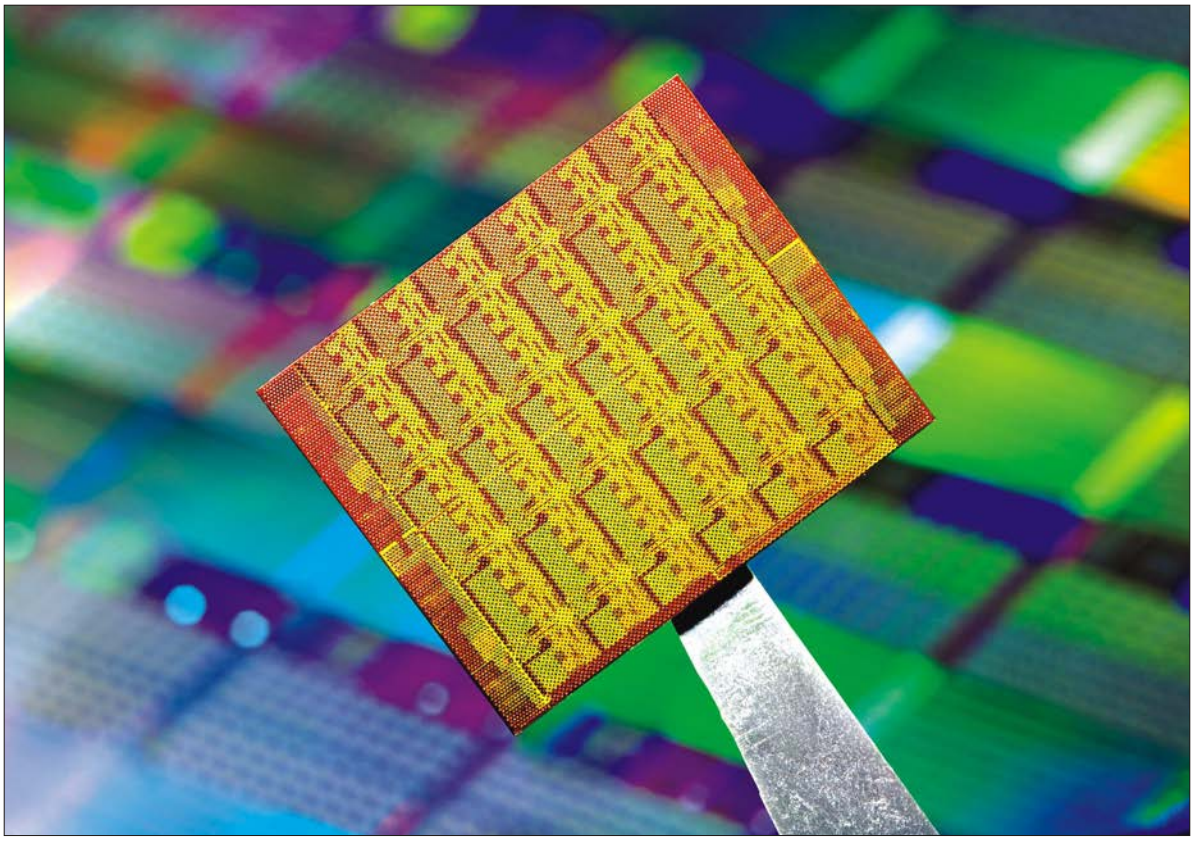

Australasian Journal of

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.2017.32.43

Research Article
Pricing Scheme for Heterogeneous Multiserver Cloud Computing System

[1]Barbara Nansamba, [1]Kyanda Swaib Kaawaase, [2]Michael Okopa and [3]Barbara K. Asingwire

[1]Department of Networks, School of Computing and Informatics Technology, Makerere University Kampala, Kampala, Uganda
[2]Department of Computer Science, Faculty of Science and Technology, Cavendish University Uganda, Kampala, Uganda
[3]Department of Computer Engineering, Faculty of Engineering, Busitema University, Tororo, Uganda

Abstract

Background and Objective: Previous works on pricing in cloud computing environments assumed cloud servers are homogeneous. The assumption of homogeneous servers was not realistic and cannot accurately model practical deployment scenarios of cloud servers since cloud providers deploy heterogeneous servers with different service rates and capacities. The objective of this study was to model a pricing scheme for heterogeneous cloud computing servers based on response time and slowdown. **Methodology:** To overcome the above challenge, this study proposed a pricing model for heterogeneous multiserver cloud computing system. Heterogeneous multiserver cloud computing systems had different capacities in terms of service rate and processing power. The proposed pricing mechanism was charged based on mean response time and mean slowdown. Mean slowdown was introduced as a performance metric because it was representative of the size of all requests in the system unlike mean response time used in previous studies which was representative of the size of requests which were larger in size and not representative of all requests. Queueing theory was employed to derive expressions for revenue in terms of mean response time and mean slowdown. The performance of the heterogeneous multiserver system was compared to homogeneous system using MATLAB. **Results:** Numerical results showed that heterogeneous multiserver system generated more revenue than homogeneous multiserver system especially at high load and high arrival rate values for both pricing mechanisms based on response time and slowdown. It was further observed that more revenue generated when mean slowdown was used as a charging metric than when mean response time was used, especially at high load values and high arrival rates. **Conclusion:** Heterogeneous multiserver system generated more revenue than homogeneous multiserver system. In addition, mean slowdown generated more revenue when used as a charging metric than mean response time.

Key words: Heterogeneous multiserver, homogeneous multiserver, mean slowdown, pricing mechanism, probability

Citation: Barbara Nansamba, Kyanda Swaib Kaawaase, Michael Okopa and Barbara K. Asingwire , 2017. Pricing scheme for heterogeneous multiserver cloud computing system. Australasian J. Comp. Sci., 4: 32-43.

Corresponding Author: Michael Okopa, Department of Computer Science, 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 has emerged as a vital practice for the online provisioning of computing resources as services[1,2]. This technology allows scalable on demand sharing of resources and costs among a large number of end users. Cloud computing enables end users to process, manage and store data efficiently at very high speeds at reasonable prices.

