Cloud Data Center Design using Delay Tolerant Based Priority Queuing Model

Meera Annamalai[1] and Swamynathan Sankaranarayanan[2]
[1]Department of Information Technology, Tagore Engineering College, India
[2]Department of Information Science and Technology, Anna University Chennai, India

Abstract: *Infrastructure as a Service (IaaS) that occupies the bottom tier in the cloud pyramid is a recently developed technology in cloud computing. Organizations can move their applications to a cloud data center without remodelling it. Cloud providers and consumers need to take into account the performance factors such as resource utilization of computing resources, availability of resources caused by scheduling algorithms. Thus, an effective scheduling algorithm must strive to maximize these performance factors. Designing a cloud data center that schedules computing resources and monitoring their performances plays a leading challenge among the cloud researches. In this paper, we propose a data center design using delay tolerant based priority queuing model for resource provisioning, by paying attention to individual customer attributes. Priority selection process defines how to select the next customer to be served. The system has a priority based task classifier and allocator that accept the customer's request. Based on the rules defined in the rule engine, task classifier classifies each request to a workload Priority classifier is modeled as M/M/S priority queue. The resource monitoring agent provides the resource utilization scenario of cloud infrastructure in the form of dashboard to the task classifier for further resource optimization.*

Keywords: *Cloud data center, (IaaS) and M/M/S priority queuing model.*

Received June 7, 2015; accepted July 28, 2016

1. Introduction

Cloud Computing is the most glorified word among the service industries. It is a recently developed cost and energy efficient computing paradigm that provides computing resources on a pay as you go model with a predefined quality of service. Cloud datacenters host thousands of machines in the high performance computing machines, blade servers or rack servers. It has large number of physical and virtual resources that accept different categories of workloads. Enterprises migrating workloads to public clouds must examine the implications of moving the applications to a public cloud environment. Migrating a critical workload such as core banking application may not be ready to move to a public cloud infrastructure. Thus, enterprises must understand the underlying infrastructure required to host their workloads. End users are guaranteed a QoS to meet service level agreement based on different categories of customer's workload. Based on the literature survey the existing Infrastructure as a Service (IaaS) service providers with the corresponding instance specifications are tabulated in Table 1.

Consider a scenario that any workload that is sent to a cloud data center is serviced by a suitable server and leaves the center upon completion. In this paper, we model the cloud data center as an M/M/S priority queuing system. Queuing models [6] have been proved to be very useful in many practical applications where it is essential to plan capacity of servers, appropriate allocation of task, to analyze the variability of arrival and service facilities.

Table 1. Leading IaaS providers and their datacenter specification.

IaaS cloud provider	Datacenter Specification	Instances specification
Amazon	Amazon EC2 instances	General-purpose instances, Compute-optimized, Memory-optimized, Storage-optimized, Micro-Instances, GPU-Instances.
Rackspace	Performance server 1	RAM - 1 GB to 8 GB, Block Storage 20 GB to 80 GB, vCPUs from 1 to 8, Bandwidth 200 Mp/s to 1600 Mb/s
	Performance server 2	RAM - 40 GB to 120 GB, Block Storage 40 GB to 1200 GB, vCPUs from 4 to 32, Bandwidth 1250 Mp/s to 10000 Mb/s.
GoGrid	Cloud servers	X.Small, Small, Medium, Large, X-Large, XX-Large, XXX-Large
Google	Compute configuration	Launch Linux VMs on-demand. 1, 2, 4 and 8 virtual core VMs are available with 3.75GB RAM per virtual core.
HP	Standard and High memory Instances type	Standard : 1 to 8 vCPU, 1GB to 30 GB RAM and 20 GB to 570 GB Disk, High Memory Instances : 4vCPU, 16 to 60GB RAM 160 GB to 570 GB Disk
Softlayer	Datacenters are SSAE-16 compliance.	196,500 servers deployed.
Opsource	Dimension Data Cloud. Built on VMware's vSphere 5.x hypervisor	Cloud Server can be customised on the fly with up to 16 CPUs, 128 GB of RAM and 2.5 TB of storage.
Lunacloud	Physical infrastructure in Tier 3 + datacenters	Cloud Servers with any choice of RAM from 512 MB to 96 GB, 1 to 8 CPU cores and 10 GB to 2 TB Disk, running Linux or Windows.
Terremark	Vcloud Express	Virtual Processors - 1, 2, 4, 8 Memory 16GB maximum Storage Up to 15 virtual disks (including system disk). Up to 512GB per disk

