]. While a traditional computer setup requires one to be in the same location as the data storage device, the cloud removes the need for one to be in the same physical location as the hardware that stores the data.

The business model based on service level agreements (SLAs) play a crucial role in cloud paradigm. The SLA provides mechanisms and tools that allow service providers and end users to express their requirements and constraints such as mean response time, mean slowdown and price scheme. The mean response time is the total amount of time a request spends in both the queue and in service[4]. Mean slowdown is the ratio of mean response time to the size of the requests. Pricing scheme is the process of determining what a service provider will receive from an end user in exchange for their services. The SLAs facilitate the transactions between customers and service providers by providing a platform for consumers to indicate their required service level or quality of service (QoS)[5]. The SLA normally specifies a common understanding about responsibilities, guarantees, warranties, performance levels in terms of availability, response time etc.[6].

As cloud computing becomes more and more popular, understanding the economics of cloud computing becomes critically important. To maximize the profit, a service provider should understand both service charges and business costs and how they are determined by the characteristics of the applications and the configuration.

Yeo *et al.*[6], described the difference between fixed and variable prices. Fixed prices were easier to understand and more straightforward for users. However, fixed pricing could not be fair to all users because not all users had the same needs. Their study proposed charging variable prices with advanced reservation. Charging variable pricing with advanced reservation would let users know the exact expenses that are computed at the time of reservation even though they were based on variable prices.

Mihailescu *et al.*[7], the authors presented a dynamic pricing scheme which improves the efficiency of batch resource trading in federated cloud environments. In their scheme, the whole cloud system is considered as a uniformed resource market where resource supply and demand can be balanced by using macro-economic equivalence theory. Unfortunately, the scheme relies on market self to automatically obtain equivalent price, making it low-efficient compared with the opening feature of cloud platform.

Zhu *et al.*[8] proposes an allocation strategy of server resources among customers to minimize the mean response time. However, this study does not consider the economic model. In a similar study Mazzuco[9], proposed two strategies for resource allocation, Heuristic and Greedy. Although, Greedy strategy is optimal, it often costs long execution time. Heuristic is simple but its validity is affected by the environment parameters.

In an effort to maximize revenue, Feng *et al.*[5] modeled revenue maximization in cloud computing using an M/M/1/FIFO queue system for a single virtual machine. First in first out (FIFO) is normally used as a base line for temporal fairness, where it is fair to serve a job in the order in which it arrives, such scenarios are found in e-commerce (that is, an item gets sold to the person who first requests for it), databases and other applications where data consistency is important[10]. The authors proposed two customer-oriented pricing mechanisms, mean response time (MRT) and instant response time (IRT), in which the customers are charged according to achieved service performance in terms of mean response time. However, mean response time tends to be representative of the performance of just a few big requests since they count the most in the mean because their response times tend to be highest[10]. In other words an improvement in mean response time could imply the performance of a few big requests have improved. The expression for mean revenue in terms of Mean Response Time (MRT) is given by Feng *et al.*[5] as:

$$G_i = \lambda_i b_i \left[1 - \frac{1}{(n_i\mu_i - \lambda_i)R_i}\right] \quad (1)$$

Where:

x_i = Request size at instance i
n_i = Number of servers at service instance i
μ_i = Service rate at service instance i
λ_i = Arrival rate at service instance i
b_i = Price constant for service instance i

On the other hand, the expression for overall mean revenue in terms of Instant Response Time (IRT) is given by Feng *et al.* as [5]:

$$G_i = \lambda_i b_i (1 - e^{x_i(\lambda_i - n_i\mu_i)R_i}) \quad (2)$$

Since resource allocation strategies have an impact on the service performance, a fundamental problem faced by any cloud service provider is how to maximize revenue by allocating resources dynamically among the service instances based on SLA and measurable performance indices.

The main objective of this study was to maximize revenue using resource allocation in cloud computing environments based on mean slowdown. This has been achieved as the model based on mean slowdown is observed to generate more revenue than the model based on mean response time.

MATERIALS AND METHODS

This study employed queueing theory to model MS and IS pricing schemes. Among existing analytical tools, queuing theory has been proved to be a useful tool to deal with queuing problems in communication networks[5,11]. Queuing theory is a primary tool for studying mean response time (MRT) and instant response time (IRT)[5] and other performance metrics[8,9]. Resource allocation model in terms of mean slowdown and instant slowdown are considered.

Resource allocation model in terms of mean slowdown: Mean slowdown is a commonly used metric to evaluate the service performance[9,11]. Mean slowdown of requests is modeled using M/M/n_i/FCFS and M/M/n_i/PS queuing systems. For a time-slotted system, it is important to calculate the mean slowdown of every time slot independently because arrival rate of requests vary over time. The billing under this model is such that each mean slowdown has its own rate. Every service instance has a different rate, which is determined by the customer's actual requirement. This pricing model is also called service demand driven model[8].

The billing under this model is such that each mean slowdown has its own rate. Every service instance has a different rate, which is determined by the customer's actual requirement.

Let F denote an offset factor of actual mean slowdown to benchmark defined F as [5]:

$$F = \frac{r}{xs} = \frac{(r/x)}{s}$$

where, (r/x) is the measured mean slowdown during a time slot, s represents a benchmark of mean slowdown defined in the SLA while r is the mean response time and x is the job size.

Every service instance has different s, which is determined by customer's actual requirement. For example, in terms of response time, the recommended response time for transactions in e-commerce is 2-4 sec Feng *et al.*[5]. The pricing mechanism can be formulated as:

$$B = b(1 - F) \text{ and } F = b\left(1 - \left(\frac{r}{xs}\right)\right) \quad (3)$$

where, B is the price of each service provision and b is the price constant.

Resource allocation model in terms of instant slowdown (IS): The pricing model in terms of mean slowdown may work well when the measurements are evenly distributed over a narrow range. However, mean slowdown is not meaningful as a performance metric when the mean slowdown varies a little over a large range. This is the motivation behind proposing another pricing model in terms of instant slowdown (IS). A request under IS is charged according to the measured slowdown. The billing under this model is determined by the number of service provisions with mean slowdown less or equal to a given threshold. The same rate is charged for a particular interval.

Given certain customer arrival patterns and service requirements, the order of service is the most important point affecting the performance of a service management facility[12]. Specifically, the M/M/n_i/FIFO and M/M/n_i/PS queuing systems were used, where the first M represents Poisson arrival with mean arrival rate (λ) per request with exponentially distributed inter arrival times. Poisson distribution best models random arrivals into systems. Poisson probability distribution is given as[4]:

$$p(x) = \frac{\lambda^x e^{-\lambda}}{x!};\ x = 0, 1, 2, \ldots$$

Where:

x = Number of arrivals in a specific period of time

λ = Average, or expected number of arrivals for the specific period of time, e = 2.71828

The second M represents exponential service time and the 1 represents the number of servers. Each service instance, a virtual machine associated with a user, is modeled as

$M/M/n_i/PS$ queue and later extended to multiple servers to give a service rate of $n\mu_i$. The exponential probability distribution is given in as[4]:

$$f(t) = \mu e^{-\mu t}; \quad t \geq 0 \tag{4}$$

Where:

t = Service time (expressed in number of time periods)

μ = Average or expected number of units that the service facility can handle in a specific period of time, processor sharing (PS) is the scheduling policy used to give service in this study

Define service intensity, ρ as the ratio of arrival rate to the service rate, $\rho = \lambda/\mu$.

The FIFO policy is used in this study because FIFO serves jobs in the order in which it arrives, such scenarios are found in e-commerce (that is, an item gets sold to the person who first requests for it), databases and other applications where data consistency is important[10]. On the other hand processor sharing (PS) is used as a base line for proportional fairness where it is fair for the response time of jobs to be proportional to the job size, such scenarios are found in web servers and routers to ensure no class of jobs is starved[10].

Assume that the Cloud data center is composed of N homogenous servers. The servers are grouped into clusters dynamically and each server can only join one cluster at a time. Each cluster is built from a number of homogeneous machines. Every service instance is mapped to a server cluster. Each cluster is virtualized as a single machine. A service provider signs long term SLAs with m customers. The dispatcher assigns the incoming requests to individual servers in the cluster i.e., every service instance is allocated to $n_1, n_2 ... 2_m$ servers to provide services. The dispatcher can also determine the scheduling policy at each server. Also assume that the requests from any service instance arrives to the system with Poisson distribution with average arrival rate λ and the service times by one server follows a negative exponential distribution with average service rate 1/μ (the number of requests processed per unit time). The service rate of the virtual machine with n_i servers is then given by 1/nμ. Each service instance, a virtual machine associated with a user, can be modeled as an $M/M/n_i/FCFS$ or $M/M/n_i/PS$ queue system. The billing under this model is determined by the number of service provisions with mean slowdown within a benchmark, S. Next, the expression for revenue in terms of mean slowdown for FCFS and PS policies are derived.