A majority of technology experts expect that by 2020 most people will access software applications online and share and access information through the use of remote server networks using the cloud, rather than depending primarily on tools and information housed on their individual personal computers[3].

Cloud computing providers offer many services to their customers, including infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS), storage as a service (STaaS), security as a service (SECaaS), test environment as a service (TEaaS), etc[4].

In cloud computing, computing infrastructure and services should always be available on computing servers (which are distributed among all continents) to enable companies access their business services and applications anywhere in the world whenever they need to[5].

Many definitions have been presented for cloud computing[6,7]. According to the University of California at Berkeley, "Cloud computing refers to both the applications delivered as services over the internet and the hardware and systems software in the data centers that provide those services"[8]. Cloud computing comes from the use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams[9,10].

One attractive cloud computing environment is a three-tier structure, which consists of infrastructure vendors, service providers and customers[11]. An infrastructure vendor maintains basic hardware and software facilities[12]. A service provider rents resources from the infrastructure vendors and provides various services to customers. A customer submits a service request to a service provider receives the desired result from the service provider with certain service level agreement and pays for the service based on the amount of the service and the quality of the service.

Basically, a service level agreement (SLA) represents an agreement between a customer and a provider to receive a particular service provision[13]. SLAs contain quality of service (QoS) parameters that must be maintained by a provider (e.g. response time, bandwidth, storage, reliability, deadline, throughput, delay and cost)[14].

A service provider can build different multiserver systems for different applications domains, such that service requests of different nature were sent to different multiserver systems[15]. Each multiserver system contains multiple servers that can be devoted to serve one type of service request and application. The configuration of a multiserver system was characterized by two basic features, i.e., the size of the multiserver system (the number of servers) and the speed of the multiserver system (execution speed of the servers).

Pricing is a critical factor for organizations offering services or products[16]. Customer behavior, loyalty to a provider and the organization's success was normally affected by how the price was set. Pricing is the process of determining what a service provider will receive from an end user in exchange for their services. According to Weinhardt *et al.*[17] cloud computing success can be obtained only by developing adequate pricing techniques. Therefore, developing an appropriate pricing model will help achieve higher revenues.

Existing studies on pricing in cloud computing environments assume cloud servers are homogeneous[15,18,19]. Homogeneous cloud computing servers consists of same storage capacity, processing power, energy supply and same service rate. However, the real case scenario of cloud server systems was not represented by homogeneous servers. Cloud server systems consist of heterogeneous servers with different service rates[20]. The need for increased heterogeneity in the computing systems was partially as the need for high-performance, highly reactive systems that interact with other environments too[21]. Therefore, the motivation for developing pricing models for heterogeneous multiserver systems stems from the fact that pricing models based on homogeneous multiserver systems cannot accurately represent current deployment scenarios of cloud servers that deploy servers with different service rates and capacities.

Relatedly several researches based on mean response time to charge in cloud computing[16,17,19] however, mean response time tends to represent the performance of just a few big requests since they count the most in the mean because their response times tend to be highest[22]. In other words an improvement in mean response time could imply the performance of a few big requests has improved. Mean response time was the total time a request spends in the system and this includes, waiting time in the queue and service time[23].

Other parameters that could be used to charge in cloud computing was mean slowdown[24]. Mean slowdown is the ratio of mean response time to the size of the request[25]. The advantages of mean slowdown over mean response time is

that it is more representative of the performance of a larger fraction of requests. Secondly, mean slowdown ensures that a request's mean response time is correlated to its size.

Wang *et al.*[26] developed two distributed algorithms for the net profit optimization, net profit optimization for divisible jobs (NPOD) and net profit optimization for indivisible jobs (NPOI). An indivisible job is a job that cannot be interrupted, while a divisible job is one that can be interrupted. The authors proved via simulations that the two algorithms can increase revenues and reduce electricity costs by comparing it to the largest job first (LJF) algorithm. However, the authors considered only static job arrivals and departures. In addition, they assumed that the servers at all data centers were homogenous, which does not depict the real cloud server deployment scenarios. The disadvantage of homogeneous multiserver system was that it exhibits increased execution time for several tasks with overall reduction in performance.

Mihailescu and Teo[27] introduced a dynamic pricing scheme for federated clouds, in which resources were shared among many cloud service providers. The authors carried out simulations to determine the efficiency of this approach by comparing it to a fixed pricing scheme. They found that dynamic pricing achieved better average performance with increasing buyer welfare and numbers of successful requests. However, fixed pricing achieved better scalability in the case of high demand in the market.

In an effort to maximize revenue, Feng *et al.*[19] scheduled the cloud resources among different service instances adaptively based on the dynamically collected information. In their study, each service instance, a virtual machine associated with a user, is modeled as a FIFO (first in first out) M/M/1/FIFO queue system. 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. The optimal number of servers required to maximize profit were obtained.

The pricing mechanism proposed by Cao *et al.*[15] is as:

$$G = \alpha.\frac{1}{\lambda}\left(m\rho + \frac{(m\rho)^m}{m!}.\frac{\rho}{(1-\rho)^2}P_o \right) \quad (1)$$

Where:

α = Service charge per unit amount of service
P_o = As given in Eq. 2
m = Number of servers
ρ = Load in the system
P_o = Probability

$$P_o = \left(\sum_{j=0}^{m-1} \frac{(m\rho)^j}{j!} + \frac{(m\rho)^m}{m!(1-\rho)} \right)^{-1} \quad (2)$$

In a recent study, Cao *et al.*[15] proposed an optimal multiserver configuration for profit maximization in a cloud computing environment. The authors assumed the multiserver system to be homogeneous implying that the servers are identical, in addition, mean response time was used as the performance metric. However, homogeneous multiserver system cannot capture the heterogeneity exhibited by cloud servers[28]. Furthermore, profit maximization based on mean response was representative of profit of requests which were larger in size and not representative of all requests. To overcome the above challenge, this study proposed a revenue model which takes into consideration the heterogeneity of the servers and charges according to achieved service performance in terms of mean response time or mean slowdown.