Due to dynamic behavior of cloud data centers, cloud researchers focus more on capacity planning, dynamic resource allocation and service management. So, queuing model may be used to analyze the performance of cloud service facilities.

The queueing notation was defined by Stallings [20] with a five-part descriptor A/B/m/K/M to specify the system's storage capacity K and the size of the customer population; if either of these last two descriptors is absent, then the model takes on the value of infinity. Another important structural discipline in the queueing system is the queueing discipline [7, 8]. Since, the proposed system categorizes the workload into groups, M/M/S non preemptive priority queuing system for server allocation with 'n' priority classes is considered.

2. Related Work

Performance analysis of cloud data center using queuing theory has been addressed by researchers in some recent works. Performance management on cloud using queuing model proposed by Chen and Li [3] aimed to enforce the scalability property on cloud. Virtual Machines (VMs) are created and removed based on the number of requests waiting for service, expected waiting time of requests and sojourn time of requests specified in Service Level Agreement (SLA). It is observed that scalability of virtual machines based on resource utilization is not paid much attention. Achieving the QoS targets is very critical as well as hard to analyze. Xia *et al*. [21] presented QoS determination of Infrastructure as a Service (IaaS) cloud using stochastic modeling. They considered the expected request completion time, rejection probability and the overhead rate as the key metrics. Calheiros *et al*. [2] proposed virtual machine provisioning to achieve QoS targets by detecting the workload arrival pattern, resource demands that occur over time. The authors result showed the importance of analyzing the workload patterns to meet the QoS. An analytical technique based on approximate Markov chain model for performance evaluation of a cloud computing center is presented by khazaei *et al*. [12, 14]. Performance Evaluation using queueing theory for network of computers was also proposed by Roberttazzi [18].

Concepts of queuing theory may be used to maximize profit, virtual machine provisioning and utilization of computing resources. Jiang *et al*. [9] has devised analytics for virtual machine provisioning. Assessment of cloud centers performance such as arrival rate of super tasks, degree of virtualization, response time and power management was proposed by khazaei *et al*. [10]. Their model showed that appropriate arrangement of server pools and the required electricity power could be identified in advance for anticipated arrival process and super task characteristics. They also had proposed an analytical model [11] that indicates request homogenization obtained by partitioning the incoming super task on the basis of super task size and coefficient of variation of task service time to improve mean response time, waiting time and queue length. khazaei *et al*. [13] have given a fine grained model of cloud computing cenetrs. Xia *et al*. [21] presented a stochastic approach for energy efficiency and performance analysis of dynamic voltage scaling enabled cloud. They introduced a framework to save energy consumption in cloud data centers by lowering the supply voltage and operating frequency to save energy Queuing theory in cloud computing to reduce the waiting time using multi server was presented by Sowjanya *et al*. [19] The research work discussed so far clearly depicted that, performance indicators were essential for efficient cloud resource allocation.

In order to maximize the revenue, customers may also oversubscribe computing resources in cloud data centers. The risk behind oversubscription of cloud resources is showed by Householder *et al.* [5] Also the workloads to cloud centers are uncertain. They must be classified on the basis of their resource requirement. A survey by Rahman *et al.* [17] summarizes the challenges of geographic load balancing in grid environment. Various cloud workload are categorized from cloud providers and users point of view is given by Mulia *et al.* [15]. It is observed in the literature that analysis of workloads and resource allocations require more attention in cloud data centers. The research work discussed above mostly covers First Come First Serve queue discipline. But there are also workloads that need immediate server allocation [16]. A cloud data center may also receive requests from different classes of clients. Ellens *et al*. [4] applied queuing theory with two priority classes to analyze the performance of cloud computing centers. However, in this work, in order to design and analyze the performance of cloud data centers five different priority classes are considered. This extended priority class approach will enhance the performance of the system by classifying the workload with better accuracy.