Derivation of expression for revenue in terms of mean slowdown for FCFS policy: The average response time for an $M/M/n_i/FCFS$ queue system is given as [4]:

$$\frac{1}{(\mu - \lambda)} \tag{5}$$

Basing on Eq. 5, the mean slowdown S_i of service instance i at the steady state is then given by:

$$S_i = \frac{1}{x_i\,(n_i\mu_i - \lambda_i)}$$

Where:

x_i = Request size at instance i

n_i = Number of servers at service instance i

μ_i = Service rate at service instance i

λ_i = Arrival rate at service instance i

The service performance level F_i is then given by:

$$F_i = \frac{1}{x_i\,(n_i\mu_i - \lambda_i)\,S_i}$$

According to Eq. 3, the mean revenue g_i brought by a service provision is:

$$g_i = b_i\left[1 - \frac{1}{x_i\,(n_i\mu_i - \lambda_i)\,S_i}\right]$$

The overall revenue during a time slot from service instance i is:

$$G_i = \lambda_i g_i = \lambda_i b_i\left[1 - \frac{1}{x_i\,(n_i\mu_i - \lambda_i)\,S_i}\right] \tag{6}$$

The optimization problem can then be formulated as:

Maximize:

$$\sum_{i=1}^{m} \lambda_i b_i\left[1 - \frac{1}{x_i\,(n_i\mu_i - \lambda_i)\,S_i}\right]$$

Such that:

$$\sum_{i=1}^{m} n_i = N \tag{7}$$

The problem in Eq. 7 is resolved using lagrange multiplier by constructing lagrange composite function. To maximize or minimize the function f(x, y) which is subject to the constraint g (x, y) = k, first create the lagrange function. This function is composed of the function to be optimized combined with the constraint function in the following way:

$$L(x, y) = f(x, y)+\lambda\,[g(x, y)-k] \tag{8}$$

The partial derivative with respect to each variable x, y and the lagrange multiplier λ of the function is found. Each of the partial derivatives are equated to zero:

$$\sum_{i=1}^{m}\lambda_i b_i\left[1-\frac{1}{x_i\,(n_i\mu_i-\lambda_i)\,S_i}\right]$$

Therefore, given the optimization problem subject to the constraint given in Eq. 7, a similar argument as in Eq. 8 is used to obtain the following lagrange function:

$$L(n_i)=\sum_{i=1}^{m}\lambda_i b_i\left(1-\frac{1}{x_i\,(n_i\mu_i-\lambda_i)\,S_i}\right)+\bar{\lambda}\left(N-\sum_{i=1}^{m}n_i\right) \tag{9}$$

where, $\bar{\lambda}$ is a constant of lagrange multiplier. To determine the maximum number of servers used for each service instance, differentiate Eq. 9 with respect to n_i and equate to zero:

For i = 0,1, 2,.....,m:

$$\frac{dL(n_i)}{dn_i}=0$$

$$\frac{dL(n_i)}{dn_i}=\lambda_i b_i\left(\frac{x_i\mu_i S_i}{x_i^2\,(n_i\mu_i-\lambda_i)^2 S_i^2}\right)-\bar{\lambda}=0$$

$$\frac{\lambda_i b_i}{S_i}\left(\frac{\mu_i}{x_i\,(n_i\mu_i-\lambda_i)^2}\right)-\bar{\lambda}=0$$

$$\frac{\mu_i}{x_i\,(n_i\mu_i-\lambda_i)^2}=\frac{\bar{\lambda}S_i}{\lambda_i b_i}$$

$$n_i\mu_i=\lambda_i+\sqrt{\left(\frac{1}{\bar{\lambda}}\frac{\mu_i\lambda_i b_i}{S_i x_i}\right)} \tag{10}$$

Simplifying Eq. 10, it is obtained.

Hence, it is obtained:

$$n_i=\rho_i+\sqrt{\left(\frac{1}{\bar{\lambda}}\frac{\rho_i q_i}{x_i}\right)} \tag{11}$$

Substituting Eq. 11 into the constraint of optimization problem in Eq. 7, it is obtained:

$$N=\sum_{i=1}^{m}\rho_i+\sqrt{\frac{1}{\bar{\lambda}}}\bullet\sum_{i=1}^{m}\sqrt{\left(\frac{\rho_i q_i}{x_i}\right)} \tag{12}$$

$$\sqrt{\frac{1}{\bar{\lambda}}}=\frac{N-\sum_{i=1}^{m}\rho_i}{\sum_{i=1}^{m}\rho_i\sqrt{\left(\frac{\rho_i q_i}{x_i}\right)}} \tag{13}$$

Substituting $\sqrt{\frac{1}{\bar{\lambda}}}$ from Eq. 13 into Eq. 11, it is obtained:

$$n_i=\rho_i+\left(\frac{N-\sum_{i=1}^{m}\rho_i}{\sum_{i=1}^{m}\sqrt{\frac{\rho_i q_i}{x_i}}}\right)\bullet\sqrt{\frac{\rho_i q_i}{x_i}} \tag{14}$$

Equation 14 is valid only when the request arrival rate of each service instance is less than service processing rate. Otherwise, the queue length will be infinitely long. That is, $\lambda_i<n_i\mu_i$ or:

$$\rho_i<n_i \tag{15}$$

Therefore, the service allocation strategy guarantees that the mean slowdown is less than S_i, that is:

$$\frac{1}{x_i\,(n_i\mu_i-\lambda_i)}<S_i$$

Which on simplification gives:

$$n_i>\frac{1}{S_i\mu_i x_i}+\rho_i \tag{16}$$

Equation 15 and 16 offer the lower bound of assigned resources for each service instance.

Derivation of expression for revenue in terms of instant slowdown for FCFS policy: The response time probability distribution is:

$$w(t)=(\mu-\lambda)e^{(\lambda-\mu)t} \tag{17}$$

From Eq. 17, it follows that the sojourn time distribution is given by:

$$w'(t)=x(\mu-\lambda)e^{x(\lambda-\mu)t} \tag{18}$$

where, x is the job size. If service instance i is allocated to n_i servers, then the mean revenue brought by a service provision is given:

$$g_i = \int_0^{S_i} b_i w'(t)\,dt = \int_0^{S_i} b_i x_i\ (\mu_i - \lambda_i)\, e^{x_i(\lambda_i - n_i\mu_i)t}\,dt$$
$$g_i = b_i x_i \left(1 - e^{x_i(\lambda_i - n_i\mu_i)S_i}\right)$$

The overall mean revenue from service instance i during a time slot is:

$$G_i = \lambda_i g_i = \lambda_i b_i x_i \left(1 - e^{x_i(\lambda_i - n_i\mu_i)S_i}\right) \quad (19)$$

The optimization problem can be formulated as, maximize:

$$\sum_{i=1}^{m} \lambda_i b_i x_i \left(1 - e^{x_i(\lambda_i - n_i\mu_i)S_i}\right)$$

Subject to:

$$\sum_{i=1}^{m} n_i = N \quad (20)$$

By constructing the lagrange composite function:

$$L(n_i) = \sum_{i=1}^{m} \lambda_i b_i x_i \left(1 - e^{x_i(\lambda_i - n_i\mu_i)S_i}\right) + \bar{\lambda}\left(N - \sum_{i=1}^{m} n_i\right) \quad (21)$$

Where:

$\bar{\lambda}$ is the lagrange multiplier. By differentiating Eq. 21 with respect to n_i:

$$\frac{dL(n_i)}{dn_i} = \lambda_i b_i x_i^2 \mu_i S_i e^{x_i(\lambda_i - n_i\mu_i)S_i} - \bar{\lambda} = 0 \quad (22)$$

$$e^{x_i(\lambda_i - n_i\mu_i)S_i} = \frac{\bar{\lambda}}{\lambda_i b_i x_i^2 \mu_i S_i}$$
$$x_i(\lambda_i - n_i\mu_i)S_i = \ln\bar{\lambda} - \ln(\lambda_i b_i x_i^2 \mu_i S_i)$$
$$\lambda_i - n_i\mu_i = \frac{\ln\bar{\lambda}}{x_i S_i} - \frac{\ln(\lambda_i b_i x_i^2 \mu_i S_i)}{x_i S_i} \quad (23)$$
$$n_i\mu_i = \frac{\ln(\lambda_i b_i x_i^2 \mu_i S_i)}{x_i S_i} - \frac{\ln\bar{\lambda}}{x_i S_i} + \lambda_i$$
$$n_i = \frac{\ln(\lambda_i b_i x_i^2 \mu_i S_i)}{\mu_i S_i x_i} - \frac{\ln\bar{\lambda}}{x_i S_i \mu_i} + \rho_i$$

