Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/130213

A Novel Analysis of the Performance of Cloud Computing Centers using Non-Markovian Queuing Model

D. Chitra devi[1], *K. Gokulnath*[2]* *and V. Rhymend Uthariaraj*[1]

[1] Anna University, Chennai, 600 025, India
[2] Lovely Professsional University, Phagwara, Punjab, India

Received: 17 Sep. 2018, Revised: 2 Nov. 2018, Accepted: 15 Nov. 2018
Published online: 1 Mar. 2019

Abstract: A novel mathematical model is presented to analyse the performance of cloud computing centers using non-Markovian queuing model M/G/c/c+r/PR with priority as a queue discipline.Task arrivals are categorized as higher-priority ($H_{priority}$) task queue and low-priority ($L_{priority}$) task queue.This model allows cloud providers to determine the performance metrics of the task queue such as average number of tasks, probability of congestion, and probability of no waiting time with respect to server and buffer size. Two feasible conditions, namely non-preemptive and preemptive task priorities are discussed in this work. This approach is validated using discrete-event simulator and Maplesoft for the above mentioned parameters and the analysis has been done based on the arrived results.

Keywords: Cloud Computing, Scheduling, Load Balancing, Priority, Non-Preemptive, Preemptive

1 INTRODUCTION

Cloud computing (CC) is one among the parallel and distributed computing system which is widely-used these days [1].Cloud computing facilitates dynamically scalable and virtualized resources through the internet. Three broad categories of CC are Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a service (SaaS) [2].There are ample advantages of cloud computing, but at the same time service users and providers of Cloud computing faces many challenges.

A task is submitted to the cloud system to get a service. At the time of task submission, there is Service Level Agreement s (SLA's) between cloud users and providers. This SLA's defines the expectation of the cloud user and QoS. Some of the common issues addressed in SLA are security, reliability, power efficiency, availability and so on. The performance metrics are also mentioned in terms of Quality of Service (QoS) such as average waiting time, probability of congestion, probability of no waiting time and so on.

Cloud providers face with many problems while sizing their cloud system in order to meet the SLA's. Hence, cloud providers should choose appropriate tool to assist in providing good service to cloud users. Queuing theory tool is used in this paper for determining the performance metrics such as average waiting time, probability of congestion, probability of no waiting time and to achieve QoS. There are two broad categories in queuing approach, namely Markovian queuing approach and non-Markovian queuing approach. The service pattern is not predictable in cloud system and thereby it should follow the general distribution [3]. Hence the cloud system is modelled as a non-Markovian queuing model.

A user may request a particular infrastructure (or) a particular platform (or) a particular platform (or) particular software with unique probabilities. Assuming the service time of arrived tasks for each and every component of the resultant Platform-Infrastructure-Software follows an exponential distribution/Erlang distribution, the total service time of the arrived tasks in the cloud center follows a hyper-Erlang distribution /hyper-exponential distribution. In such a case, the relative standard deviation ($RSD = \frac{Standarad\ \ deviation}{Mean\ \ value}$) of the resultant task distribution of service time exceeds one [4]. From here it is arrived with a result that the task service time should be

* Corresponding author e-mail: gokulnath.22733@lpu.co.in

modelled with a general distribution, relatively one that permits the RSD to be changed independently of the average (mean) value.

Therefore, in this proposed work, cloud center is developed in the form of M/G/c/c+r/PR queuing model with characteristics of single task arrival and a finite capacity task buffer. Performance is evaluated using Laplace transform -based mathematical model and Markov Chain (MC) is used to obtain task count. This work is aimed to obtain the performance metrics of the task queue such as average number of tasks, probability of congestion, and probability of no waiting time with respect to server and buffer size. This approach is validated using discrete event simulator [5] and Maplesoft [6] for the above mentioned parameters.

Such non-Markovian queuing model can be analysed by means of stochastic process. Initially, a continuous time process is defined, which records the number of tasks in the cloud system. Since the proposed model is not a Markovian process, it is necessary to employ the known embedded Markov Chain process for analysing the cloud system and hence obtained with an approximate preferred performance metrics. This proposed model has the following stages:

- This proposed model considers cloud computing centers with Poisson task arrivals and follows general service distribution of task.
- This model provides probability distribution of the arrived response time of tasks and number of tasks in the cloud system. This also offers average response time, average waiting time, probability of congestion and probability of no waiting time.
- Cloud computing centers performance is dependent on the RSD of the arrived service time of the task and size of the system. Larger values of RSD results in higher response time and low utilization. Hence in this proposed work it is assumed that RSD=0.5 and 1.4.
- Performance can be improved based on the RSD of service time.

Queuing model considered in this work is a non-Markovian queuing with priority as a queue discipline. Basically, there are two feasible priority conditions, namely

- Non-Preemptive task priority
- Preemptive task priority

These priority conditions can be used based on the application. To make the Scheduling mechanism efficient, it is necessary to allocate and distribute the tasks to various resources without compromising the Quality of Service (QoS). Moreover, it is mandatory to schedule the task in an optimized and workload that should be balanced based on two priority conditions. These two conditions are used in different situations based on the application. Now from the previous research work of the authors, namely non-preemptive task priority (Improved-Weighted Round Robin algorithm) [7] and Preemptive task priority (Preemptive Improved-Weighted Round Robin algorithm) [8], scheduling and load balancing of the arrived tasks can be performed. In cloud computing, resource scheduling and load balancing have attracted many researchers attention. There are many research work under progress and some of the related works has been studied to solve the mechanism-related issues. In [9], performance (evaluation and characterization) of queuing system on cloud was proposed with general arrival patterns and also general service patterns. Queuing theory approach has been considered for performance analysis and validated their approximation using discrete event simulator for QoS metrics like response time of task, probability of congestion and so on. Validation of this work is limited to the accuracy of the approximate solution. This work does not validate the adequacy of the general model itself in a specific cloud System.

In[10], a successful and exact solution used for the determination of probability along with cumulative distribution functions of a task response time was derived, by the assumption that both task inter-arrivals and task service times are distributed exponentially. By means of the response time distribution, the correlation along with the maximum number of customers, the minimum resources for service and the maximum service levels was obtained. Only performance measure analysed here is “Response time”.

Non-Markovian with single server queuing model for the non-preemptive task has been developed in [11]. Moreover, user’s need of resources from the cloud computing providers with differentiated quality of service has been analysed. Analysis and numerical results shows that quality of service has been guaranteed for scheduling user-submitted tasks and also CC service providers yields the highest profit. Priority-queue discipline is considered in this paper. But only Non-pre-emptive priority is discussed here.