3. System Architecture

This section defines the architecture of data center design using M/M/S non preemptive priority queuing system as shown in Figure 1. The customer request to cloud data center is processed by priority based task classifier and allocator. The priority based task classifier and allocator has three components namely task classifier, priority classifier and task allocator.

The task classifier classifies the customer request to a workload type using the rule engine. It classifies the request based on the workload characteristics such as maximum expected time to complete a request;

maximum expected number of virtual machines required and also on the basis of parallelizable or non parallelizable workloads. The rule engine has predefined rules to classify the request to a workload category.

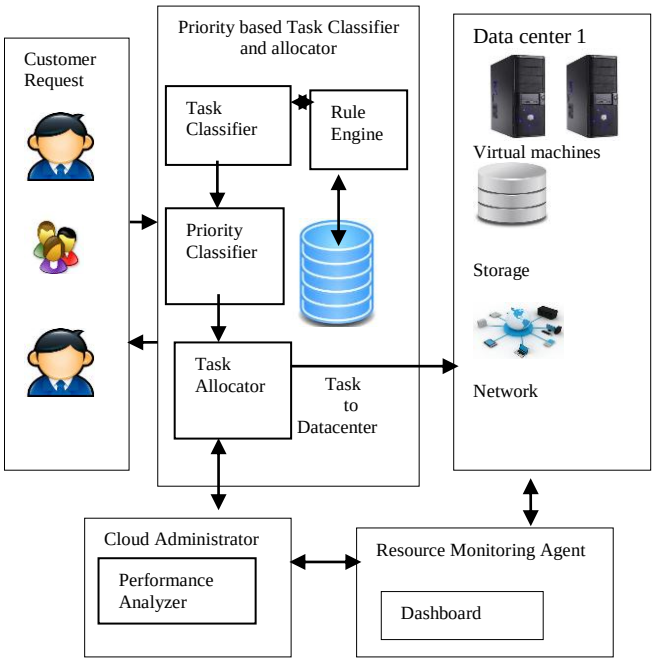

Figure 1. Architecture of cloud data center using priority queuing model.

The priority classifier assigns priority for resource provisioning of workloads, by paying attention to the workload attributes.

Virtual machines are allocated by the task allocator component. The resource monitoring agent monitors the virtual machine's Central Processing Unit (CPU) and memory utilization and displays the usage scenario in the form of dashboard. The cloud administrator analyzes the utilization of computing resources for further resource optimization. Task allocator also verifies the virtual machine usages for further allocation services. The task classifier which takes care of different categories of required workload is one of the important components of our system. The following section briefs about the categories of workload.

3.1. Workload Categories

In cloud data centers, customers make different classes of workloads according to their application requirements. The system classifies workloads into web workloads, batch workloads, database workloads, analytical workloads and high numeric intensive workloads categories.

Table 2. Categorization of cloud workloads.

Categories of job request		Types of workload	Applications	Types of Virtual instance	Priority classification
Delay tolerant job request	Non shared VMs	Simple Web content serving	Web search, free trails to explore VM functionality	Micro instance type	Type 5
	Shared VMs	Batch workloads	Sales transactions.	General purpose instance type	Type 4
Delay sensitive job request	Sharable heterogeneous VMs and less delay sensitive with memory intensive request.	Database workloads	Data mining and data warehousing, business intelligence, business decision applications,	Memory optimized instance type	Type 3
	Sharable heterogeneous VMs and medium delay sensitive with compute intensive request.	Analytic workloads	OLAP, marketing and sales forecasting, risk management	Medium compute optimized instance	Type 2
	Sharable heterogeneous VMs, highly delay sensitive with compute optimized request.	High Numeric intensive workloads	Engineering design, HPC, scientific applications, simulations.	Compute optimized instance with more vCPUs.	Type 1