Substituting n_i in Eq. 20:

$$N = \sum_{i=1}^{m} \frac{\ln(\lambda_i b_i x_i^2 \mu_i S_i)}{\mu_i S_i x_i} - \ln\bar{\lambda} \sum_{i=1}^{m} \frac{1}{\mu_i S_i x_i} + \sum_{i=1}^{m} \rho_i$$
$$\ln\bar{\lambda} = \frac{\sum_{i=1}^{m} \frac{\ln(\lambda_i b_i x_i^2 \mu_i S_i)}{x_i S_i \mu_i} + \sum_{i=1}^{m} \rho_i - N}{\sum_{i=1}^{m} \frac{1}{x_i S_i \mu_i}} \quad (24)$$

Substituting lnλ in Eq. 23, it is obtained:

$$n_i = \frac{\ln(\lambda_i b_i x_i^2 \mu_i S_i)}{\mu_i S_i x_i} - \frac{\sum_{i=1}^{m} \frac{(\lambda_i b_i x_i^2 \mu_i S_i)}{\mu_i S_i x_i} + \sum_{i=1}^{m} \rho_i - N}{x_i S_i \mu_i \sum_{i=1}^{m} \frac{1}{x_i S_i \mu_i}} + \rho_i \quad (25)$$

Equation 25 also holds when arrival rate is less than the service rate of the virtual machine composed of all the assigned servers.

Derivation of expression for revenue in terms of mean response time for PS policy: The average response time for an M/M/n_i/PS queue system is given in as[4]:

$$\frac{\mu x}{(\mu - \lambda)} \quad (26)$$

Therefore, the average response time r_i of service instance i at the steady state is given as:

$$r_i = \frac{n_i \mu_i x}{n_i \mu - \lambda_i}$$

The service performance level F_i is given as:

$$F_i = \frac{n_i \mu_i x}{(n_i \mu - \lambda_i) R_i}$$

According to the pricing mechanism, B = b (1-F), the mean revenue g_i brought by a service provision is:

$$g_i = b_i \left(1 - \frac{n_i \mu_i x}{(n_i \mu_i - \lambda_i) R_i}\right)$$

The overall revenue generated during a time slot from the service instance i is given by:

$$G_i = \lambda_i g_i = \lambda_i b_i \left(1 - \frac{n_i \mu_i x}{(n_i \mu_i - \lambda_i) R_i}\right) \quad (27)$$

Formulating the optimization problem:

$$\max \sum_{i=1}^{m} \lambda_i b_i \left(1 - \frac{n_i \mu_i x}{(n_i \mu_i - \lambda_i) R_i}\right)$$

Subject to:

$$\sum_{i=1}^{m} n_i = N \tag{28}$$

Resolve the above problem using lagrange multiplier method by constructing lagrange composite function:

$$L(n_i) = \sum_{i=1}^{m} \lambda_i b_i \left(1 - \frac{n_i \mu_i x}{(n_i \mu_i - \lambda_i) R_i}\right) + \bar{\lambda} \left(N - \sum_{i=1}^{m} n_i\right)$$

where, $\bar{\lambda}$ is a constant of lagrange multiplier. By differentiating $L(n_i)$ with respect to n_i.

After further simplification, it is obtained:

$$\begin{aligned} &\frac{dL(n_i)}{dn_i} = 0,\ i = 0, 1, 2, 3, ..., m \\ &\frac{\lambda_i b_i}{R_i} \left(\frac{\lambda_i \mu_i x}{(n_i \mu_i - \lambda)^2}\right) - \bar{\lambda} = 0 \\ &\frac{\lambda_i b_i}{R_i} \left(\frac{\lambda_i \mu_i x}{(n_i \mu_i - \lambda)^2}\right) = \bar{\lambda} \\ &n_i = \sqrt{\frac{\rho_i \lambda_i b_i x}{\bar{\lambda} R_i}} + \rho_i \end{aligned} \tag{29}$$

Substituting Eq. 29 into the constraint of the optimization problem, $N = \sum_{i=1}^{m} n_i$, it is obtained:

$$\begin{aligned} &N = \sqrt{\frac{1}{\bar{\lambda}}} \left(\sum_{i=1}^{m} \sqrt{\frac{\rho_i \lambda_i b_i x}{R_i}}\right) + \sum_{i=1}^{m} \rho_i \\ &\sqrt{\frac{1}{\bar{\lambda}}} = \frac{N - \sum_{i=1}^{m} \rho_i}{\left(\sum_{i=1}^{m} \sqrt{\frac{\rho_i \lambda_i b_i x}{R_i}}\right)} \\ &n_i = \frac{N - \sum_{i=1}^{m} \rho_i}{\left(\sum_{i=1}^{m} \sqrt{\frac{\rho_i \lambda_i b_i x}{R_i}}\right)} \bullet \sqrt{\rho_i \lambda_i q_i x} + \rho_i \end{aligned} \tag{30}$$

The number of servers n_i required to optimize revenue is given by Eq. 30.

Derivation of expression for revenue in terms of instant response time for PS policy: The average response time probability distribution of an M/M/n_i/PS system is given as[5]:

$$w(t) = \frac{\mu - \lambda}{\mu x} e^{\left(\frac{\lambda - \mu}{\mu x}\right) t}$$

The mean revenue brought by a service provision with n_i servers is then given by:

$$g_i = \int_0^{R_i} w(t)\, dt = \int_0^{R_i} b_i \frac{\mu_i - \lambda_i}{\mu_i x} e^{\left(\frac{\lambda_i - n_i \mu_i}{n_i \mu_i x}\right) t} dt$$

$$g_i = b_i \left(1 - e^{\left(\frac{\lambda_i - n_i \mu_i}{n_i \mu_i x}\right) R_i}\right)$$

The overall mean revenue from service instance *i* during a time slot is:

$$G_i = \lambda_i g_i = \lambda_i b_i \left(1 - e^{\left(\frac{\lambda_i - n_i \mu_i}{n_i \mu_i x}\right) R_i}\right) \tag{31}$$

Derivation of expression for revenue in terms of mean slowdown for PS policy: The expression for mean slowdown for PS policy can be deduced by dividing mean response time under PS policy by job size *x* to get:

$$\frac{\mu}{(\mu - \lambda)} \tag{32}$$

The average mean slowdown s_i of service instance i at the steady state is given as:

$$S_i = \frac{n_i \mu_i}{(n_i \mu_i - \lambda_i)}$$

The service performance level F_i is given as:

$$F_i = \frac{n_i \mu_i}{(n_i \mu_i - \lambda_i) S_i}$$

where, S_i is a benchmark mean slowdown for service instance i. According to the pricing mechanism, B = b (1-F), the mean revenue g_i brought by a service provision is:

$$g_i = b_i \left(1 - \frac{n_i \mu_i}{(n_i \mu_i - \lambda_i) S_i}\right)$$

This gives the overall revenue generated during a time slot from the service instance i as:

$$G_i = \lambda_i g_i = \lambda_i b_i \left(1 - \frac{n_i \mu_i}{(n_i \mu_i - \lambda_i) S_i}\right) \quad (33)$$

Derivation of expression for revenue in terms of instant slowdown (IS) for PS policy: The expression for instant slowdown for PS policy can be deduced by dividing mean response time for PS policy by job size *x* to get:

$$\frac{\mu}{(\mu - \lambda)} \quad (34)$$

The corresponding mean slowdown probability distribution of an M/M/n_i/PS system is then given by:

$$w(t) = \frac{(\mu - \lambda)}{\mu} e^{\frac{(\lambda - \mu)}{\mu} t}$$

The mean revenue brought by a service provision with n_i servers is then:

$$g_i = \int_0^{S_i} w(t)\,dt = \int_0^{S_i} b_i \left(\frac{(n_i \mu_i - \lambda_i)}{n_i \mu_i}\right) e^{\frac{(\lambda_i - n_i \mu_i)}{n_i \mu_i} t} dt$$

$$g_i = b_i \left(1 - e^{\frac{(\lambda_i - n_i \mu_i) S_i}{n_i \mu_i}}\right)$$

The overall mean revenue from service instance i during a time slot is given by:

$$G_i = \lambda_i b_i \left(1 - e^{\frac{(\lambda_i - n_i \mu_i) S_i}{n_i \mu_i}}\right) \quad (35)$$

RESULTS

In this study the performance of the derived models are tested. In particular, the variation of revenue with number of servers and arrival rate of packets in the system are analyzed. In each case, the performance using response time and slowdown as performance metrics are compared. The tool used for analysis is MATLAB. Basic mathematical symbols and evaluation parameters used in the analysis are indicated in Table 1 and 2. Evaluation parameters used in the analysis are indicated in Table 2.