Priority queue discipline with non-preemptive task scheduling has been discussed in [12]. In this work, single server queuing model considered with finite capacity having buffer size two. These buffers are considered for two different categories of traffic (real time and non-real time). An efficient algorithm has been developed to compute various performance measures for priority queuing. Preemptive task scheduling has not been discussed in this work.

Single-server queue system has been modelled for real-time soft system and its performance is approximated in In[13]. Non-preemptive earliest-deadline first is considered as the service discipline for single-server Markovian queuing model. In this work, there is no time limit for the task to end the service. Loss rate is the key parameter considered in this work for “n” of tasks in the cloud system.

In [14,15,16], various Markovian queuing model techniques were proposed. Various performance parameters like allocation of resources, and task scheduling for cloud computing centers are evaluated by introducing Markovian queuing model to satisfy the demands of the users and performing the service effectively. Response time is the only parameter that has been analysed.

In [17], markovian queuing system is modelled with multiple services in which arrival process follows markovian task arrival and service time follows phase type distribution. Different phases considered in this queuing system are balking of tasks, intolerance and so on. Such queuing system suits only for modelling wireless communication networks.

Non-markovian queuing model is proposed for heterogeneous type of distributed system in [18]. Reallocation policies are proposed for dynamic task scheduling to reach the optimal result for heterogeneous system. Main advantage of the proposed model is to identify the excess load servers and transferring the task to other servers for effective service. No data dependency considered in this work. This model works only for independent tasks.

In [19], M/M/1 queuing system for non-preemptive tasks was proposed with priority scheme. In [20], multiple server queuing system for non-preemptive tasks with priority scheme was proposed for heterogeneous systems. NS-2 simulator was used for experimental analysis. These two approaches suits only for non-pre-emptive task priority.

Priority queue discipline model is discussed in [21] by using fuzzy theory. Here, single-server queuing system is assumed for scheduling the arrived task. Moreover, both arrivals and tasks service follows Poisson distribution.

From the above mentioned research work, it is clearly identified that, the mathematical model for cloud centers, addressing of queue performance, priority queue discipline, namely non-preemptive task priority, preemptive task priority, task scheduling and load balancing has been addressed and handled independently. And also it is identified that only few works have addressed the non-Markovian Queuing system and very few among these adopted the analytical approach. Moreover, analytical model and task performance analysis have not been done for cloud-computing environment using non-Markovian queuing system with priority as queue discipline. This proposed work is aimed towards simultaneous addressing of all the above mentioned aspects and worked towards the development of non-Markovian queuing model with priority as a queue discipline. In this proposed work, mathematical model are verified through discrete event simulator and Maplesoft.

Further this paper is organized as follows. In Section 2, Diagrammatic representation of scheduling and load balancing using non-Markovian queuing model and mathematical model are discussed in detailed manner and arrived with different performance metrics by solving the mathematical model. Mathematical and simulation results are discussed in Section 3. Derived results and future work are summarized in Section 4.

2 QUEUING APPROACH TO IMPROVE QUEUEING PERFORMANCEl

Queuing approach is proposed in this work for the analysis of the performance of arrived task queue and aimed to achieve better quality of service. This approach can be categorized as Markovian queuing approach and non-Markovian queuing approach. From the literature survey it is clear that Markovian approach is not suitable for the analysis of cloud centers. Hence in this proposed work, cloud computing centers are modelled as non-Markovian queue system with priority as queue discipline. Hence Mathematical model is derived for non-Markovian queuing approach and verified the proposed work is verified through simulation. In the next section, diagrammatic representation of task flow submitted by users is explained briefly.

2.1 Task Flow

Fig.1 represents the flow of task arrivals submitted by cloud users and how they are served in optimized manner. Initially, task arrivals are submitted to task manager for getting service. Arrived tasks are categorized as higher-priority ($H_{priority}$) task queue and low-priority ($L_{priority}$) task queue, the tasks are queued separately. Here arrived tasks follow Poisson distribution and tasks are submitted to scheduler and load balancer module. Improved-Weighted Round Robin (IWRR) for Non-Preemptive and Pre-emptive Improved-Weighted Round Robin (PIWRR) for Preemptive priorities are used for scheduling and load balancing. Arrived tasks are assigned to the hosts (servers) in a balanced way based on their priorities. Service pattern follows the general distribution with finite buffer size and finally the departure of tasks takes place.

In the next section, mathematical model for non-Markovian queuing is discussed in a detailed manner and arrived with different performance metrics by solving the mathematical model.

2.2 Mathematical Model

A non-Markovian (M/G/C/C+r/PR) queuing model is proposed for comprising a multi-server queuing system, Poisson arrival pattern and a service pattern which follows the general distribution. In this model, it is assumed that the tasks are scheduled based on priority with independent service and inter arrival times. Arrival rate is denoted by λ and it does't change with respect to

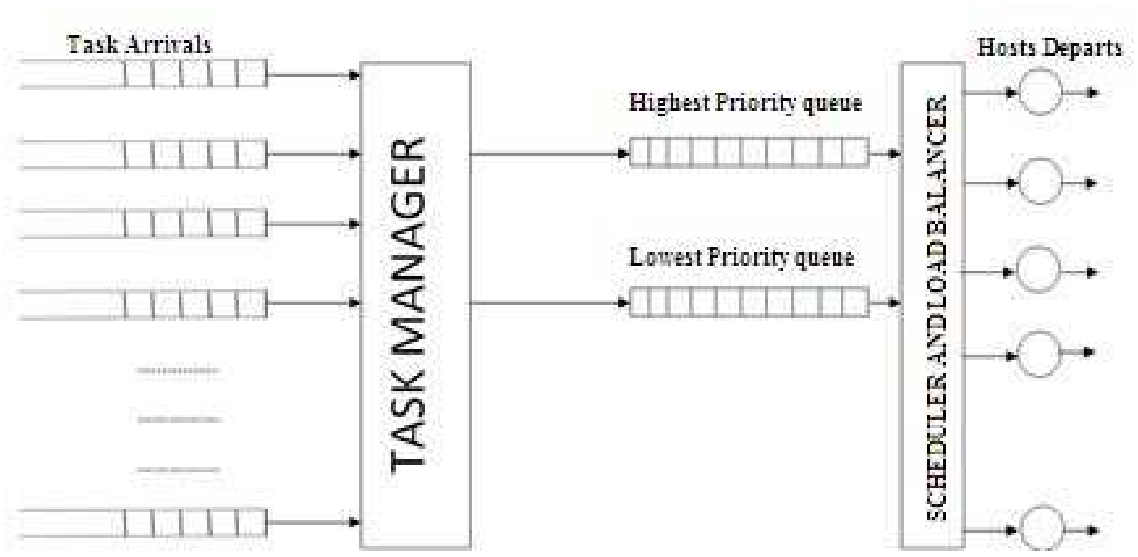

Fig. 1: Task Flow

time (ie)., Input process is independent of time. The general task distribution $\mu = \frac{1}{E(S)}$ is followed, where as S is a random service time. Service times are independent which follows identical and indefinite general distribution for accomplishing the tasks.