The priority system assigns least priority or type 5 to simple web workloads such as free trails to explore virtual machine functionality. Batch workloads make repetitive transactions to a database. It places high demands on both processor and I/O resources. There is no time constraint to complete batch workloads. Response is not immediately needed. So, batch workloads are categorized as type 4 customers. Database workloads are managed across several computing environments and assigned type 3 request. Databases may span across multiple heterogeneous virtual machines. Those workloads are categorized as sharable with less delay sensitive and moderate memory intensive jobs. Analytical workloads perform business optimization such as market analysis, revenue forecasting performed by business analysts. It requires extreme data volume and also makes complex computation. Their response time is frequently measured in ten to hundred seconds. They require much more compute capabilities for running the applications. So, it is categorized as type 2 request. They may share even VMs and these workloads are medium delay sensitive with compute intensive jobs.

High numeric intensive workloads such as scientific applications, engineering design and simulations need complex compute capabilities are scheduled as type I request. They are categorized as highly delay sensitive workloads. Table 2 summarizes the various categories of workload considered in our system.

3.2. Rule Engine

Table 3. Mapping of cloud workloads to virtual instances.

Nature of job request	Workload category	Instance type
If Job request is DELAY TOLERANT JOB REQUEST and NON SHARED VMs	**Then** Categorize(Simple web workload)	Assign(Micro instances)
If Job request is DELAY TOLERANT JOB REQUEST and SHARED VMs	**Then** Categorize(Batch workload)	Assign(General purpose instances)
If Job request is SHARABLE HETEROGENEOUS VMs AND LESS DELAY SENSITIVE WITH MEMORY INTENSIVE	**Then** Categorize(Database workloads)	Assign(High memory instances)
If Job request is SHARABLE HETEROGENEOUS VMs AND MEDIUM DELAY SENSITIVE WITH COMPUTE INTENSIVE	**Then** Categorize(Analytic workloads)	Assign(High CPU instances)
If Job request is SHARABLE HETEROGENEOUS VMs, HIGHLY DELAY SENSITIVE WITH COMPUTE OPTIMIZED.	**Then** Categorize(High Numeric intensive workloads)	Assign(Cluster compute instances)

The rule engine Table 3 is framed using IF- THEN construct. The 'IF' part analyzes the nature of job request, 'THEN' part categorizes the request to any one of the 5 workloads. Based on the workload category a virtual machine instance is assigned to the requested job. The table shows how jobs are assigned to an amazon instance type [1] based on the categories of workload.

4. Cloud Datacenter Design based on M/M/S Queue Model

In IaaS Cloud, when a customer request arrives, it is assigned an infrastructure instance that fulfils the user request. In order to evaluate the performance of our proposed architecture, the system considers 3 categories of design using priority queuing discipline.

Consider a cloud computing environment that classifies customer's request into i classes with arrival rate follows Poisson distribution of λ_1, λ_2,… λ_i. The system assigns different priorities to the incoming customer's request such that type 1 as the highest priority and type i as the lowest priority. Let the service rates of i classes be denoted by μ, with the service time distribution being exponentially distributed Table 3 Each customer request is classified by the task classifier in to a workload category such as simple web workload that makes free trails to explore VM functionality, batch workload to analyze the daily sales transaction, database workload to run business intelligence application, analytic workload to perform OLAP and risk management and high numeric intensive workload for engineering design and scientific applications. The paper evaluates the performance of cloud datacenters using three categories. The category 1 shows M/M/1 queuing model with Non preemptive Priority discipline, category 2 explains M/M/S system with Non preemptive Priority discipline with same service rate and category 3 shows M/M/S system with Non preemptive Priority discipline using different service rate.