Table 1: Basic mathematical symbols used in the analysis

Parameters	Meaning
λ_i	Arrival rate of service requests of each instance
μ	Service rate of each service instance
ρ	Service intensity
m	Number of service instances
n	Variable of assigned servers to an instance
N	Number of all the servers in the resource pool
g	Mean revenue from a service provision
G	Provider's revenue from the cloud provision

Table 2: Evaluation parameters

Parameters	Values
Arrival rate (λ)	2.....30 packets sec^{-1}
Service rate (μ)	10 packets sec^{-1}
Intercept (b)	20/60
Number of servers	20
Number of service instances	20

Comparison of MRT and MS under FCFS policy: This section investigates the variation of revenue with number of servers and arrival rate of packets in the system.

Figure 1 shows a graph of revenue as a function of number of servers for mean response time (MRT) and mean slowdown (MS) pricing mechanisms under FCFS policy. In doing this, Eq. 1 and 6 were used to plot the graph of revenue as a function of number of servers. To investigate the effect of increasing the number of servers, the arrival rate, service rate and size of requests were fixed. It is observed that revenue generally increases with increase in number of servers regardless of the pricing mechanism. This is because as the number of servers increase, the number of tasks completed also increases and hence more revenue is generated. Further, it is observed that more revenue is generated when MS pricing mechanism is used than when MRT pricing mechanism is used. The difference in revenue generated using MRT and MS is more pronounced for low number of servers as compared to high number of servers. For example, when the number of servers is 20, the revenue generated using MRT is approximately $5.5 while the revenue generated using MS is approximately $6.5. On the other hand, when the number of servers is 100, the revenue generated using MRT is $6.5, while the revenue generated using MS is approximately $6.75.

Figure 2 shows the variation of revenue as a function of average arrival rate for mean response time (MRT) and mean slowdown (MS) pricing mechanisms under FCFS policy. In doing this, Eq. 1 and 6 were used to plot the graph of revenue as a function of average arrival rate. To investigate the effect of increasing the arrival rate on revenue, the number of servers, the service rate and size of requests were fixed. It is observed that revenue generally increases with increase in

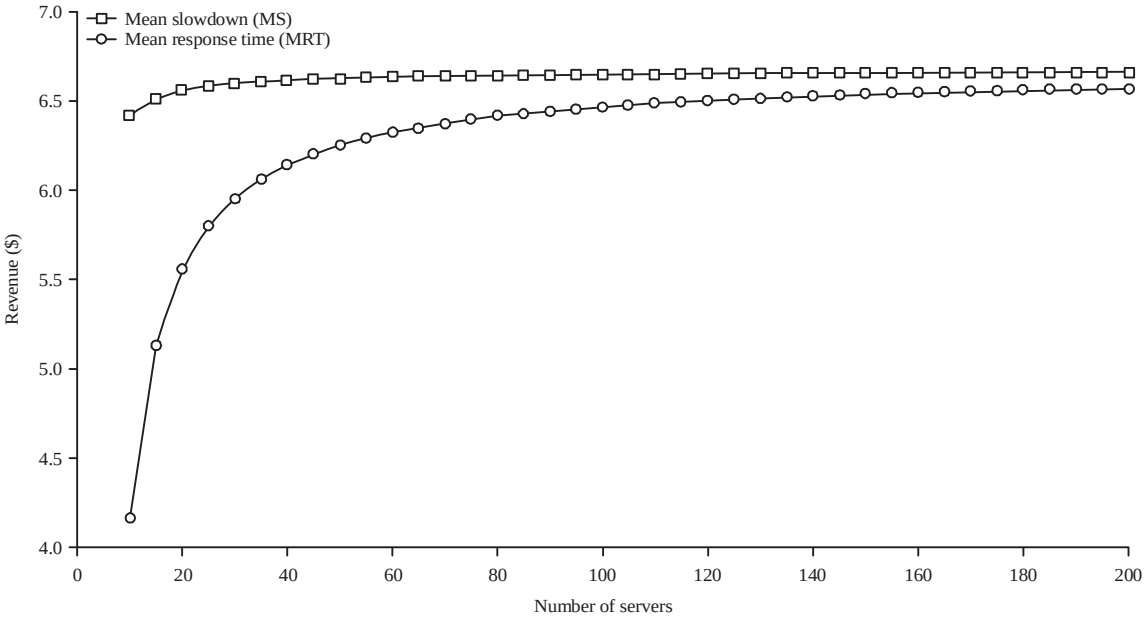

Fig. 1: Variation of revenue with number of servers for MRT and MS under FCFS policy

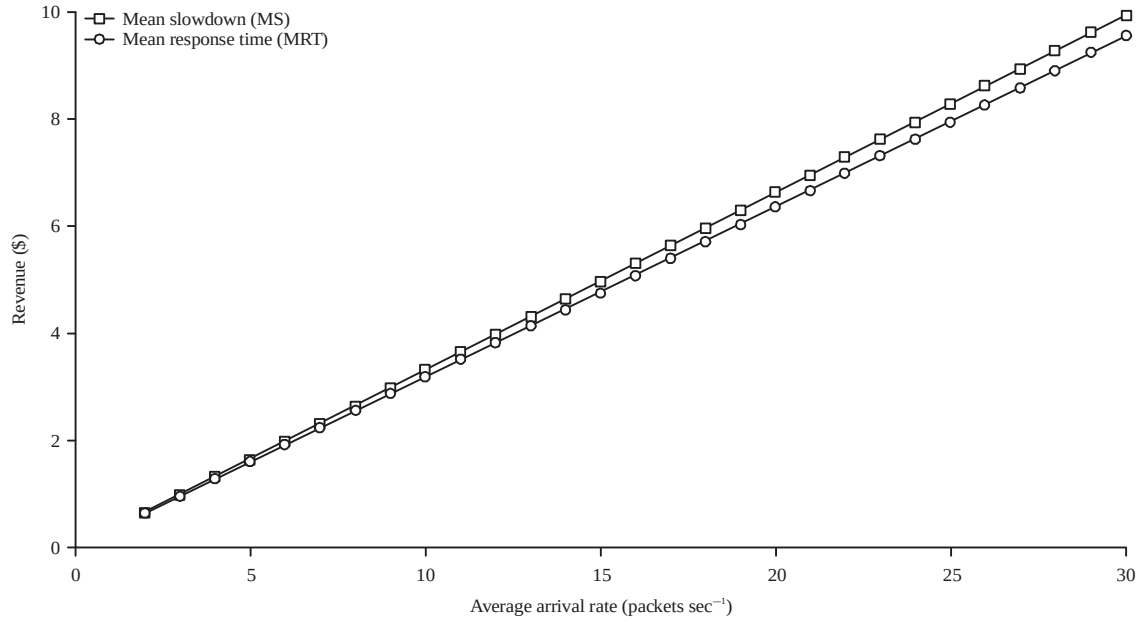

Fig. 2: Variation of revenue with average arrival rate for MRT and MS under FCFS policy

average arrival rate regardless of the pricing mechanism. This is because as the average arrival rate increases, the number of requests served also increases and hence more revenue is generated. Furthermore, it is observed that more revenue is generated when MS pricing mechanism is used than when MRT pricing mechanism is used. For example when the arrival rate is 25 packets sec^{-1}, the revenue generated using MRT is \$8.0 while the revenue generated when MS is used is \$8.2. The difference in revenue generated using MRT and MS is much closer for lower arrival rates and less close as the arrival rate increases.

Comparison of IRT and IS under FCFS policy: This section investigates the variation of revenue with number of servers and arrival rate of packets in the system for IRT and IS charging models under FCFS.