A transform is defined as a drawing of a function from one place to another. In this work we are using Laplace transform for the analysis of non-Markovian queuing models. Let us define a real-valued function $f(t)$ in the interval $0 \leqslant t < \infty$ .Laplace transform of $f(t)$ is defined as $F(s) = \int_0^\infty e^{-st} f(t)dt$ where s is assumed as a complex variable. In this model, an arrival of task follows a process which is Poisson and the arrival time between the tasks is distributed exponentially with an arrival rate of $\frac{1}{\lambda}$. Its CDF (Cumulative Distribution Function) of a non-negative random variable (RV) X is defined as $f(t) = \lambda e^{-\lambda t}, t > 0$. Then the LST (Laplace-Stieltjes transform) of arrival time between the tasks is given by

$$F^*(s) = Exp[e^{-sax}] = \int_0^\infty e^{-st} dF(t)$$
$$= \int_0^\infty e^{-st} f(t)dt = \int_0^\infty e^{-st} \lambda e^{-\lambda t} dt$$
$$= \lambda \int_0^\infty e^{-t(\lambda+s)} dt = \frac{\lambda}{\lambda + s} \quad (1)$$

In this model, the general distribution for task service and mean service time is given by $\frac{1}{\mu}$. Its cumulative distribution function is given by $G(t)$ and its pdf is given by $g(t)$ and hence the LST (Laplace-Stieltjes transform) of task's service time $G^*(s)$ is given by

$$G^*(s) = \int_0^\infty e^{-st} dG(t)$$
$$= \int_0^\infty e^{-st} g(t)dt \quad (2)$$

Residual service time G_+ is defined as the time interval, since an arbitrary point of a task's service time to the ending of the task's service time. Elapsed service time of task G_- is defined as the time interval, since the start of a task's service time to an arbitrary point of the task service time. Now the residue task service time G_+ and elapsed service time of task G_- then have the identical probability distribution [22] then the LST (Laplace-Stieltjes transform) is given by

$$G_+^*(s) = G_-^*(s) = \frac{1 - G^*(s)}{s.\frac{1}{\mu}} \quad (3)$$

The traffic intensity for multi-server model is defined as $\frac{\rho}{m\mu} = \frac{mean\ \ service\ \ time}{Number\ \ of\ \ servers\ \ \times mean\ \ inter\ \ arrival\ \ time}$ where mean service time is denoted by $\frac{1}{\mu}$ and λ is the reciprocal of the mean inter-arrival time.

In the proposed work,the load of the cloud system M/G/C/C+r is denoted by ρ and is hence assumed that the traffic intensity $\rho < 1$ for steady-state results to exist. Because if $\rho > 1$ (i.e) $\lambda > c\mu$ then the system is overloaded and if the buffer size is full during the arrival of task, then the newly-arrived task gets blocked and vanished.If $\lambda = c\mu$ ($\rho = 1$) then the arrival of tasks and service of tasks are deterministic and moreover tasks are completely scheduled, steady state does not exists. Hence in our model we assume that $\rho \equiv \frac{\lambda}{c\mu} < 1$ (i.e) $\lambda < c\mu$.

2.3 Non-Markovian Queues- Embedded Markov chain (EMC)

In Markovian queuing systems, both arrival times and task's service times follows an exponential distributions. The number of tasks $N(t)$ at any time't' in the queuing system represents the state of the systems.

When more general service times are allowed, then $N(t)$ and also the remaining (or residual) service time $R(t)$ for the current task is needed to predict future values of $N(t)$. In such cases we construct a Markov Chain (MC) out of $N(t)$ and the markov chain is named as "embedded", which permits us to construct many of the parameters of interest. Let the tasks arrivals follows Poisson process with λ as a task arrival rate. Task-service times are independent and identical distribution of RV's (random variables) with general probability distribution of task's service time. Between two departures of task's service time T, the Probability distribution of T is given by $g(t)$. From the ergodicity definition and known result "MC is ergodic" it is proved that EMC is homogeneous and also ergodic. Hence the solution of EMC is in steady state. Then average response time and task count in the cloud system can be calculated.

At $t \geq 0, N(t_n)$ is defined as the no. of tasks in the cloud system . Let t_n be the time instant at which the $n^t h$ task, number of task remaining in the cloud system is defined as $X_n = N(t_n), n = 1,2,\ldots$ and the Markov chain is defined as a sequence of RV's $\{X_n; n = 1,2,\ldots\}$. Hence

$$X_{n+1} = \begin{cases} X_n - 1 + A, if \;\; X_n > 0 \;\; (i.e) \;\; X_n \geq 1 \\ A, if \;\; X_n = 0 \end{cases} \tag{4}$$

Where A denotes the number of tasks arriving during $(n+1)^{th}$ task's service time T.

The first condition for X_{n+1} is obvious. The second condition outcomes from the reality that T_{n+1} is the point of departure of the task which arrives after the task t_n. It is in reality $X_{n+1} = 1 - 1 + A$. Let $U(X_n) = \begin{cases} 1, X_n > 0 \\ 0, X_n = 0 \end{cases}$ where U - "Unit step function". Hence $X_{n+1} = X_n - U(X_n) + A$. To justify that $X_1, X_2, \ldots, X_n$. is a Markov Chain, it is necessary to prove that an upcoming state of the Markov chain (MC) depends simply on the current state. (i.e) it is necessary to prove that given current state X_n, the upcoming state X_{n+1} is independent of prior states $X_{n-1}, X_{n-2}, \ldots$ From (1) it's clear that X_{n+1} depends only on X_n and A. Moreover A is independent of the prior states $X_{n-1}, X_{n-2}, \ldots$ then it is proved that $X_1, X_2, \ldots, X_n$ is a MC (Markov Chain). This is because of "A" Where A denotes the number of tasks arriving during $(n+1)^{th}$ task's service time T, which depends on the service time with respect to task length, but this service time does not depend on the events that have occurred previously (i.e)., the size of the queue at prior departure points are $X_{n-1}, X_{n-2}, \ldots$. Hence $X_1, X_2, \ldots$ is an embedded MC.