MATERIALS AND METHODS

This study used analytical methodology to evaluate the performance of the proposed models. Analytical methodology is a generic process combining the power of the scientific method with the use of formal process to solve any type of problem. An analytical model therefore is a set of computational algorithms or formulae used to analyze systems. Analytical models provide faster and more computationally efficient methods of obtaining performance measures.

In modeling revenue based on mean response time and mean slowdown, queueing theory was used. Queueing models are suitable in a variety of environments ranging from common daily life scenarios to complex service and business processes, operations research problems, or computer and communication systems. Queueing theory has been extensively applied to evaluate and improve system behavior[19,15]. Specifically, this study used the $M/M_i/m/FCFS$ queue system, where 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[23] is given as:

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad x = 0, 1, 2, \ldots \quad (3)$$

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

M_i represents the service time for server i. The amount of time dedicated to each request on each server is exponentially distributed, with a service rate μ_i, where i = 1, 2, 3... m. The number of parallel servers is m. The queue discipline for each queue was first come first served. The server allocation policy was fast server first (FSF). FSF policy is used to select which server to be allocated a request whereas in each server, requests were serviced using FCFS policy. The service intensity, ρ is defined as the ratio of arrival rate to the service rate, $\rho = \lambda/\mu$. The last m represents the number of servers.

System model: A cloud service provider serves user's requests by using a multiserver system which was constructed and maintained by an infrastructure vendor and rented by the service provider. This study considers a heterogeneous multiserver system where the service rate of each server was different. Customers submit service requests to a service provider and the service provider serves the requests by using the multiserver system. Since the servers were heterogeneous, the selection policy employed to select servers was fast server first (FSF) and the scheduling policy in each server was FIFO. The response time for each customer were determined and mean slowdown calculated upon which the price was charged.

This study considers a multiserver heterogeneous queueing system in which the arrivals follow a Poisson process with mean arrival rate λ and exponentially distributed inter arrival times. Poisson arrival rates were assumed since the requests into the servers were random and memory less. Memory loss was due to the fact that the arrival of the next request does not depend on the arrival of the past requests. The multiserver system maintains a queue with an infinite capacity. The multiserver system was treated as an $M/M_i/m/FSF$ queueing system. The $M/M_i/m/FSF$ queue model was used to drive the mean revenue brought by a service provision. There were m servers (i.e., blades/processors/cores) with different service rates (measured by the number of packets that can be executed per unit time) μ_m, (i = 1, 2... m) for each of the m servers and the service times at each server follows exponential distribution. Each request requires exactly one server and the queue discipline was first come first served (FCFS). This study also assumed that the servers were ordered in decreasing service speed. This implies that the customers were always served by the fastest servers, i.e., when k<m customers were present, servers 1, 2... k were used.

The study formalized the resource allocation problem by considering various parameters such as pricing mechanisms, arrival rates, service rates and available resources. The model metric used was revenue. Revenue was the income generated. Revenue has been used in literature as a performance metric to evaluate the performance of different pricing schemes[15,18]. The revenue generated was expressed in terms of performance parameters like task mean slowdown, i.e., the ratio of the time taken to complete a task to the size of the request, this includes task waiting time and task execution time. The mean slowdown was the source of customer satisfaction. A service provider should keep the mean slowdown to a low level by providing enough servers and/or increasing server speed and be willing to pay back to a customer in case the mean slowdown exceeds certain limits.

The promised mean slowdown to complete a service was expressed in the service level agreement. If the actual length of a service was within the service level agreement, the service could be fully charged. However, if the actual length of a service exceeds the service level agreement, the service charge could be zero. Therefore, longer length of service implies that the service was not charged for. On the other hand, the shorter the actual length of a service was, the greater the service charge.

Derivation of revenue under heterogeneous multiserver system in terms of response time: The heterogeneous multiserver system consists of servers with different capabilities in terms of general purpose processor, special purpose processor, or a co-processor. The heterogeneous multiserver system could be modeled using the $M/M_i/m$ queue system. In which case the first M denotes Markovian and represents Poisson arrivals into systems, m_i represents the service rates for servers i = 1, 2... m. The service rates were exponentially distributed and variable and depends on the state i in which the system was. The allocation policy in the system was fastest server first (FSF). The service rate is defined as shown in Eq. 4:

$$\mu_i = \begin{cases} 0 & i = 0 \\ \mu_1 & i = 1 \\ \mu_1 + \mu_2 & i = 2 \\ . & \\ . & \\ . & \\ \mu_1 + \mu_2 + ... + \mu_m & i \geq m \end{cases} \quad (4)$$

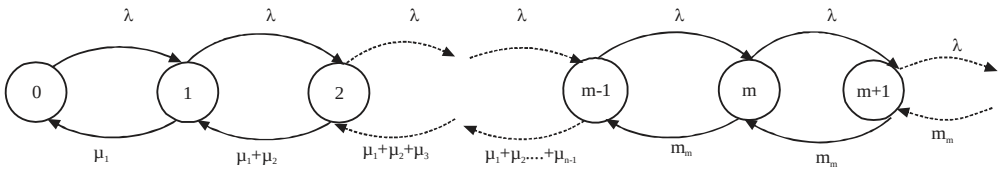

Fig. 1: State transition diagram for $M/m_i/m$ queue system

In this case, $\mu_1 \geq \mu_2 > ... \mu_m$. This implies μ_1 is the fastest server and μ_m is the slowest server. Equation 5 formulated in two ways when the system contains less than m jobs, in which μ_i is a variable and when there are m or more jobs in the system, in which case μ_i is a constant.

The state transition diagram for $M/m_i/m$ system is shown in Fig. 1.

Define:

$$m_i = \begin{cases} \sum_{j=1}^{i} \mu_j & i < m \\ m_m = \sum_{j=1}^{m} \mu_j & i \geq m \end{cases} \quad (5)$$

When i = 1, the system was in state 1 and there was 1 job present in the system and only one server was processing the work (this server assumed to be the fastest). When i = 2, the system was in state 2 and there were 2 jobs present in the system and two servers were processing the work (the two servers are assumed to be the fastest).