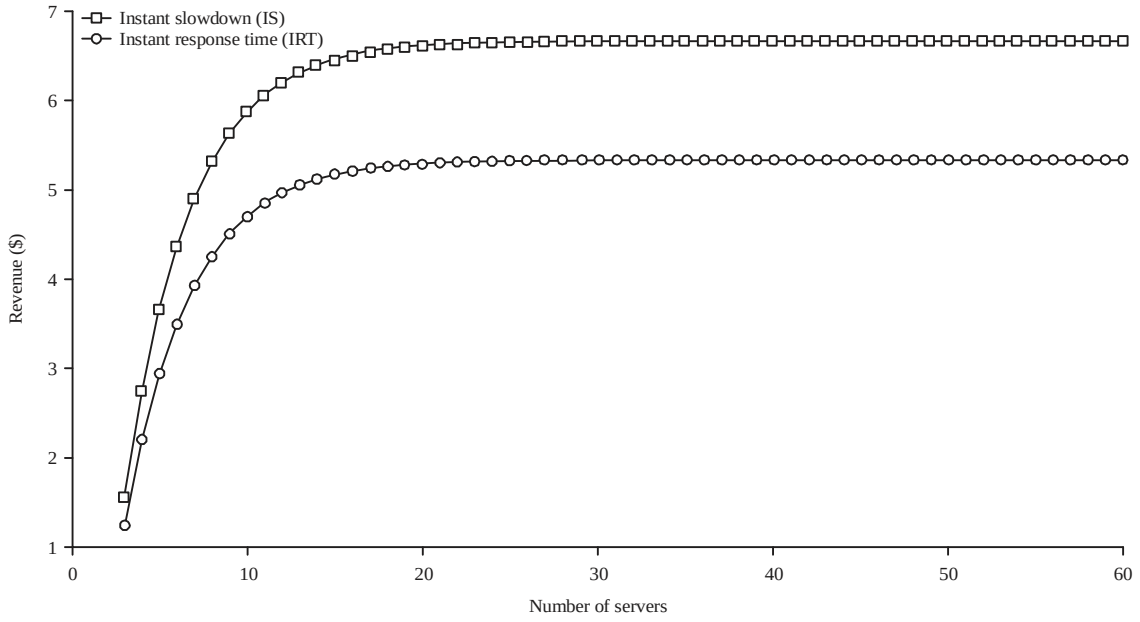

Fig. 3: Variation of revenue with number of servers for IRT and IS under FCFS policy

Figure 3 shows the variation of revenue as a function of number of servers for instant response time (IRT) and instant slowdown (IS) pricing mechanisms under FCFS policy. In doing this, Eq. 2 and 19 were used to plot the graph of revenue as a function of number of servers. To investigate the effect of increasing the number of servers on revenue for the two pricing schemes, fix the arrival rate, the service rate and size of requests. It is observed that revenue generally increases with increase in number of servers regardless of the pricing mechanism. This is because as the number of servers increase, the number of requests also increases and hence more revenue is generated. It is further observed that more revenue is generated when IS pricing mechanism is used and when IRT pricing mechanism is used. The difference in revenue generated using IRT and IS is more pronounced for high number of servers as compared to low number of servers. For example, when the number of servers is 20, the revenue generated using IRT pricing mechanism is approximately $5.2 while the revenue generated using IS pricing mechanism is approximately $6.8. On the other hand, when the number of servers is 10, the revenue generated using IRT pricing mechanism is approximately $4.8, while the revenue generated using IS is approximately $5.9.

Figure 4 shows the variation of revenue as a function of arrival rate for instant response time (IRT) and instant slowdown (IS) pricing mechanisms under FCFS policy. In doing this, Eq. 2 and 19 were used to plot the graph of revenue as a function of arrival rate. To investigate the effect of increasing the arrival rates on revenue for the two pricing schemes, fix the number of servers, the service rate and size of requests. It is observed that revenue generally increases with increase in arrival rate regardless of the pricing mechanism used. This is because as the arrival rate increases, the number of requests into the system also increases and hence more revenue is generated. It is further observed that more revenue is generated when IS pricing mechanism is used and when IRT pricing mechanism is used. For example, when the arrival rate is 8 packets sec^{-1}, the revenue generated using IRT pricing mechanism is approximately $2.0 while the revenue generated using IS pricing mechanism is approximately $3.6. On the other hand, when the arrival rate is 18 packets sec^{-1}, the revenue generated using IRT pricing mechanism is approximately $4.6, while the revenue generated using IS pricing mechanism is approximately $6.0.

Figure 5 shows the variation of revenue as a function of number of servers for mean response time (MRT) and mean slowdown (MS) pricing mechanisms under PS scheduling policy. In doing this, Eq. 27 and 33 were used to plot the graph of revenue as a function of number of servers. To investigate the effect of increasing the number of servers on revenue for the two pricing schemes, fix the arrival rate, the service rate and size of requests. It is observed that revenue generally increases with increase in number of servers regardless of the pricing mechanism used. It is further observed that more revenue is generated when MS pricing mechanism is used and when MRT pricing mechanism is used. For example,

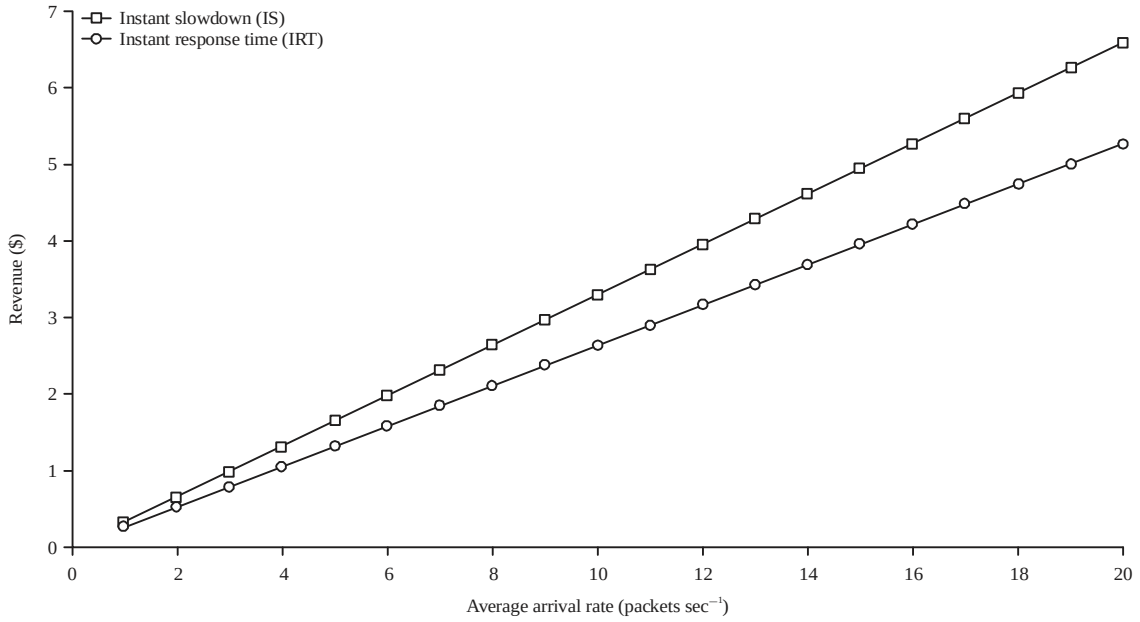

Fig. 4: Variation of revenue with arrival rate for IRT and IS under FCFS policy

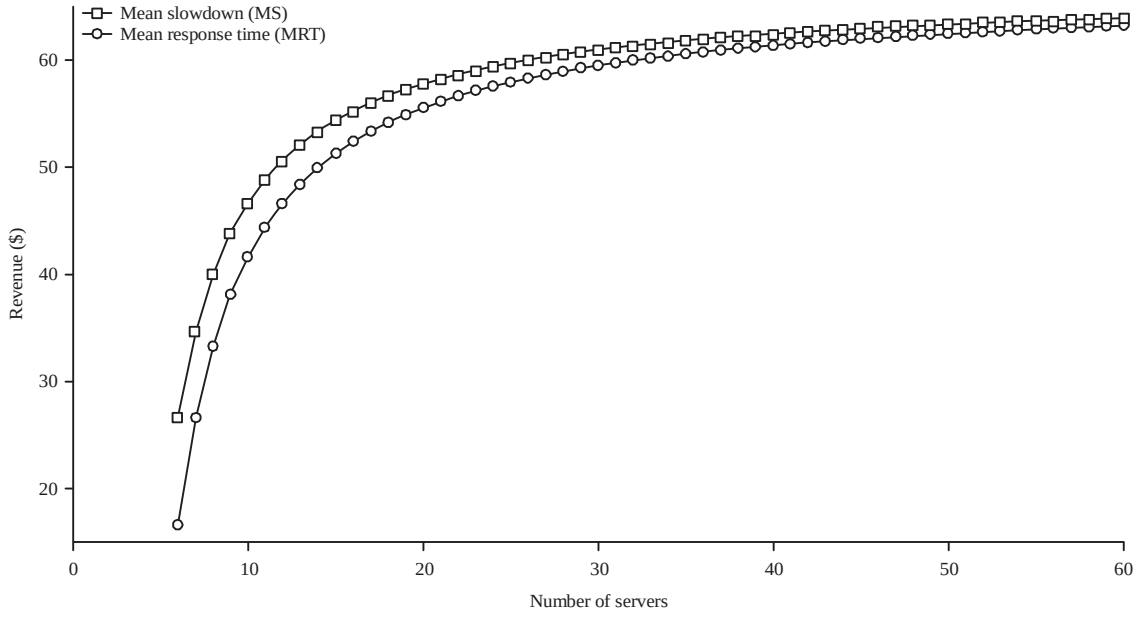

Fig. 5: Variation of revenue with number of servers for MRT and MS under PS policy

when the number of servers is 10, the revenue generated using MRT pricing mechanism is approximately $42.0, while the revenue generated using MS pricing mechanism is approximately $47.0. The difference in revenue generated using MS and MRT is higher for lower number of servers as compared to higher number of servers where the difference in revenue is less.