Next, we develop transition probabilities on behalf of the above derived embedded MC (Markov chain). Let us assume $[P[i,j]]^{(n)} \equiv Pro[X_n = j/X_0 = i], i,j \in S$ where S is the state space and $S = \{0,1,2,\ldots\}$ is the number of tasks in the cloud system. Let $[P[i,j]]^{(i)} = Pro[i,j]$ from the relationship of X_n and X_{n+1}, then

$$Pro[i,j] \equiv Pro[X_{n+1} = j/X_n = i]$$

$$= \begin{cases} Pro(i + A - 1 = j), i > 0 \\ Pro(A = j), i = 0 \end{cases}$$

$$= \begin{cases} k_{(j-i+1)}, i > 0 \\ k_{(j)}, i = 0 \end{cases} \tag{5}$$

$whereK_{(j)} = Pro(X_n = j)$

A balanced probability distribution exists for the no. of tasks exhibit at the task arrival time. Assuming that steady-state probability vector $\pi P = \pi$ where the elements of the matrix P are defined as one step TPM (Transition probability matrix P_{ij}) and $\pi = \{\pi_0, \pi_1, \ldots, \pi_{c+r}\}$. Now the stationary probability for the non-Markovian model M/G/C can be constantly written in the form of waiting time-cumulative distribution function as

$$\pi_n^q = Pro['n' \;\; in \;\; queue \;\; just \;\; after \;\; a \;\; departure]$$

$$= \frac{1}{n!} \int_0^{c+r} (\lambda t)^n e^{-\lambda t} dw_q(t) \tag{6}$$

Where "n" is the cloud system size, for the "n" Poisson arrivals at the departure point arbitrarily during the waiting time of the cloud system departure. At the departure points, the average queue length $L_q^{(D)}$ is defined as

$$L_q^{(D)} = \sum_{n=0}^{c+r} n\pi_n^q = \int_0^{c+r} (\lambda t) dw_q(t) = \lambda w_q \tag{7}$$

Which is Little's formula.

In general k^{th} factorial moment queue length of the departure point is $L_q^{(D)}(k) = \lambda^k w_{(q,k)}$ where the ordinary waiting time of k^{th} moment in the queue is denoted by $w_{(q,k)}$. Next we identify the elements of the TPM (Transition probability matrix), it is necessary to calculate the no. of tasks leaving from the cloud system in between the consecutive arrival of the tasks. Each one of the host (server) has nil or more task departures during the inter-arrival time. For a arrived task to end and depart from the cloud system between 2 successive tasks arrivals "the residual of task service time must be smaller than the task inter arrival time by the probability".

$$\begin{aligned}Pro(x) &= Pro[F(s) > G_+(s)] \\ &= \int_{T=0}^{\infty} Pro[L[f[t]] > G_+(s)/G_+(s) = T]dG_+(T) \\ &= \int_{t=0}^{\infty}(\int_{T=t}^{\infty} \lambda e^{-\lambda t}dt)dG_+(T) \\ &= \int_{0}^{\infty} e^{-\lambda t}dG_+(T) = G_+^*(\lambda) \quad (8)\end{aligned}$$

The length of the queue size is zero (idle server), then an arriving task can accommodate immediately. The probability $Pro(y)$ for such an arrived task leaves prior to the next arrival of the task which is given by

$$\begin{aligned}Pro(y) &= Pro[F(s) > G(s)] \\ &= \int_{T=0}^{\infty} Pro[L[f[t]] > G(s)/G(s) = T]dG(T) \\ &= \int_{t=0}^{\infty}(\int_{T=t}^{\infty} \lambda e^{-\lambda t}dt)dG(T) \\ &= \int_{0}^{\infty} e^{-\lambda t}dG(T) = G^*(\lambda) \quad (9)\end{aligned}$$

Using the above values, TPM (Transition probability matrix) can be calculated.

2.4 Transition Probability Matrix

To calculate the TPM, it is necessary to categorize EMC (Embedded Markov-Chain) among 4 different operating cases as follows:

Case (i): From the relationship of X_n and X_{n+1} we have $Pro(i,j) \equiv Pro[(X_{(n+1)} = j)/(X_n = i)]$. Now the probability that $i+1-j$ tasks is serviced during the inter-arrival time of tasks. Here for $(i+1) < j, Pro[i,j] = 0$, because we have at most $i+1$ tasks existing between X_n and $X_{(n+1)}$.

Case (ii): $i < c$ and $j \leq c$,in the task queue none of the tasks arrived are waiting. Now "c" denotes the number of hosts (servers) between inter-arrival of tasks request; the sum of $(i+1-j)$ task finishes their service and leave from the cloud system by means of the probability $Pro[i,j]$

$$= \binom{i}{i-j}[Pro(x)]^{i-j}(1-Pro(x))^{j}Pro(y)$$
$$+ \binom{i}{i+1-j}[Pro(x)]^{i+1-j}(1-Pro(x))^{j-1}(1-Pro(y) \quad (10)$$

Case (iii): For $i, j \geq c$, all hosts (servers) are busy during the two consecutive arrivals of the request. Let us assume $(i+1-j) = Z$ which denotes the number of tasks arrived or leave from the cloud system among Markovian points. This point might lies between 0 to ∞ (its close to 1). For this case, state-transition probabilities are given by$Pro[i,j] =$

$$\sum_{s_1=min(z,1)}^{Min(z,c)} \binom{c}{s_1}(Pro(x))^{s_1}(1-Pro(x))^{c-s_1}$$
$$\sum_{s_2=min(z-s_1,1)}^{Min(z-s_1,s_1)} \binom{s_1}{s_2} Pro(u,2)^{s_2}(1-Pro(u,2))^{s_1-s_2}$$
$$\binom{s_2}{z-s_1-s_2}[Pro(u,3)]^{z-s_1-s_2}(1-Pro(u,3))^{s_2} \quad (11)$$

Case (iv): For $i \geq c$ and $j < c$, all hosts (servers) are busy at the initial arrival of task and in the queue $(i-c)$ arrived tasks are being waiting. During the next task arrival, the queue is vacant and specifically "j" arrived tasks are in service. For this case, state transition probability is given by $Pro[i,j] =$

$$\sum_{s_1=(c-j)}^{Min(z,c)} \binom{c}{s_1}(Pro(x))^{s_1}(1-Pro(x))^{c-s_1}$$
$$\sum_{s_2=min(z-s_1,c-j)}^{Min(z-s_1,s_1)} \binom{s_1}{s_2} Pro(z,2)^{s_2}(1-Pro(z,2))^{s_1-s_2}$$
$$\binom{s_2}{z-s_1-s_2}[Pro(z,3)]^{z-s_1-s_2}(1-Pro(z,3))^{s_2} \quad (12)$$