Using the state transition diagram in Fig. 1, determine the probability of queueing (i.e. the probability that a newly arriving service request must wait because all the servers were busy). Under the stable conditions:

$\mu_1 P_1 = \lambda P_0$ from which:

$$P1 = \frac{\lambda}{\mu_1} P_0 \quad (6)$$

$(\mu_1+\mu_2) P_2 = \lambda P_1$ from which:

$$P_2 = \frac{\lambda^2}{\mu_1(\mu_1+\mu_2)} P_0 \quad (7)$$

$(\mu_1+\mu_2+\mu_3) P_3 = \lambda P_2$ from which:

$$P_3 = \frac{\lambda^3}{\mu_1(\mu_1+\mu_2)(\mu_1+\mu_2+\mu_3)} P_0 \quad (8)$$

Expression in Eq. 6, 7 and 8 can be generalized into:

$$P_m = \frac{\lambda^m}{\mu_1(\mu_1+\mu_2)(\mu_1+\mu_2+\mu_3)...(m_m)} . P_0 \quad (9)$$

Where:

$$m_m = \mu_1+\mu_2+\mu_3+.... +\mu_m$$

From Eq. 9, note that μ_1 appears in all the denominators. This implies that μ_1 appears *m* times and thus has a great influence on the probability. Similarly, the service rate μ_2 has the second largest weight on the probability and appears (m-1) times, μm which is the slowest service rate appears once.

The probability P_i to find i jobs in the system is as follows:

If i<m:

$$P_i = \frac{\lambda^i}{\pi_{k=i}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} . P_0$$

And if i≥m:

$$P_i = \frac{\lambda^i}{\left(\pi_{j=i}^{m} m_j \right) (m_m)^{i-m}} . P_0$$

$$P_i = \begin{cases} \dfrac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right) m_j} . P_0 & i < m \\ \left(\dfrac{\lambda^i}{\left(\pi_{j=1}^{m} m_j \right) (m_m)^{i-m}} \right) . P_0 & i \geq m \end{cases} \quad (10)$$

Where:

$$m_j = \sum_{k=1}^{j} \mu_k$$

and:

$$m_m = \sum_{k=1}^{m} \mu_k$$

Using the fact that all probabilities Pi must obey the relationship:

$$\sum_{i=0}^{\infty} P_i = 1$$

$$\sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} . Po \right) + \sum_{i=m}^{\infty} \left(\frac{\lambda^i}{\pi_{j=1}^{m} m_j (m_m)^{i-m}} . Po \right) = 1 \quad (11)$$

$$P_o^{-1} = \sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} \right) + \sum_{i=m}^{\infty} \left(\frac{\lambda^i}{\pi_{j=1}^{c} m_j (m_m)^{i-m}} \right) \quad (12)$$

$$P_o^{-1} = \sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} \right) + \left(\frac{(m_m)^m}{\pi_{j=1}^{m} m_j} \right) . \sum_{i=m}^{\infty} \left(\frac{\lambda^i}{m_m} \right)^i \quad (13)$$

In order to prevent the queue from growing indefinitely, the system utilization:

$$\rho = \frac{\lambda}{\pi_{j=1}^{c} \mu_j} = \frac{\lambda}{m_m} < 1$$

Substituting ρ into Eq. 13 leads to the following expression for P_o:

$$P_o^{-1} = \sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} \right) + \left(\frac{(m_m)^m}{\pi_{j=1}^{m} m_j} \right) . \sum_{i=m}^{\infty} \rho^i \quad (14)$$

But:

$$\sum_{i=m}^{\infty} \rho^i = (1-\rho)^{-1} . \rho^m$$

$$P_o^{-1} = \sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} \right) + \left(\frac{(m_m)^m}{\pi_{j=1}^{m} m_j} \right) . \frac{\rho^m}{(1-\rho)} \quad (15)$$

$$P_o^{-1} = \sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} \right) + \left(\frac{(m_m)^m}{\pi_{j=1}^{m} m_j} \right) . \frac{\lambda^m}{\left(1 - \left(\frac{\lambda}{m_m} \right) \right) . (m_m)^m} \quad (16)$$

Finally:

$$P_o^{-1} = \sum_{i=0}^{m-1} \left(\frac{\lambda^i}{\pi_{k=1}^{i} \left(\sum_{j=1}^{k} \mu_j \right)} \right) + \left(\frac{\lambda^m}{(1-\rho) \pi_{j=1}^{m} m_j} \right) \quad (17)$$

To find the average number in the queue or execution, it is necessary to find its expectation (L_q):

$$L_q = \sum_{i=m}^{\infty} (i-m) . P_i \quad (18)$$

$$L_q = \sum_{i=m}^{\infty} (i-m) . P_o . \frac{\lambda^m}{\left(\pi_{j=1}^{m} m_j \right) (m_m)^{i-m}} \quad (19)$$

$$L_q = \frac{P_o . (m_m)^m . \rho^m}{\left(\pi_{j=1}^{m} m_j \right)} . \sum_{i=m}^{\infty} (i-m) \rho^{i-m} \quad (20)$$

Let k = i-m:

$$L_q = \frac{P_o . (m_m)^m . \rho^m}{\left(\pi_{j=1}^{m} m_j \right)} . \sum_{k=0}^{\infty} k . \rho^k \quad (21)$$

$$L_q = \frac{P_o . (m_m)^m . \rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j \right)} . \sum_{k=0}^{\infty} k . \rho^{k-1} \quad (22)$$

$$L_q = \frac{P_o . (m_m)^m . \rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j \right)} . \frac{d}{d\rho} . \left(\sum_{k=0}^{\infty} \rho^k \right) \quad (23)$$

$$L_q = \frac{P_o . (m_m)^m . \rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j \right) . (1-\rho)^2} \quad (24)$$

Using little's law, the average task response time is:

$$\bar{T} = \frac{L_q}{\lambda} = \frac{P_o.(m_m)^m.\rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j\right).(1-\rho)^2}.\frac{1}{\lambda} \quad (25)$$