Figure 6 shows the variation of revenue as a function of arrival rate for mean response time (MRT) and mean slowdown (MS) pricing mechanisms under PS scheduling policy. In doing this, Eq. 27 and 33 were used to plot the graph of revenue as a function of arrival rate. To investigate the effect of increasing the arrival rate on revenue for the two pricing schemes, fix the number of servers, the service rate

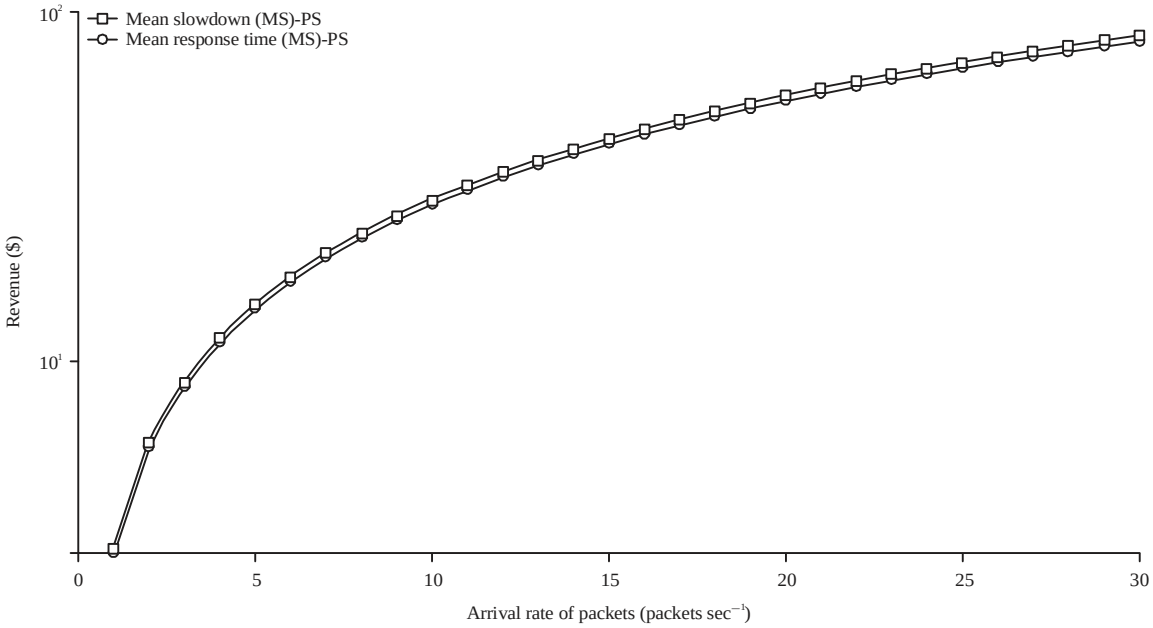

Fig. 6: Variation of revenue with arrival rate for MRT and MS under PS policy

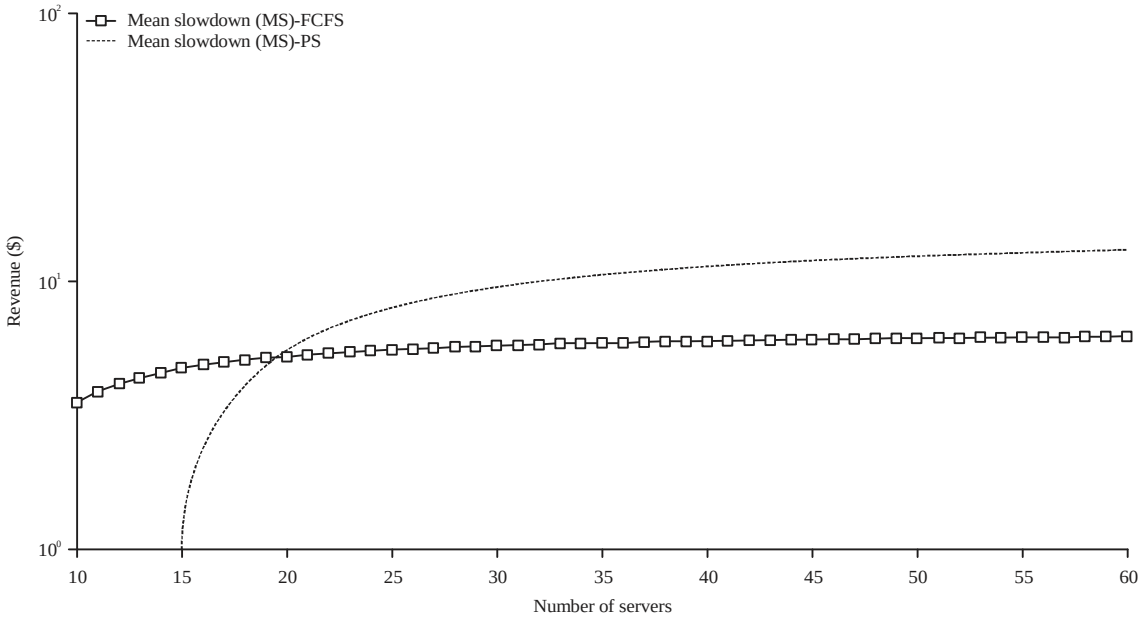

Fig. 7: Variation of revenue with number of servers in terms of mean slowdown for FCFS and PS

and size of requests. It is observed that revenue generally increases with increase in arrival rate regardless of the pricing mechanism used. This is because as the arrival rate increases, the number of requests into the system also increases and hence more revenue is generated. It is further observed that more revenue is generated when MS pricing mechanism is used and when MRT pricing mechanism is used.

Comparison of FCFS and PS policies in terms of MS: This section investigates the variation of revenue with number of servers and arrival rate of packets in the system under FCFS and PS policies charged based on MS.

Figure 7 shows the variation of revenue as a function of number of servers for mean slowdown (MS) pricing mechanism under FCFS and PS scheduling policies. Equations 6 and 33 were used to plot the graph of revenue as

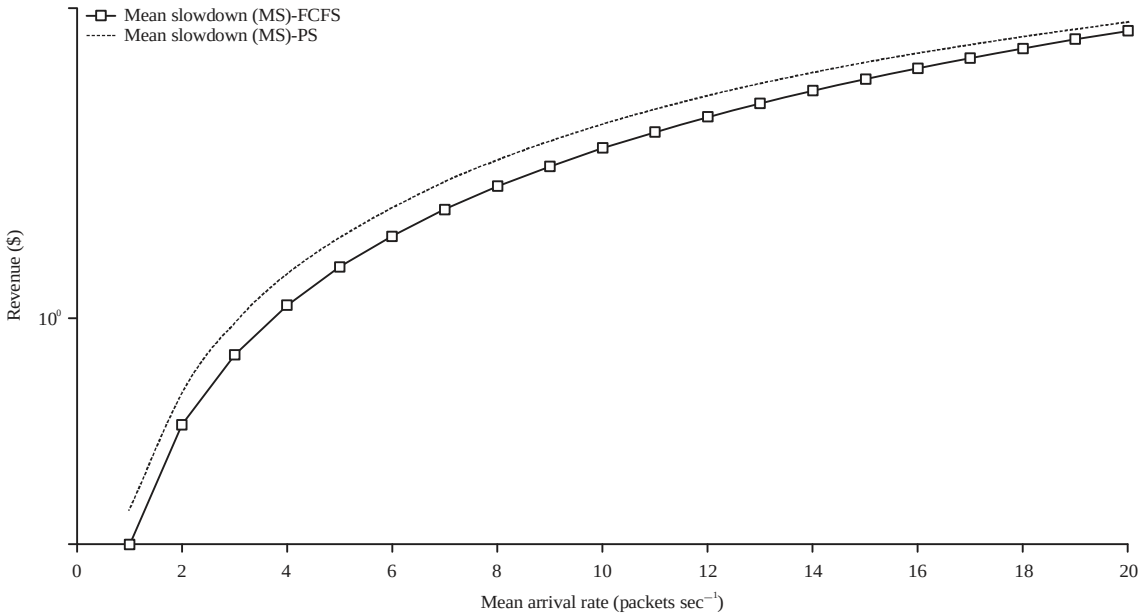

Fig. 8: Variation of revenue with arrival rate in terms of mean slowdown for FCFS and PS

a function of number of servers. To investigate the effect of increasing the number of servers on revenue for the two scheduling policies in terms of mean slowdown, fix the arrival rate, the service rate and size of requests. It is observed that revenue generally increases with increase in number of servers regardless of the scheduling policy used. Furthermore, it is observed that for low number of servers, FCFS policy generates more revenue than PS policy, however as the number of servers increase, PS policy generates more revenue than FCFS. For example, when the number of servers is 40, the revenue generated under the FCFS policy is $6.0 while the revenue generated under the PS policy is approximately $11.0. In addition, when the number of servers is approximately 20, the revenue generated by the two scheduling schemes are equal.