For case (iii) and case (iv) it is assumed that there are only up to 3 task departures between inter-arrival tasks. But steady-state distribution $\{\pi_i\}$ is to be present for the Markov chain, then the solution of the cloud system $\forall j \geq 0$ is given by

$$\sum_{i=0}^{c+r} P(i,j)\pi_i = \pi_j \ \ and \ \ \sum_{i=0}^{c+r} \pi_i = 1 \quad (13)$$

Where π_j denotes the balance equation which links the entering and leaving state probabilities. Now the generating function for the no. of arrived tasks in the cloud system at the point of a task arrival is defined as

$$\pi(z) = \sum_{j=0}^{c+r} \pi_j z^j \quad (14)$$

2.5 Priority-Based Queue Discipline

In priority queue discipline model tasks with the $H_{priority}$ are chosen for task service in advance of those with $L_{priority}$, which is independent of the arrived task with respect to time in the cloud system. This can be further enhanced into two feasible priority conditions, namely non-preemptive and preemptive cases. In the case of non-preemptive, the task is not interrupted which is in service and at the same time the $H_{priority}$ tasks immediately go to the top of the queue and wait for its turn of service. In the case of preemptive, when the $H_{priority}$ task arrived the cloud system, it is permitted to enter the service directly even though another task with $L_{priority}$ is in service. In this case, it is necessary to decide that whether to resume the service of the task which is preempted or the service of the task have to start from the initial state. A real-life queuing situation contains priority. For optimal design, priorities are absolutely essential. From the above, we have arrived with the 2 common situations if the queue discipline follows a priority queuing. Priority queue discipline is considered in this paper.

Case (i): In the first priority situation, which is called as the Non-preemptive case, the task with the $H_{priority}$ go to the top of the queue but the task does not get into the service till the task complets its service which occurs currently in the service, although the current running task has $L_{priority}$. Suppose the k^{th} priority task (if the number is small then the priority will be high) arrives according to a Poisson process with rate λ_k, $(k = 1,2,\ldots,r)$. Let us assume X_k is a task with priority k where the service time is random. Next, let us assumue X_k with general distribution having first moment of X_k which is given by $Exp[X_k] = \frac{1}{\mu_k}$ and 2nd moment is given by $Exp[X_k^2]$.Let us consider a i^{th} priority task which arrives at the cloud system. After the task arrived at the cloud system or queue then upon this new arrival of task we suppose that there aren_1 tasks with priority 1 in the top of the queue, n_2 tasks having the priority 2 and so on. Now the required time to complete the tasks service is given by $S_0$and this time S_0 will be equal to zero if the cloud system is idle.

Now we define $E[S_0] = P[Sytem\ is\ busy]$. $E[S_0/(sytem\ is\ busy\ with\ priority\ k\ task)]$. Here S_0 is the service time which remains for the task arrived, this is true for the idle cloud system if the arriving task has the value zero. If the cloud system is busy then the probability is given by

$$\lambda.[Mean\ service\ time] = \lambda \sum_{k=1}^{r} \frac{\lambda_k}{\lambda}.\frac{1}{\mu_k} = \rho$$

$$E[S_0/system\ is\ busy\ with\ k^{th}\ priority\ task] = \frac{E[X_k^2]\mu_k}{2}$$

Therefore,

$$\begin{aligned} E[S_0] &= \rho \sum_{k=1}^{r} \frac{E[X_k]^2 \mu_k [\frac{\lambda_k}{\mu_k}]}{2\rho} \\ &= \sum_{k=1}^{r} \frac{E[X_k]^2 \lambda_k}{2} \\ &= \frac{\lambda}{2} \sum_{k=1}^{r} \frac{\lambda_k}{\lambda} E[X_k]^2 \\ &= \frac{\lambda E[s^2]}{2} \end{aligned} \tag{15}$$

Hence, waiting time of task in queue is given by

$$W_q^{(i)} = \frac{\lambda E[S^2]/2}{(1-\sigma_{i-1})(1-\sigma_i)} \tag{16}$$

By Little's formulae, Expecting waiting time of task in system is

$$\begin{aligned} W_s^{(i)} &= W_q^{(i)} + \frac{1}{\mu} \\ &= \frac{\lambda E[S^2]/2}{(1-\sigma_{i-1})(1-\sigma_i)} + \frac{1}{\mu} \end{aligned} \tag{17}$$

Length of the task queue (or) mean number of task in queue is given by $L_q^{(i)} = W_q^{(i)} \lambda'$where the parameter is the overall effective arrival rate at the system. It equals the arrival rate λ when all arriving tasks can join the system. Otherwise, if some tasks cannot join because the system is full then $\lambda' < \lambda$.

Therefore,

$$L_q^{(i)} = \frac{\lambda E[S^2]/2}{(1-\sigma_{i-1})(1-\sigma_i)}.\lambda' \tag{18}$$

Mean number of tasks in the cloud system (or) length of the tasks in the cloud system is given by

$$\begin{aligned} L_s^{(i)} &= L_q^{(i)} + \frac{\lambda'}{\mu} \\ &= \frac{\lambda E[S^2]/2}{(1-\sigma_{i-1})(1-\sigma_i)}.\lambda' + \frac{\lambda'}{\mu} \end{aligned} \tag{19}$$

Now we define $\overline{C}$ the average number of busy servers as

$$\overline{C} = L_s^{(i)} - L_q^{(i)} \tag{20}$$

Hence facility utilization is given by $\frac{\overline{C}}{C}$ where "C" is the number of hosts (servers).

Case (ii): Next priority situation, called the preemptive case. In this situation, if the $H_{priority}$ task arriving the cloud system as soon as the arrival $H_{priority}$ task is permitted to enter service directly even though another task with $L_{priority}$ is in service. In this case, task with $L_{priority}$ is preempted from service; service of the $L_{priority}$ task is stopped from the current state and resumed service from the current state after the $H_{priority}$ task is being served. From this we can come to the conclusion that the preempted task which is resumed from the service either continues the service of the task from the preempted point or else starts from the initial stage. In this model, we assume that the task service continues their service from the preempted point. Here the preempted task resumes their service from the interrupted point. Then the expected or mean number of tasks in the system is given by

$$L_s^{(i)} = \frac{\rho_i}{(1-\sigma_{i-1})} + \frac{\lambda_i \sum_{j=1}^{i} \lambda_j E[S_j^2]}{2(1-\sigma_{i-1})(1-\sigma_i)} \quad (21)$$

S_j - Service time which is random.

$$\rho_i = \lambda_i E[S_i] \;\; and \;\; \sigma_i = \sum_{j=1}^{i} \rho_i$$

Hence we can calculate the remaining parameter by using Little's formulae.