4.1. Category 1: M/M/1 System with Non Pre-emptive Priority Discipline

Consider a scenario where a single server facility receives two types of job requests namely type 1 and type 2 jobs. Type 1 and Type 2 jobs arrive according to independent poisson process with rate λ_1 and λ_2 respectively with the processing times of all jobs are exponentially distributed with the same mean $1/\mu$.

Consider $\rho_1+\rho_2<1$ where $\rho_i = \lambda_i / \mu$ be the utilization rate due to type i jobs. Type 1 jobs are treated with priority over type 2 jobs. Let the random variable Li denote the number of type *i* jobs in the system and Si the throughput time of a type *i* job. $E[Li]$ and $E[Si]$ for $i = 1, 2$. Consider a high priority workload such as high numeric intensive workload and low priority workload such as simple web search workload that arrive according to independent poisson process with rate $\boldsymbol{\lambda_1}$ and λ_2 and processing times of both jobs are exponentially distributed with the same mean $1/\mu$.

Let $E[L_1]$ and $E[L_2]$ be the number of high numeric intensive workload and simple web search workload, $E[S_1]$ and $E[S_2]$ be the mean throughput time of high numeric intensive workload and simple web search workload. The mean throughput time or system time $E[S_1]$ of Type1 job is given by

$$E[S_1] = E[L_1]\frac{1}{\mu} + \frac{1}{\mu} + \rho_2 \frac{1}{\mu} \quad (1)$$

The first term defines the mean number of type1 jobs with their processing time $E[L_1]1/\mu$ and second defines residual processing time of job in the server and last term defines the type 2 job being processed by the server which is ρ_2. In non Pre-emptive priority type1 has to wait until the completion of type 2 job.
Using Little's law

$$E[L_1] = \lambda_1 E[S_1] \quad (2)$$

substituting $E[L_1]$ in (1) we get

$$E[S_1] = \frac{(1+\rho_2)\rho_1}{(1-\rho_1)} \quad (3)$$

The number of high numeric intensive job or the high priority job is

$$E[L_1] = \frac{(1+\rho_2)\rho_1}{(1-\rho_1)} \quad (4)$$

Where $\rho_1 = \lambda_1/\mu$ and $\rho_2 = \lambda_2/\mu$ and ρ_1 is the utilization rate of high priority or high numeric intensive job and ρ_2 the utilization of low priority or simple web search workload. The total number of jobs in the system $E[L]$ is either type 1 or type 2 job and does depend on the order of their arrival.

$$\text{So, } E[L] = E[L_1] + E[L_2] \quad (5)$$

Substituting $E[L_1]$ in Equation (5) we get $E[L_2]$, the number of type 2 job is given by

$$E[L_2] = \frac{(1-\rho_1(1-\rho_1-\rho_2))\rho_2}{(1-\rho_1)(1-\rho_1-\rho_2)} \quad (6)$$

Applying Little's law, the throughput or system time of type 2 job $E[S_2]$

$$E[S_2] = \frac{(1-\rho_2(1-\rho_1-\rho_2))/\mu}{(1-\rho_1)(1-\rho_1-\rho_2)} \quad (7)$$

4.2. Category 2: M/M/S System with Non Pre-emptive Priority Discipline with Same Service Rate

In category 2 we consider multi server facility that process 5 different type of priority classes such as Type 1, Type 2, Type 3, Type 4 and Type 5. Assume they arrive according to independent Poisson processes with rate λ_1, λ_2, λ_3, λ_4, and λ_5 respectively.

Thus the overall arrival pattern is also Poisson with mean

$$\lambda = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5 \quad (8)$$

The server utilization is given by

$$\rho = \rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5 < 1 \quad (9)$$

Where the server utilization due to type$_i$ job is given by $\rho_i = \lambda_i/(S\mu)$, and S, the total number of servers, μ defines the service rate of servers. The processing times of all jobs are exponentially distributed with the mean $1/\mu$.