The mean revenue G brought by a service provision is derived from Cao *et al.*[15] as:

$$G = \alpha\left(\frac{P_o.(m_m)^m.\rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j\right).(1-\rho)^2}.\frac{1}{\lambda}\right) \quad (26)$$

Where a is the service charge per unit amount of service and P_o is as given in Eq. 17. Derivation of revenue under homogeneous multiserver system in terms of mean slowdown

In this section, the expression for revenue in terms of mean slowdown was derived. Using the expression for the average task response time given in Eq. 25 and the definition of mean slowdown being the ratio of average task response time to the size of the request, the expression for mean slowdown can be given as $\bar{T}/x$, where x is the size of the request. x can also be expressed as the reciprocal of average service rate, 1/μ. Therefore, mean slowdown is given as:

$$\bar{S} = \frac{\frac{1}{\lambda}\left(m\rho + \frac{(m\rho)^m}{m!}.\frac{\rho}{(1-\rho)^2}P_o\right)}{\frac{1}{\mu}} \quad (27)$$

Basing on pricing model in Cao *et al.*[15], the mean revenue G brought by a service provision is given as:

$$G = \alpha.\left(\frac{\frac{1}{\lambda}\left(m\rho + \frac{(m\rho)^m}{m!}.\frac{\rho}{(1-\rho)^2}P_o\right)}{\frac{1}{\mu}}\right) \quad (28)$$

Where:

a = Service charge per unit amount of service

P_o = As given in Eq. 17

Derivation of revenue under heterogeneous multiserver system in terms of mean slowdown: In this section, the expression for revenue under heterogeneous multiserver system in terms of mean slowdown is derived. Using Eq. 25 and the definition of mean slowdown being the ratio of average task response time to the size of the request, the expression for mean slowdown can be given as $\bar{T}/x$, where x is the size of the request. x can also be expressed as the reciprocal of average service rate, 1/μ. Therefore, mean slowdown is given as:

$$\bar{S} = \frac{\frac{P_o.(m_m)^m.\rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j\right).(1-\rho)^2}.\frac{1}{\lambda}}{\frac{1}{\mu}} \quad (29)$$

Basing on pricing model in Cao *et al.*[15], the mean revenue G brought by a service provision is:

$$G = \frac{\alpha.\frac{P_o.(m_m)^m.\rho^{(m+1)}}{\left(\pi_{j=1}^{m} m_j\right).(1-\rho)^2}.\frac{1}{\lambda}}{\frac{1}{\mu}} \quad (30)$$

Where α is the price constant and P_o is as given in Eq. 17. The price constant is the service charge per unit amount of service[15]. Next, the performance of the derived models was evaluated.

RESULTS

The basic mathematical symbols and the evaluation parameters that were used in this study were presented. Table 1 shows the basic mathematical symbols used in the analysis.

Table 2 shows the evaluation parameters used in the analysis. The number of servers has been fixed to 5 to ensure that the total service rate for both homogeneous and heterogeneous servers were the same and to ensure that the difference in performance was not brought by the fact

Table 1: Basic mathematical symbols used in the analysis

Parameter	Meaning
λ	Mean arrival rate of requests
α	Service charge per unit amount of service
μ	Mean service rate of requests
ρ	System utilization
m	Number of servers

Table 2: Evaluation parameters

Parameter	Value
Servers in homogeneous multiserver system	5
Servers in heterogeneous multiserver system	5
Packet arrival rate	0-14 packets sec^{-1}
Service rate for homogeneous multiserver	3 packets sec^{-1}
Service rates for heterogeneous multiserver	1, 2, 3, 4, 5 packets sec^{-1}
Price constant for service instance, a	10 cents
Total service rate for all servers	15 packets sec^{-1}

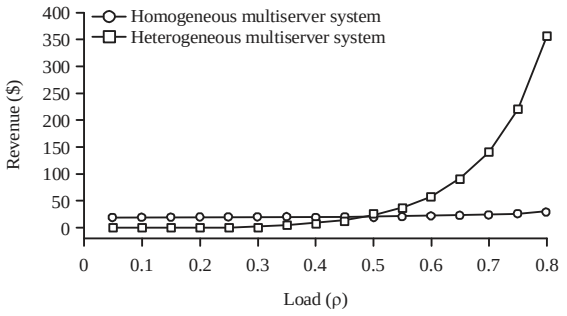

Fig. 2: Variation of revenue with load for homogeneous and heterogeneous multiserver systems in terms of mean response time

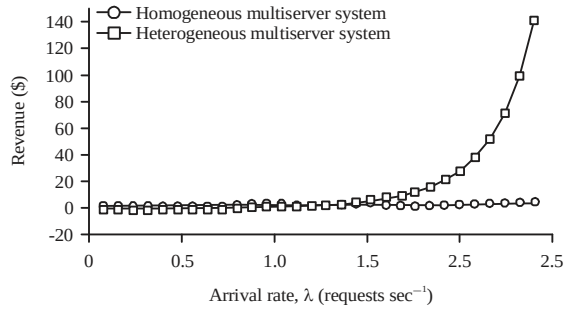

Fig. 3: Variation of revenue with arrival rate for homogeneous and heterogeneous multiserver systems in terms of mean response time

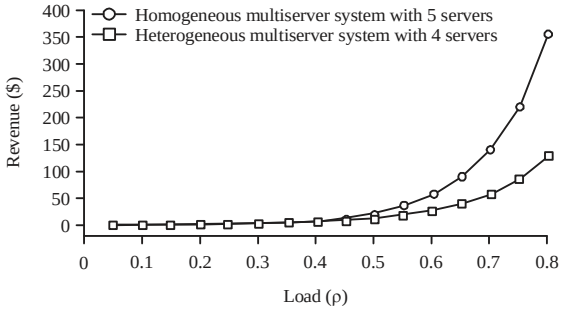

Fig. 4: Variation of revenue with load for a heterogeneous multiserver system in terms of mean response time

that the total service rate for each type of servers was different. That was, the total service rate for 5 homogeneous servers each with a service rate of 3 requests sec^{-1} was 15 packets sec^{-1}, while the total service rate for 5 heterogeneous servers with service rates of 1, 2, 3, 4, 5 packets sec^{-1} was also 15 packets sec^{-1}.