Here, the task arrivals have independent buffer state. Moreover, we have mean no. of tasks in the cloud system. Then the Probability of Congestion cloud system $Pr_{Congestion}$ can be evaluated as $Pr_{Congestion} = \pi_{c+r}$ here the size of the buffer is given by r. In this model the task will continuously arrives to the system even if its capacity is full. Since there is a buffer, it is not difficult to take care of the case if the system is blocked. Here probability of congestion is always below a threshold value ε.

The probability of no waiting time (probability of immediate service for the arrived tasks) is given by

$$Pro_{(no \;\; wait)} = \sum_{i=0}^{c-1} \pi_i \quad (22)$$

3 EXPERIMENTAL AND PERFORMANCE ANALYSIS

Discrete-event simulator is used to validate an analytical model (Petri net-based simulation engine) and the balance equations derived in this model is solved numerically using Maple 16 (Maplesoft). The above derived parameter values give a practical approach to the performance of cloud centers. Practically in a large service provider, if the number of hosts (servers) is very low then the traffic intensity will be high.

It is assumed that service time has gamma distribution. This distribution is selected because it permits Relative Standard Deviation (RSD) to be assigned independent of the average value. Hence the values used for RSD: low-value $RSD = 0.5$, which results in hyper-exponential service time and high value $RSD = 1.4$, which results in hyper-exponential service time.

The following parameter values are considered for the analysis purpose. The values assumed may be applicable to all types of service providers (medium to high) for maximum server utilization as much as possible.

- Number of Hosts : $c = 100, 200, 500$
- System capacity (Input buffer) :

$$r = 0 \;\; to \;\; \frac{m}{2} \;\; (5 \;\; steps)$$

- Relative Standard Deviation (RSD) of task service time : 0.5 and 1.4
- Traffic intensity (server busy time) :$\rho = 0.85$

First illustration is shown in Fig. (2,3 and 4) the mean number of tasks increases gradually with the size of the buffer when the no. of host is c=100, however for higher number of servers ($c = 200$ and 500) it is less distinct. In the second illustration, the system capacity is examined which is shown in Fig. (5,6 and 7), by increasing the buffer size it is clear that probability of congestion decreases rapidly. From this illustration it is clear that, probability of congestion is maintained below 0.5% ($Pr_{Block} \leq 0.005$) . Moreover it is observed that, with less arrival variability ($cv = 0.5$), the probability of congestion is decreasing.

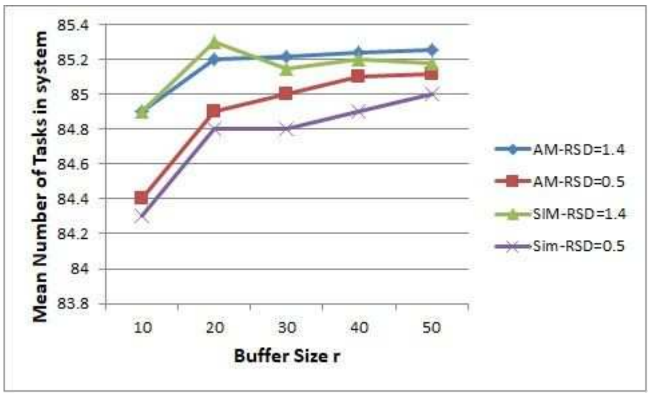

Fig. 2: Mean number of tasks with $c = 100$ at $\rho = 0.85$

Finally in the third illustration, immediate probability of service is examined without waiting in queue and shown in Fig.(8,9 and 10). Probability of no waiting time should not depend on the system capacity; at least one idle server is required for arriving task. It is clearly observed that if the buffer size is small, the probability is closer to 1. This probability stabilizes if the buffer size is increased by r. From this behaviour, it has been concluded that the arriving tasks will not be queued if the buffer size r is small, instead the arriving tasks get service immediately or the arriving tasks are blocked. Hence, both the congestion probability and probability of no wait will be decreased by increasing the buffer size. It is quite