Let π_w be the probability that a job has to wait in M/M/S with no priority is given by,

$\pi_W = P_S + P_{S+1} + P_{S+2} \ldots\ldots\ldots\ldots$ and P_s denotes the probability that there are 's' jobs in the system.

$$\pi_W = \frac{(s\rho)^s}{s!}\left((1-\rho)\sum_{n=0}^{s-1}\frac{(s\rho)^n}{n!} + \frac{(s\rho)^s}{s!}\right)^{-1} \quad (10)$$

n denotes the number of jobs in the system.

Mean waiting time for a Type 1 job is given by

$$E[W_1] = \pi_W \frac{1}{s\mu} + E(L_1^q)\frac{1}{s\mu} \quad (11)$$

$E(L_1^q)$ denotes the number of Type1 job in the queue.

From Little's law

$$E[L_1^q] = \lambda_1 E[W_1] \quad (12)$$

Substituting $E[L_{q1}]$ in (11) $E[W_1]$ becomes

$$E[W_1] = \frac{\pi_W}{1-\rho_1} \bullet \frac{1}{s\mu} \quad (13)$$

Substituting $E[W_1]$ in $E(L_1^q)$ becomes

$$E[L_1^q] = \frac{\pi_W \rho_1}{1-\rho_1} \quad (14)$$

If the system accepts only two different job types then

$$E[L_1^q] + E[L_2^q] = \frac{\pi_W \rho}{1-\rho} \quad (15)$$

By inserting Equation (14) and Equation (15) $E(L_2^q)$ becomes

$$E[L_2^q] = \frac{\pi_W \rho_1}{(1-\rho)(1-\rho_1)} \quad (16)$$

$$E[W_2] = \frac{\pi_W}{(1-\rho)(1-\rho_2)} * \frac{1}{s\mu} \quad (17)$$

Similarly, for job types more than 2 (i>2), then the mean waiting time for a class i job is given by

$$E[W_3] = \frac{\pi_W}{(1-\rho)(1-(\rho_1+\rho_2))} * \frac{1}{s\mu} \quad (18)$$

$$E[W_4] = \frac{\pi_W}{(1-\rho)(1-(\rho_1+\rho_2+\rho_3))} * \frac{1}{s\mu} \quad (19)$$

$$E[W_5] = \frac{\pi_W}{(1-\rho)(1-(\rho_1+\rho_2+\rho_3+\rho_4))} * \frac{1}{s\mu} \quad (20)$$

In general,

$$E[W_i] = \frac{\pi_W}{(1-\sum_{j=1}^{i}\rho_j)(1-\sum_{j=1}^{i-1}\rho_j)} * \frac{1}{s\mu} \quad (21)$$

4.3. Category 3: M/M/S System with Non Pre-Emptive Priority Discipline with Different Service Rate

In category 3 we consider multi server facility that process 5 different type of priority classes such as Type1, Type 2, Type 3, Type 4 and Type 5. Assume they arrive according to independent Poisson processes with rate λ_1, λ_2, λ_3, λ_4, and λ_5, service time with μ and 2μ respectively. All other parameters are considered same as category 2.

4.4. Drawbacks of Strict Priority Queue Model

In strict priority queue, low priority workloads have to wait until the completion of high priority workloads. So if the arrival rates of high priority jobs are higher, then low priority jobs may suffer from starvation. So, in our model we define a delay tolerance level for each workload category.

5. Delay Tolerance based Non Preemptive Priority Queue Model

Consider a cloud datacenter with multi server facility that process 5 different type of priority classes such as Type 1, Type 2, Type 3, Type 4 and Type 5. Assume they arrive according to independent Poisson processes with rate $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, $\boldsymbol{\lambda}_3$, $\boldsymbol{\lambda}_4$, and $\boldsymbol{\lambda}_5$ and μ be the service rate of servers and D_1, D_2, D_3, D_4, and D_5 be the delay tolerance level of each workload category.