The monetary unit "cent" in this study may not be identical but should be linearly proportional to the real cent in US dollars[15].

Comparison of homogeneous and heterogeneous multiserver systems in terms of load and arrival rate: In this study, the variation of revenue with load and arrival rate for homogeneous and heterogeneous multiserver systems were investigated.

Figure 2 and 3 shows a graph of revenue as a function of load and arrival rate for homogeneous and heterogeneous multiserver systems. Equations 1 and 26 were used to plot the graphs. It was observed that revenue increases with increase in load as well arrival for both homogeneous and heterogeneous multiserver systems. This was due to the fact that as the load in the system increases, there was increased number of requests to be processed and hence more revenue generated. The reason for the increase in arrival rate is same as discussed by load. It was further observed that homogeneous multiserver system generates more revenue than heterogeneous multiserver system for low load values less than 0.5, however, for load values greater than 0.5 the heterogeneous multiserver system generates more revenue than homogeneous multiserver systems. The difference in revenue was more pronounced at higher load values.

It was also observed that for low arrival rate values, the revenue generated from homogeneous and heterogeneous multiserver systems are almost the same, however, as the arrival rate increases the revenue generated from heterogeneous multiserver system was higher than revenue generated from homogeneous multiserver system. The difference in revenue was higher at higher arrival rate values.

Variation of revenue with load and arrival rate in terms of mean response time: In this study, the effect of increasing number of servers on revenue for a heterogeneous multiserver system in terms of mean response time was investigated.

Figure 4 and 5 shows a graph of revenue as a function of load and arrival rate. In doing this, Eq. 26 was used to plot the graph. It was observed that revenue generally increases with increase in load and arrival rate for both considered number of servers. This was due to the fact that as the load increases, the number of requests to be processed also increases thereby generating more revenue. Similarly, when arrival rate increases, the number of requests in the system also increases and generates more revenue. It was further observed that for low load and arrival rate values, revenue generated was almost the same for both considered number of servers, however, as the load and arrival rate increases the revenue generated was higher for higher number of servers as compared to lower number of servers. This observation implies that deploying higher number of servers was more effective at higher load and arrival rate values.

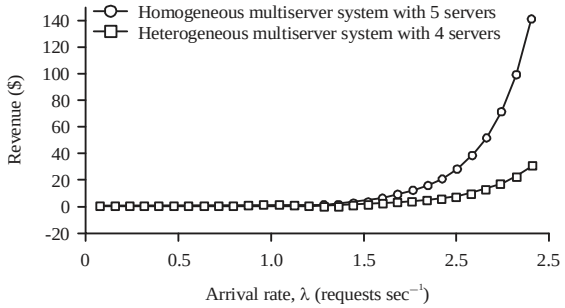

Fig. 5: Variation of revenue with arrival rate for a heterogeneous multiserver system in terms of mean response time

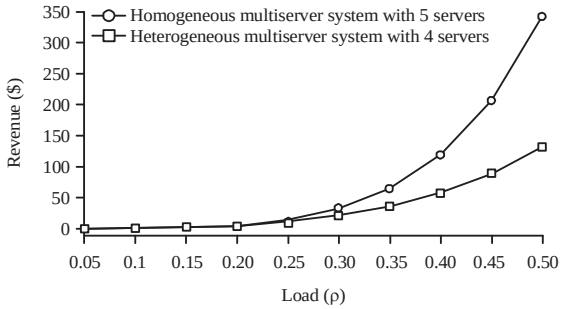

Fig. 6: Variation of revenue with load for a heterogeneous multiserver system in terms of mean slowdown

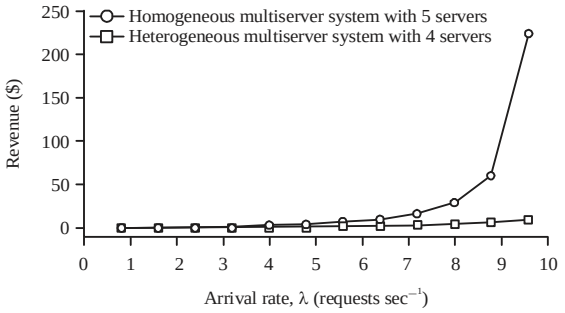

Fig. 7: Variation of revenue with arrival rate for a heterogeneous multiserver system in terms of mean slowdown

Variation of revenue with load and arrival rate in terms of mean slowdown: In this study, the effect of increasing number of servers on revenue for a heterogeneous multiserver system in terms of mean slowdown was investigated.

Figure 6 and 7 shows a graph of revenue as a function of load and arrival rate for heterogeneous multiserver system in terms of mean response time. Equation 30 was used to plot the graph of revenue as a function of load and arrival rate. It was observed that revenue generally increases with increase

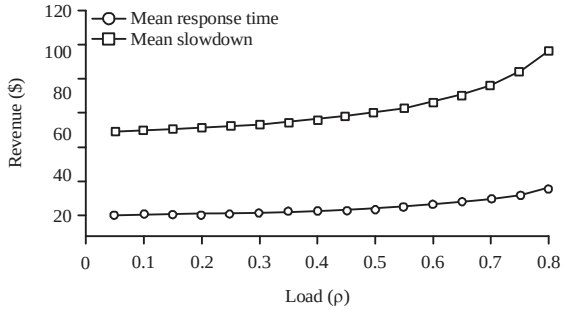

Fig. 8: Variation of revenue with load for a homogeneous multiserver system in terms of mean response time and mean slowdown

in load and arrival rate for both considered number of servers. The increase in revenue was due to the fact that as the load increases, the number of requests to be processed also increases thereby generating more revenue. Similarly, the increase in revenue with arrival rate can be explained by the fact that as the arrival rate increases, the number of requests to be processed also increases hence leading to increase in revenue. It was further observed that for low load and arrival rate values the revenue generated was almost the same for both considered number of servers, however as the load and arrival rate increases the revenue generated was higher for higher number of servers as compared to lower number of servers. This same trend is observed for mean response time.