Figure 8 shows the variation of revenue as a function of arrival rate for mean slowdown (MS) pricing mechanism under FCFS and PS scheduling policies. Equations 6 and 33 were used to plot the graph of revenue as a function of arrival rate. To investigate the effect of increasing the arrival rate on revenue for the two scheduling policies in terms of mean slowdown, fix the number of servers, the service rate and size of requests. It is observed that revenue generally increases with increase in arrival rate regardless of the scheduling policy used. Furthermore, it is observed that PS scheduling policy generates more revenue than FCFS policy irrespective of the arrival rate. For example, when the arrival rate is 2 packets sec^{-1}, the revenue generated under FCFS policy is approximately $0.35, while the revenue generated under PS policy is approximately $0.45.

Comparison of IRT and IS under PS: This section compares IRT and IS pricing mechanisms under PS scheduling scheme.

Figure 9 shows the variation of revenue as a function of number of servers for instant response time (IRT) and instant slowdown (IS) pricing mechanisms under PS scheduling policy. Equations 2 and 19 were used to plot the graph of revenue as a function of number of servers. To investigate the effect of increasing the number of servers on revenue for the two pricing schemes, fix the arrival rate, the service rate and size of requests. It is observed that revenue generally increases with increase in number of servers regardless of the pricing mechanism used. It is further observed that for low number of servers, IRT pricing mechanism generates more revenue than IS pricing mechanism, however as the number of servers increase, IS pricing mechanism generates more revenue than IRT pricing mechanism. In addition, the increase in revenue remains constant after deploying approximately 20 servers.

Figure 10 shows the variation of revenue as a function of arrival rate for instant response time (IRT) and instant slowdown (IS) pricing mechanisms under PS scheduling policy. To investigate the effect of increasing the arrival rate on revenue for the two pricing schemes, fix the service rate, the number of servers and size of requests. Equations 2 and 19 were used to plot the graph of revenue as a function of arrival rate. It is observed that revenue generally increases with increase in arrival rate regardless of the pricing mechanism used. Furthermore, it is observed that IS pricing mechanism generates slightly more revenue than IRT pricing scheme.

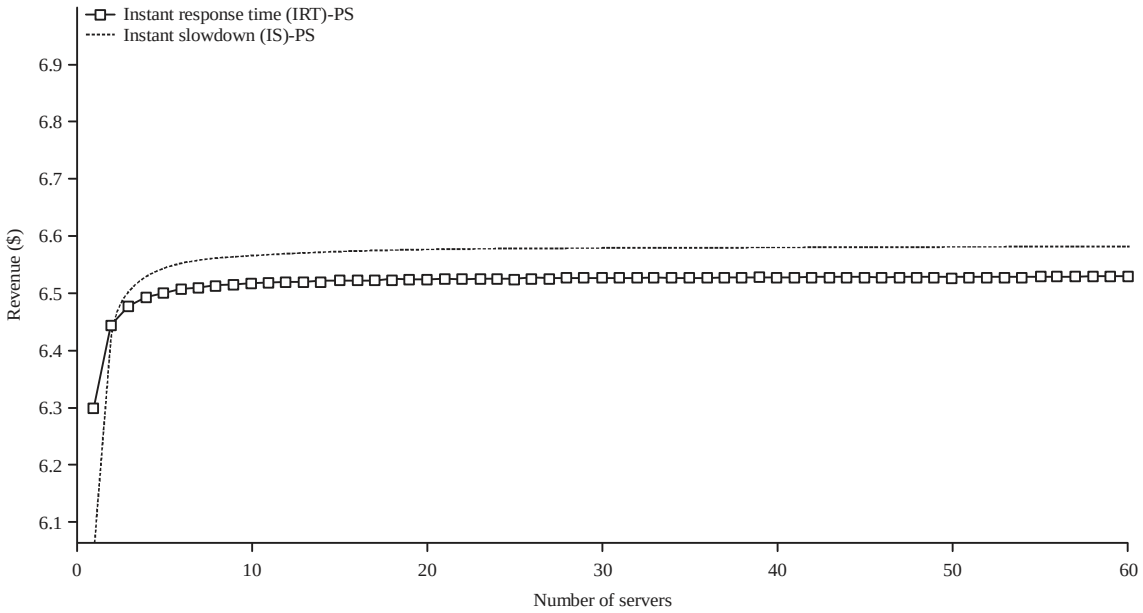

Fig. 9: Variation of revenue with number of servers for IRT and IS under PS

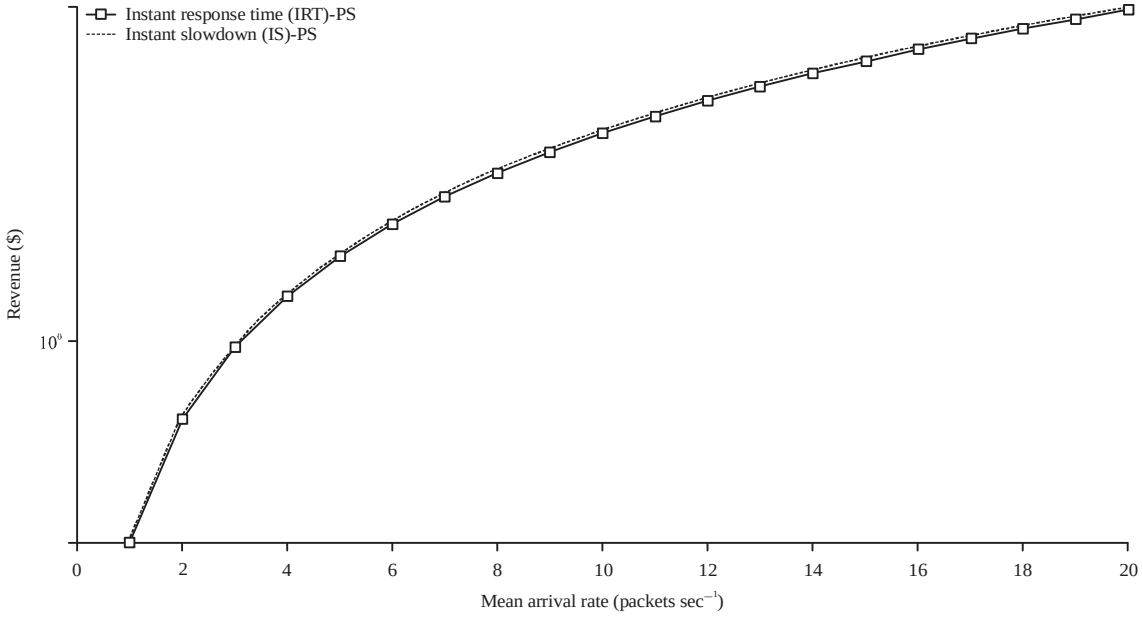

Fig. 10: Variation of revenue with arrival rate for IRT and IS under PS

Comparison of FCFS and PS policies in terms of IS: This section, evaluates the performance of FCFS and PS policies under IS pricing mechanism in terms of revenue generated.

Figure 11 shows a graph of revenue against number of servers for instant slowdown (IS) pricing mechanism under FCFS and PS scheduling policies. Equations 19 and 35 were used to plot the graph of revenue as a function of number of servers. To investigate the effect of increasing the number of servers on revenue for the two scheduling policies, fix the arrival rate, the service rate and size of requests. It is observed that revenue generally increases with increase in number of servers irrespective of the scheduling policy used. It is also observed that PS scheduling policy generates more revenue than FCFS for lower number of servers, however as the number of servers increase the revenue generated under the two policies become closer and finally become the same after deploying approximately 17 servers.