Thus the overall arrival pattern λ is also Poisson with mean.

$$\lambda = \lambda_1 + \lambda_2 + \lambda_3 + \lambda_4 + \lambda_5$$

The server utilization is given by

$$\rho = \rho_1 + \rho_2 + \rho_3 + \rho_4 + \rho_5 < 1$$

Where the server utilization due to type$_i$ job is given by $\rho_i = \boldsymbol{\lambda}_i / (S\mu)$, and S, the total number of servers, μ defines the service rate of servers. The processing times of all jobs are exponentially distributed with the mean 1/μ. For workload of type i, the delay tolerance is given by Di. The system considers delay tolerance Di in the arrival rate of workload i. To avoid starvation of lower priority jobs, the arrival rate of higher priority jobs will be dynamically slowed down in such a way that their Quality of service is not degraded.

Algorithm 1: Delay Tolerant based M/M/S Non Preemptive Priority Queue

Input: $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, $\boldsymbol{\lambda}_3$, $\boldsymbol{\lambda}_4$, and $\boldsymbol{\lambda}_5$, Arrival Rates of 5 priority classes of Workloads
μ, Service Rate of Virtual Machines
c, Number of Virtual Machines.
Variable : Di, Delay of ith class of CLOUDLETS
Constraint: Arrival count of CLOUDLETS[i] < CLOUDELTS[i+1]
Output: Waiting Time of 5 priority classes of Workloads.
Begin
TotalDatacenters = 1;
TotalHost = 1;
//VM DESCSRIPTION
int[] VM_TYPES= {0,1,2,3};
int[] VM_MIPS= { 2600, 2100, 1100, 600 };
int[] VM_RAM = { 875, 1760, 1730, 600 };
int VM_BW = 100000;
int VM_SIZE = 2300;
//CLOUDLET DESCRIPTION
int CLOUDLET_TYPES = 5;
int [] CLOUDLET_LENGHT = {75000, 60000, 50000, 40000, 30000 };
TotalVM = nVM
TotalCloudlets = nREQ;
// CREATE VMLIST
createVmList(brokerId,totalVM, VM_TYPES[i]);
//CREATE CLOUDLET LIST
create CLOUDELTLIST
For vms=1 to TotalVM;
For cloudlets = 1 to TotalCloudlets;
//FOR EACH CLOUDLET TYPE;
For i=0;i<cloudlets
myCloudlet[i] = new MyCloudlet(i); list.add(myCloudlet[i]);
Assign cloudets to a vmtype
Calculate the overall utilization ρ (%) of Virtual machines
*Observe the utilization **ρ (%)**,*
Calculate the waiting time E[WLETi] of each class of CLOUDLET

$$E[WCLETi] = \frac{\pi_w}{(1 - \sum_{j=1}^{i} \rho_j)(1 - \sum_{j=1}^{i-1} \rho_j)} * \frac{1}{(nVM) * \mu}$$

For a utilization (%) ρ > 90
Store the values of $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$, $\boldsymbol{\lambda}_3$, $\boldsymbol{\lambda}_4$, $\boldsymbol{\lambda}_5$, μ, c
For the same utilization value

do
Decrease / Delay Di < $\boldsymbol{\lambda}$i, the arrival rate of available highest Priority workload and adjust the arrival rates of remaining workloads.
Calculate new waiting time E[newWCLETi].
while E[new WCLETi] < E[WCLETi];
End for
End for
end.

6. Performance Evaluation

The resulting models have been implemented and solved using Java. We show the effects of changing the arrival rates and service rates with different number of servers using three design categories. Category 1 defines M/M/1 model with Non preemptive priority discipline. Category 2 defines M/M/S Non preemptive priority model with same service rate and finally M/M/S non preemptive model with different service rates is defined in category 3.

- **Category 1: M/M/1 system with Non Preemptive Priority Discipline**

Consider a cloud server that receives 2 types of workloads with two different priority levels. Table: 4 shows the number of high and low priority jobs (L) and system time (S) required for both priority jobs with varying levels of utilization rate (ρ). It shows the system time required to complete high priority job is less than the low priority job. It is also depicted in graphically in Figure 2.