Variation of revenue with load and arrival rate for a homogeneous multiserver system in terms of mean response time and mean slowdown: In this study, the variation of revenue with load and arrival rate for a homogeneous multiserver system in terms of mean response time and mean slowdown was investigated.

Figure 8 and 9 shows a graph of revenue as a function of load and arrival rate for a homogeneous multiserver system charged based on mean response time and mean slowdown. Equation 1 and 28 were used to plot the graph. It was observed that revenue generally increases with increase in load and arrival for both mean response time and mean slowdown. The increase in revenue was due to the fact that as the load and arrival rate increases, the number of requests being processed also increases hence leading to increase in revenue. It was also observed that more revenue was generated when a homogeneous multiserver system was charged based on mean slowdown than when it was charge based on mean response time. More revenue was generated using mean slowdown as a charging metric due to the fact

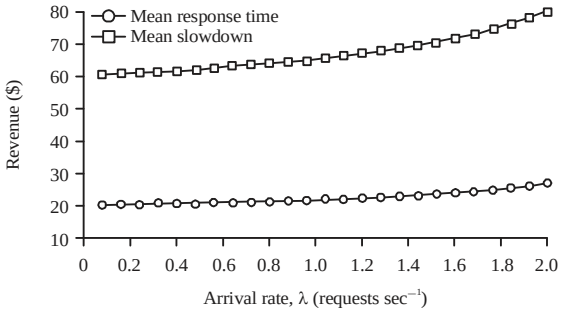

Fig. 9: Variation of revenue with arrival rate for a homogeneous multiserver system in terms of mean response time and mean slowdown

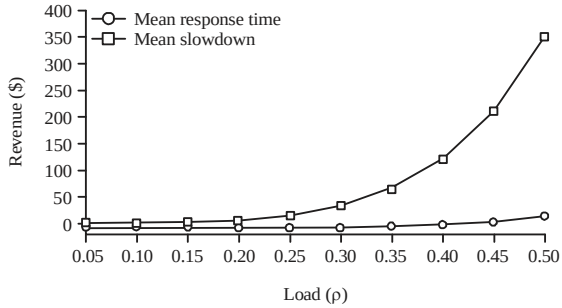

Fig. 10: Variation of revenue with load for a heterogeneous multiserver system in terms of mean response time and mean slowdown

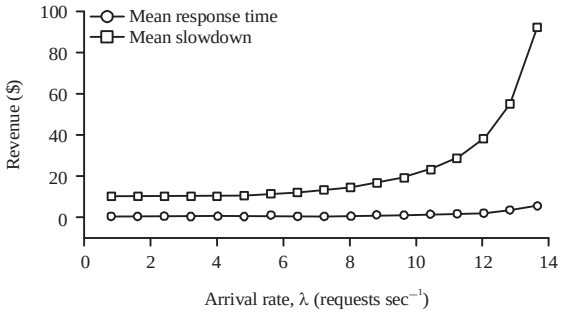

Fig. 11: Variation of revenue with arrival rate for a heterogeneous multiserver system in terms of mean response time and mean slowdown

that mean slowdown takes into account the size of the requests while charging whereas mean response time does not take into account the size of requests.

Variation of revenue with load and arrival rate for a heterogeneous multiserver system in terms of mean response time and mean slowdown: This study investigated the variation of revenue with load and arrival rate for a heterogeneous multiserver system in terms of mean response time and mean slowdown. In doing this, the effect of using mean response time and mean slowdown on revenue for a heterogeneous multiserver system was analyzed.

Figure 10 and 11 shows a graph of revenue as a function of load and arrival rate for a heterogeneous multiserver system charged based on mean response time and mean slowdown. Equation 26 and 30 were used to plot the graph. It was observed that revenue generally increases with increase in load and arrival rate for both mean response time and mean slowdown. The increase in revenue was as a result of increased number of requests in the system due to increase in load and arrival rate. It was also observed that for low load and arrival rate values the difference in revenue between the charging metric based on mean slowdown and mean response time was low, however as the load and arrival rate increases the difference in revenue between the two charging metrics was higher.

DISCUSSION

Previous studies on pricing mechanisms assumed the multiserver system to be homogeneous implying that the servers were identical[19,15]. However, the limitations of the above works are that the usage of heterogeneous servers was not taken into account in cloud networks which does not represent the real case scenario of cloud server systems. This study proposed a model for a heterogeneous multiserver cloud computing system in which the heterogeneity of the servers were taken into consideration in order to represent real case scenario of cloud server systems. The numerical results obtained from the derived models showed that revenue generated for heterogeneous multiserver system was higher than for homogeneous multiserver system proposed by Feng *et al.*[19], Cao *et al.*[15], Anselmi *et al.*[29], Niyato *et al.*[30] and Nan *et al.*[31]. Heterogeneous multiserver systems generated more revenue because they have high-performance and highly interact with other environments. It was also observed that more revenue was generated when mean slowdown was used as a charging metric than when mean response time was used as a charging metric for both homogeneous and heterogeneous servers especially at high load values and high arrival rates. More revenue was generated using mean slowdown as a charging metric due to the fact that mean slowdown takes into account the size of the requests while charging whereas mean response time does not take into account the size of requests.

CONCLUSION

This study derived models for revenue for heterogeneous multiserver system in terms of mean response time and mean slowdown. The performance of the proposed revenue models are compared with the homogeneous revenue model. The numerical results obtained from the derived models show that heterogeneous multiserver system generates more revenue than homogeneous multiserver system. It is also observed that more revenue is generated when mean slowdown is used as a charging metric than when mean response time is used. The study analyzed only the heterogeneous multiserver system with one class of customers. In future, it can be interesting to extend the model to multiple classes of customers where some customers have priority over other customers.

SIGNIFICANCE STATEMENTS

This study discovers the possible ways of modeling revenue for heterogeneous multiserver cloud computing systems and pricing scheme based on mean slowdown. It is expected that this study will help researchers to uncover possible ways of modeling revenue for heterogeneous multiserver cloud computing systems and new ways of charging cloud computing clients.