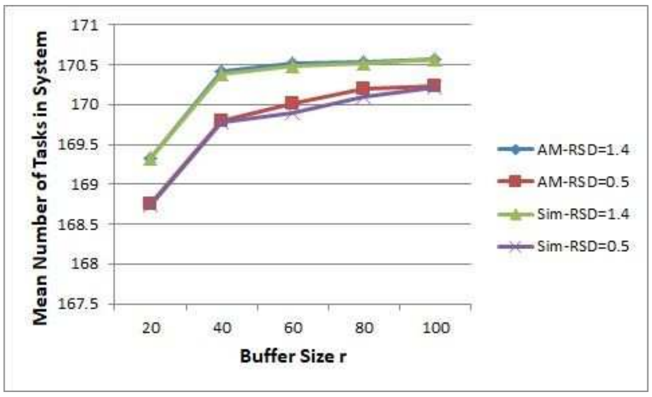

Fig. 3: Mean number of tasks with $c = 200$ at $\rho = 0.85$

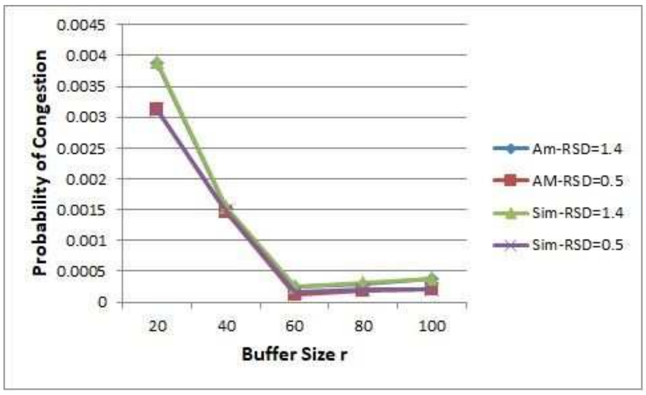

Fig. 6: Probability of congestion with $c = 200$

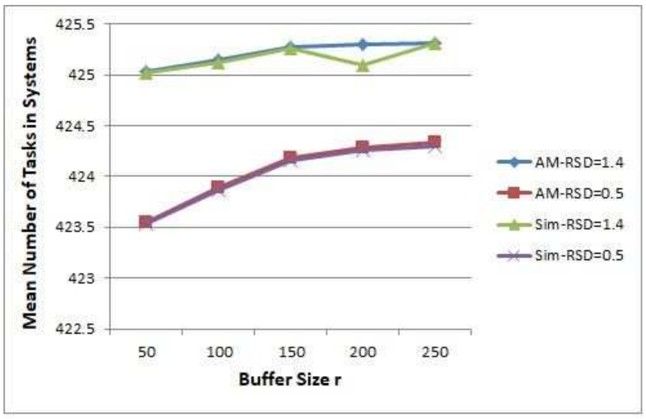

Fig. 4: Mean number of tasks with $c = 500$ at $\rho = 0.85$

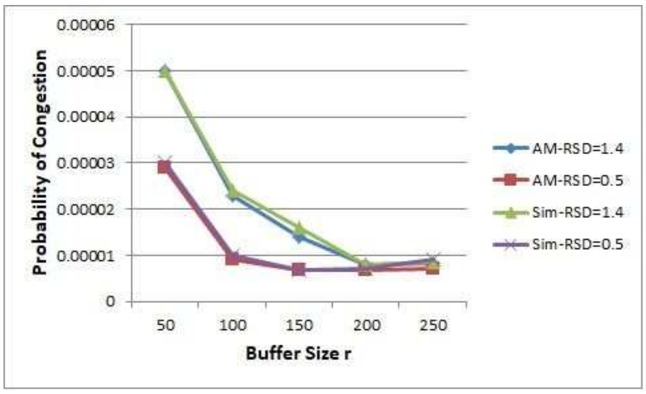

Fig. 7: Probability of congestion with $c = 500$

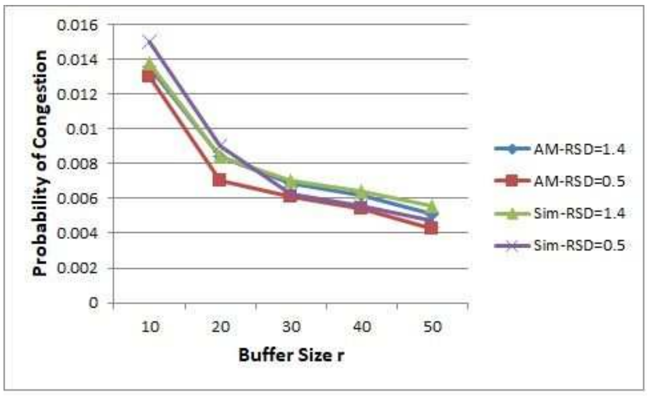

Fig. 5: Probability of congestion with $c = 100$

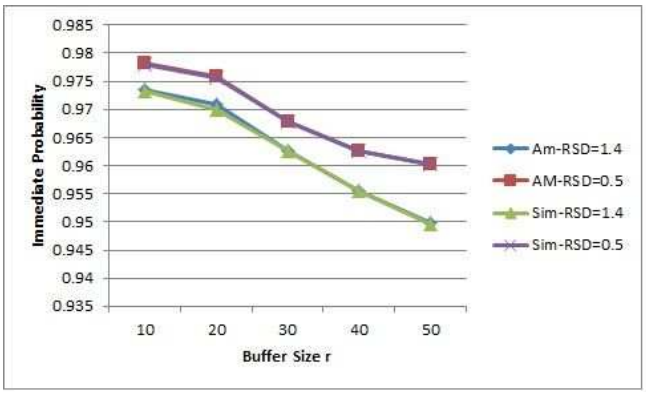

Fig. 8: Immediate probability with $c = 100$

obvious that 100% of tasks do not get immediate service, hence it should be aimed that at least 85%, or above, tasks to get immediate service without waiting in queue (i.e) $Pr_{no\ \ wait} \geq 0.85$. Figure 4 shows that the analysis for host size $c = 100, 200, 500$ and set buffer space equal to $c + r$. Hence from the overall analysis it is clearly identified that performance is poor when the $RSD = 1.4$. As a future work, to obtain further understanding into the performance of cloud centers, it is also necessary to find higher moments like standard deviation, skewness and kurtosis of response time. From the higher moments,

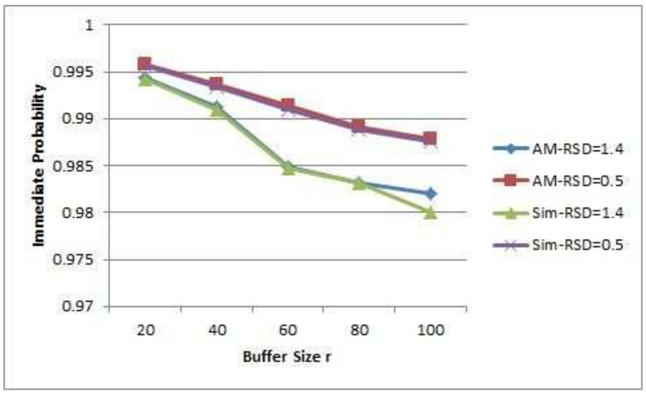

Fig. 9: Immediate probability with $c = 200$

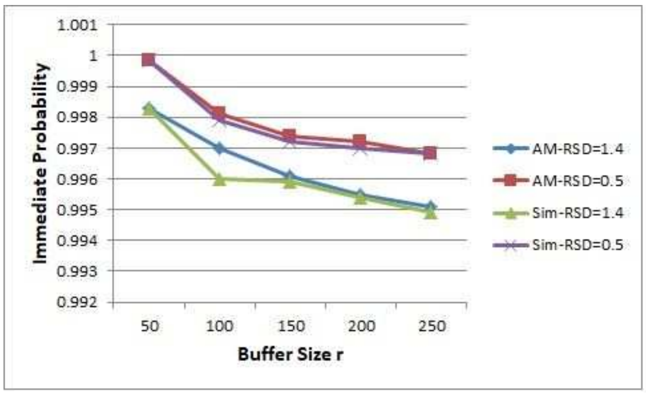

Fig. 10: Immediate probability with $c = 500$

performance of response time can be analysed. Analytical model is proposed in this paper based on Markov chain approximation for performance analysis of task queue in cloud computing centers. Further in this paper, numerical analysis and simulation are conducted to validate the proposed model. Simulation and numerical results prove that the approximate method which has been proposed in this paper ends with high amount of accurateness for the average no. of tasks in the cloud systems, probability of congestion and immediate probability.