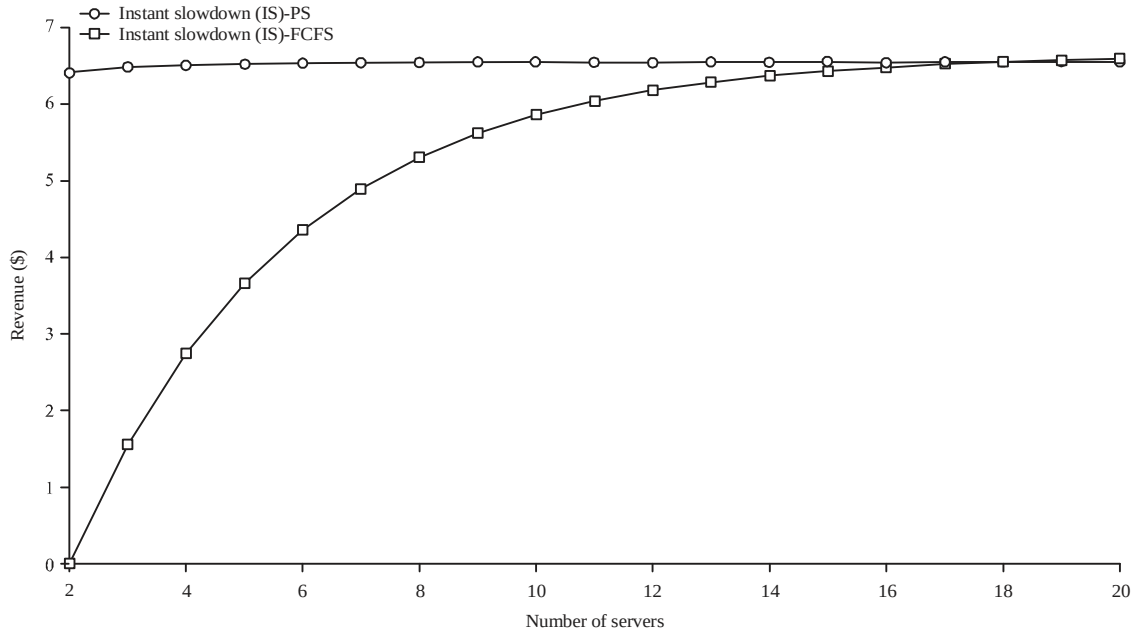

Fig. 11: Variation of revenue with number of servers for FCFS and PS in terms of IS

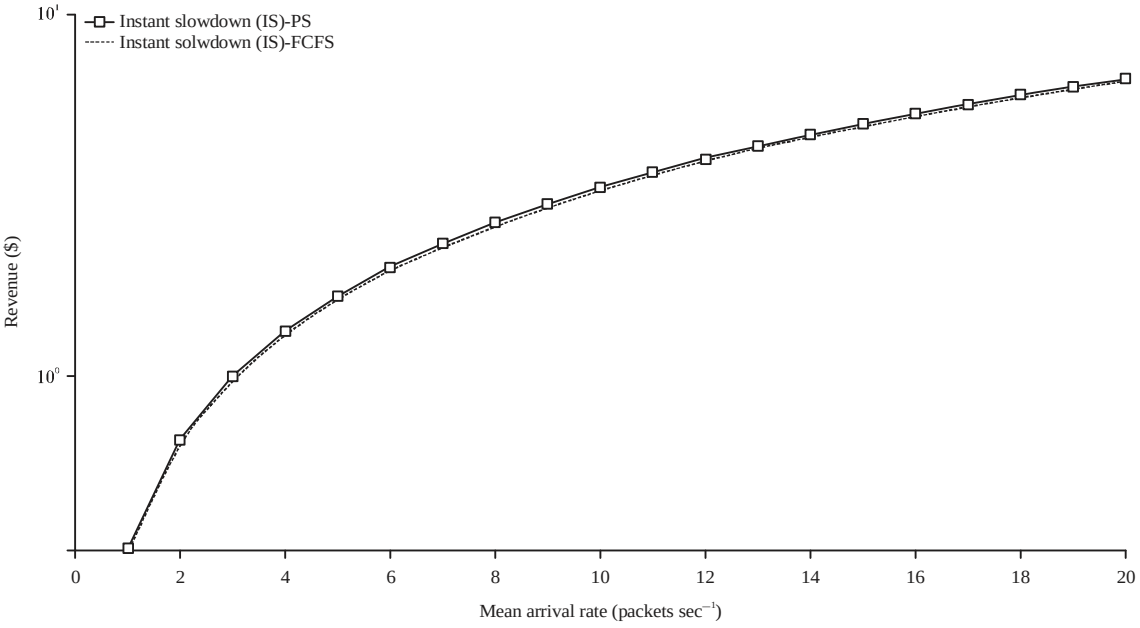

Fig. 12: Variation of revenue with arrival rate for FCFS and PS in terms of IS

Figure 12 shows a graph of revenue against arrival rate for instant slowdown (IS) pricing mechanism under FCFS and PS scheduling policies. Equations 19 and 35 were used to plot the graph of revenue as a function of arrival rate. To investigate the effect of increasing the arrival rate on revenue for the two scheduling policies, fix the number of servers, the service rate and size of requests. It is observed that revenue generally increases with increase in number of servers irrespective of the scheduling policy used. It is also observed that FCFS and PS scheduling policies generate almost the same revenue for all considered arrival rate values.

DISCUSSION

Previous study done by Feng *et al.*[5] showed that the resource allocation strategy of MRT and IRT outperforms the Heuristic strategy proposed by Mazzucco[9], under FCFS policy. In this study, the proposed customer oriented pricing mechanisms MS and IS are found to outperform MRT and IRT resource allocation strategies proposed by Feng *et al.*[5]. This is due to the fact that MS and IS takes into consideration the time when the request terminates in addition to the length of the request unlike MRT and IRT which only focuses on the time when a request terminates. It is further observed that revenue generated under MS and IS pricing mechanisms are higher than revenue generated under MRT and IRT pricing mechanisms. The higher revenue generated under MS and IS pricing mechanisms are due to the fact that MS and IS pricing mechanisms are more representative of the performance of a larger fraction of requests compared to MRT and IRT where the performance of some few large requests may imply an overall increase in performance. It is also observed that PS policy generally generates more revenue than FCFS policy especially when there are more servers. PS policy generates more revenue due to the fact that PS policy shares the servers equally at any given time, while for FCFS a large request may starve short requests. However, when the number of servers is low, FCFS scheduling policy generates more revenue than PS policy.

CONCLUSION

Analytical models of pricing mechanisms based on mean slowdown and instant slowdown are developed for cloud computing under FCFS and PS scheduling policies. The models are used to compare the performance of response time and slowdown under FCFS and PS scheduling policies in terms of revenue generated. The numerical results obtained from the derived models show that revenue generated under slowdown pricing mechanisms are higher than revenue generated under response time pricing mechanisms. It is further observed that PS policy generally generates more revenue than FCFS policy especially when there are more servers.

SIGNIFICANCE STATEMENTS

This study discovers the possibility of charging prices based on slowdown that can be beneficial for the service provider. This study will help researchers to uncover the critical area of charging prices based on slowdown that many researchers were not able to explore. Thus, a new theory on cloud pricing mechanisms may be arrived at.

REFERENCES

1. Mell, P. and T. Grance, 2011. The NIST Definition of Cloud Computing. NIST Special Publication, USA., pp: 1-3.
2. Huth, A. and J. Cebula, 2009. The basics of cloud computing. United States Emergency Rescue Team (CERT), USA., pp: 1-4.
3. WEF., 2010. Exploring the future of cloud computing: Riding the next wave of technology-driven transformation. World economic forum in partnership with accenture report. World Economic Forum, Switzerland, pp: 179-208.
4. Kleinrock, L., 1976. Queuing Systems. Vol. 1. John Wiley and Sons, New York.
5. Feng, G., S. Garg, R. Buyya and W. Li, 2012. Revenue maximization using adaptive resource provisioning in cloud computing environments. Proceedings of the 13th ACM/IEEE International Conference on Grid Computing, Volume 13, September 20-23, 2012, IEEE Computer Society Washington, DC, USA., pp: 192-200.
6. Yeo, C.S., S. Venugopal, X. Chu and R. Buyya, 2010. Autonomic metered pricing for a utility computing service. Future Generat. Comput. Syst., 26: 1368-1380.
7. Mihailescu, M. and Y.M. Teo, 2010. Dynamic resource pricing on federated clouds. Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, May 17-20, 2010, IEEE Computer Society, pp: 513-517.
8. Zhu, H., H. Tang and T. Yang, 2001. Demand-driven service differentiation in cluster-based network servers. Proceedings of the 20th Annual Joint Conference of the IEEE Computer and Communications Societies, Volume 2, April 22-26, 2001, IEEE., pp: 679-688.
9. Mazzucco, M., 2009. Revenue maximization problems in commercial data centers. Ph.D. Thesis, University of Newcastle, Australia.
10. Wierman, A., 2010. Scheduling for today's computer systems: Bridging theory and practice. Ph.D. Thesis, Carnegie Mellon University.
11. Okopa, M. and T. Bulega, 2012. Analysis of fixed priority SWAP scheduling policy for real-time and non real-time jobs. Int. J. New Comput. Archit. Appli., 2: 488-495.
12. Downey, A.B., 1997. A parallel workload model and its implications for processor allocation. Proceedings of the International Symposium of High Performance Distributed Computing, August 5-8, 1997, Portland, OR, USA., pp: 112-123.