REFERENCES

1. Al-Roomi, M., S. Al-Ebrahim, S. Buqrais and I. Ahmad, 2013. Cloud computing pricing models: A survey. Int. J. Grid Distrib. Comput., 6: 93-106.
2. Lochan, V.B. and N.K. Gupta, 2015. Dynamic business model outsourcing for data integrity in clouds. Int. J. Curr. Eng. Technol., 5: 935-941.
3. Garimella, S., N. Garg and Vikasdeep, 2012. Features, benefits, futuristic projections of cloud and intercloud extensions to the NET. Int. J. Innov. Eng. Technol., 1: 23-30.
4. Buyya, R., D. Abramson, J. Giddy and H. Stockinger, 2002. Economic models for resource management and scheduling in grid computing. J. Concurr. Commun. Pract. Exp., 14: 1507-1542.
5. Gorelik, E., 2013. Cloud computing models. Master's Thesis, Massachusetts Institute of Technology, Engineering Systems Division, Cambridge, MA., USA.
6. Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. Proceedings of the Grid Computing Environments Workshop, November 12-16, 2008, Austin, TX., USA., pp: 1-10.
7. Vaquero, L.M., L. Rodero-Merino, J. Caceres and M. Lindner, 2009. A break in the clouds: Towards a cloud definition. ACM SIGCOMM Comput. Commun. Rev., 39: 50-55.
8. Etro, F., 2009. The economic impact of cloud computing on business creation, employment and output in Europe. An application of the endogenous market structures approach to a GPT innovation. Rev. Bus. Econ. Lit., 54: 179-208.
9. Acharjya, D.P., S. Dehuri and S. Sanyal, 2015. Computational Intelligence for Big Data Analysis: Frontier Advances and Applications. Springer International Publishing, Switzerland, ISBN-13: 9783319165981, Pages: 267.
10. Uma, V. and V.J. Suseela, 2014. Current Practices in Academic Librarianship. Allied Publishers Pvt. Ltd., India, ISBN-13: 9788184249422, Pages: 268.
11. Lee, Y.C., C. Wang, A.Y. Zomaya and B.B. Zhou, 2010. Profit-driven service request scheduling in clouds. Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, May 17-20, 2010, Melbourne, Australia, pp: 15-24.
12. Popovici, F.I. and J. Wilkes, 2005. Profitable services in an uncertain world. Proceedings of the ACM/IEEE Conference on Supercomputing, November 12-18, 2005, Seattle, WA., USA., pp: 36.
13. Sahal, R., M.H. Khafagy and F.A. Omara, 2016. A survey on SLA management for cloud computing and cloud-hosted big data analytic applications. Int. J. Database Theory Applic., 9: 107-118.
14. Firdhous, M., S. Hassan and O. Ghazali, 2013. Monitoring, tracking and quantification of quality of service in cloud computing. Int. J. Scient. Eng. Res., 4: 112-117.
15. Cao, J., K. Hwang, K. Li and A.Y. Zomaya, 2013. Optimal multiserver configuration for profit maximization in cloud computing. IEEE Trans. Parallel Distrib. Syst., 24: 1087-1096.
16. Dutta, S., M.J. Zbaracki and M. Bergen, 2003. Pricing process as a capability: A resource-based perspective. Strategic Manage. J., 24: 615-630.
17. Weinhardt, C., A. Anandasivam, B. Blau, N. Borissov, T. Meinl, W. Michalk and J. Stoβer, 2009. Cloud computing-a classification, business models and research directions. Bus. Inform. Syst. Eng., 1: 391-399.
18. Zhang, L. and D. Ardagna, 2004. SLA based profit optimization in autonomic computing systems. Proceedings of the 2nd International Conference on Service Oriented Computing, November 15-19, 2004, New York, NY, USA., pp: 173-182.
19. 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.
20. Narman, H.S., M.S. Hossain and M. Atiquzzaman, 2014. h-DDSS: Heterogeneous dynamic dedicated servers scheduling in cloud computing. Proceedings of the IEEE International Conference on Communications, June 10-14, 2014, Sydney, NSW., Australia, pp: 3475-3480.

21. Suri, P.K. and M. Sumit, 2012. A comparative study of various computing processing environments: A review. Int. J. Comput. Sci. Inform. Technol., 3: 5215-5218.
22. 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.
23. Kleinrock, L., 1976. Queueing Systems, Volume 2: Computer Applications. John Wiley and Sons Inc., New York, USA., ISBN-13: 978-0471491118, Pages: 576.
24. Wierman, A., 2007. Scheduling for today's computer systems: Bridging theory and practice. Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA., USA.
25. Rai, I.A. and M. Okopa, 2011. Modeling and evaluation of swap scheduling policy under varying job size distributions. Proceedings of the 10th International Conference on Networks, January 23-28, 2011, St. Maarten, The Netherlands Antilles, pp: 115-120.
26. Wang, W., P. Zhang, T. Lan and V. Aggarwal, 2012. Datacenter net profit optimization with individual job deadlines. Proceedings of the 46th Annual Conference on Information Sciences and Systems, March 21-23, 2012, Princeton, NJ., USA.
27. 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.
28. Crago, S.P. and J.P. Walters, 2015. Heterogeneous cloud computing: The way forward. Computer, 48: 59-61.
29. Anselmi, J., U. Ayesta and A. Wierman, 2011. Competition yields efficiency in load balancing games. Perform. Eval., 68: 986-1001.
30. Niyato, D., E. Hossain and Z. Han, 2009. Dynamics of multiple-seller and multiple-buyer spectrum trading in cognitive radio networks: A game-theoretic modeling approach. IEEE Trans. Mobile Comput., 8: 1009-1022.
31. Nan, G., Z. Mao, M. Yu, M. Li, H. Wang and Y. Zhang, 2014. Stackelberg game for bandwidth allocation in cloud-based wireless live-streaming social networks. IEEE Syst. J., 8: 256-267.