4 CONCLUSION AND FUTURE ENHANCEMENT

Mechanism-related issues in cloud computing are scheduling the arrived tasks to the resources and balancing the load in an optimized way. It is necessary to reduce waiting time and response time of the task submitted by the cloud users. In this work, the performance of the task queue is analysed using non-Markovian queuing model based on Markov chain model. Due to the inconsistency of workloads in cloud environment it is important to consider Markovian arrival of task and general service time distribution with priority as a queue discipline. Numerical analysis has been done using Maplesoft and simulated using discrete event simulator for validating the proposed analytical model. Simulation and numerical analysis show that the proposed model provides accurate results for the Expected (mean) number of tasks, probability of congestion, and immediate probability with high amount of exactness. Improved Weighted Round Robin algorithm with non-preemptive and pre-emptive Improved Weighted Round Robin algorithm with Preemptive priorities is used for scheduling and load balancing of task and experimental analysis is done by using the simulator Cloudsim. Further research can be done for evaluating time complexity of non-markovian queuing model for cloud environment.

References

[1] Chia-WeiLee, Hung-ChangHsiao, Kuang-YuHsieh and Sun-YuanHsieh, A Dynamic Data Placement Strategy for Hadoop in Heterogeneous Environments, Big Data Research, Elsevier, 14-22 (2014).
[2] Armando Escalante and Borko Furht, Cloud Computing Fundamentals, Handbook of Cloud Computing, Springer, 3-19 (2010).
[3] Qiang Duan, Cloud Service Performance Evaluation: Status, Challenges, and Opportunities – A Survey from the System Modeling Perspective, Digital Communications and Networks, 1-36 (2016).
[4] Anum L. Enlil Corral-Ruiz, Felipe A. Cruz-Pérez, and Genaro Hernández-Valdez, Teletraffic Model for the Performance Evaluation of Cellular Networks with HyperErlang Distributed Cell Dwell Time, IEEE Conference, (2010).
[5] OptSim, RSoft Design group, (1989).
[6] Maplesoft, Maple 16, (2012).
[7] Chitra Devi, D and Rhymend Uthariaraj, V, Load Balancing In Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Nonpreeemtive Dependent Tasks, The Scientific World Journal, 2016, Article ID 3896065.
[8] Chitra Devi, D and Rhymend Uthariaraj, V, Load Balancing in Cloud Computing for Independent and Dependent Tasks with Pre-emption, Transylvanian Review, XXIV,1662-1683 (2016).
[9] Alexandre Brandwajn, Hind Castel-Taleb,Tulin Atmaca and Thomas Begin, Performance Evaluation of Cloud Computing Centers with General Arrivals and Service, IEEE Transactions on Parallel and Distributed Systems, 2341 – 2348 (2016).
[10] Harry Perros and Kaiqi Xiong, Service Performance and Analysis in Cloud Computing, Congress on Services Part I (SERVICES-1), 2009 IEEE, (2009).
[11] Luqun Li, An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Providers, Third International Conference on Multimedia and Ubiquitous Engineering, 295-299, (2009).

[12] Sashisekaran Thiagarajan and Srinivas R. Chakravarthy, A Stochastic Model for a Hysteresis based Priority Queueing Strategy for ATM Networks with Batch Arrivals – Theory, Proceedings Third IEEE symposium on Computers and Communications, (1998).
[13] Ali Movaghar and Mehdi Kargahi , Non-Preemptive Earliest-Deadline-First Scheduling Policy: A Performance Study, Proceedings of the 13th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, (2005).
[14] Hsiao-Hwa Chen and Xing Liu, Yun Li, Wireless Resource Scheduling based on Backoff for Multi-user Multi-service Mobile Cloud Computing, IEEE Transactions on Vehicular Technology,1-13 (2015).
[15] Zheng Xiao, Zhao Tong, Kenli Li, Probabilistic Scheduling Based on Queueing Model for Multi-user Network Applications, IEEE 12th International Conference on Computer and Information Technology, 224-229 (2012).
[16] Xiaoming Nan, Yifeng He, Ling Guan, Joint Optimization of Resource allocation and workload Scheduling for Cloud based multimedia services, IEEE 18th International workshop on Multimedia signal processing, (2017).
[17] Alexander Dudin, Valentina Klimenok, Retriaal queue of BMAP/PH/N type with customers balking, impatience and non-persistence, IEEE Conference on Future Internet Communications, (2013).
[18] Jorge E. Pezoa, Member, and Majeed M. Hayat, Performance and Reliability of Non-Markovian Heterogeneous Distributed Computing Systems, IEEE Transactions on Parallel and Distributed systems, Vol. 23, 1288-1301 (2012).
[19] Manu K. Gupta and N.Hemachandra, On 2-moment completeness of non-pre-emptive, non-anticipative work conserving scheduling policies in some single class queues, 13th International ymposium on Modelling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 267-274 (2015).
[20] Peng Ke, Wenxiang Li and Yunhe Wu, Research on Multiple Services Scheduling Based on Priority-Queuing Model, IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, 372-376 (2014).
[21] David de la Fuenteb and Marıa Jose Pardoa, Optimizing a priority-discipline queueing model using fuzzy set theory, An International Journal Computers and Mathematics with Applications, Elsevier, 267-281(2007).
[22] H.Takagi, Queueing Analysis: Vacation and Priority Systems, Vol.1, (1991).

CHITRA DEVI D. received the M.E degree in Systems Engineering and Operations Research at Anna University, Chennai and pursuing PhD in Faculty of Information and Communication Engineering, Anna University, Chennai. Her research interests are in the areas of Cloud Computing and applied mathematics . She has published research articles in reputed international journals. She is referee of Iranian Journal of Science and Technology, Transactions A: Science.

GOKULNATH K. is Associate Professor of Computer Science and Engineering at Lovely professional University, Punjab. He received the PhD degree in Faculty of Information and Communication Engineering from Anna University, Chennai. He is referee of several international journals in the frame of Cloud Computing. His main research interests are: Cloud Computing, game theory and optimization theory.He has published research articles in reputed international journal

RHYMEND UTHARIARAJ V. received the PhD degree in Computer Science and Engineering at Anna University, Chennai. He is Professor and Director of Ramanujan Computing centre, Anna University Chennai. His research interests are in the areas of applied mathematics , Networking and Operations Research. He has published research articles in reputed international journals. He is referee of many reputed journals.