6.1. Category 1: M/M/1 system with Non Pre-Emptive Priority Discipline

Consider a cloud server that receives 2 types of workloads with two different priority levels. Table 4 shows the number of high and low priority jobs (L) and system time (S) required for both priority jobs with varying levels of utilization rate (ρ). It shows the system time required to complete high priority job is less than the low priority job.

Table 4. Number of Jobs (L_i) and throughput (S_i) for 2 priority classes with μ=1.

Server Utilization ρ (%)	20	30	40	50	60	70	80	90
Number of High Priority Jobs L_1)	0.122	0.13	0.3	0.325	0.35	0.375	0.40	0.6
Number of Low Priority Jobs(L_2)	0.127	0.29	0.36	0.675	1.15	1.95	3.6	8.3
Throughput / System time of High priority Jobs (S_1)	1.22	1.33	1.5	1.624	1.75	1.87	2	2.28
Throughput / System time of Low priority Jobs (S_2)	1.27	1.47	1.83	2.25	2.87	3.91	6	13.8

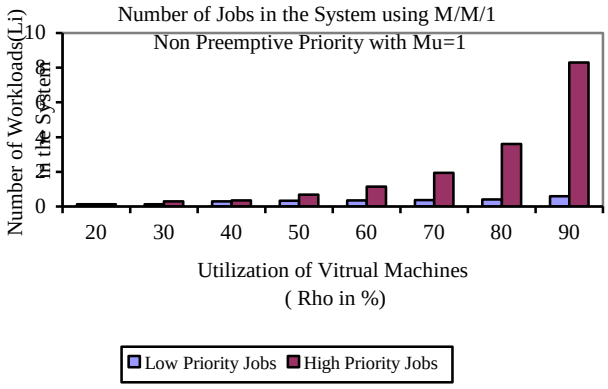

Figure 2. Number of Jobs (L_i) for high and low Priority jobs with μ = 1.

6.2. Category 2: M/M/S System with Non Preemptive Priority Discipline with Same Service Rate μ=6

Consider the cloud datacenter housed with multiple physical servers are virtualized to form multiple virtual machines. These virtual machines accept customer's workloads form poisson arrivals and service time being exponentially distributed. The model analyzes the waiting time of 5 different types of workloads with varying number of virtual machines and utilization rate (ρ). Table 5 shows the performance analysis of M/M/S Non-Preemptive Multi Server Model with λ_1=12 workloads/sec, λ_2=11 workloads/sec, λ_3=10 workloads/sec, λ_4=9 workloads/sec, and λ_5=8 workloads/sec and service rate μ = 6. It clearly shows that the waiting time of type 1 is less when compared to other types of job.

Table 5. Performance analysis of M/M/S non-preemptive multi server model with same service rate.

Number of virtual machines Vs Waiting time (sec)	9 virtual machines	10 virtual machines	11 virtual machines	12 virtual machines	13 virtual machines
Utilization Rate ρ (%)	92	83	75	69	64
Waiting time of Type1 job (W1)	0.06	0.017	0.007	0.003	0.001
Waiting time of Type2 job (W2)	29.732	3.696	1.065	0.393	0.159
Waiting time of Type3 job (W3)	40.282	4.795	1.338	0.482	0.191
Waiting time of Type4 job (W4)	59.464	6.570	1.744	0.605	0.234
Waiting time of Type5 job (W5)	104.062	9.337	2.398	0.787	0.293

Consider the performance analysis using 9 virtual machines. If the arrival rate of highest priority job is 12 jobs per second and lowest priority job is 8 jobs per second then the waiting time of highest priority job is given 0.06 sec and the lowest priority job is given by 104.06 sec. For the same arrival rates the waiting time is decreased drastically if there are 10 virtual machines installed. Figure 3 shows the graphical representation of the waiting times.

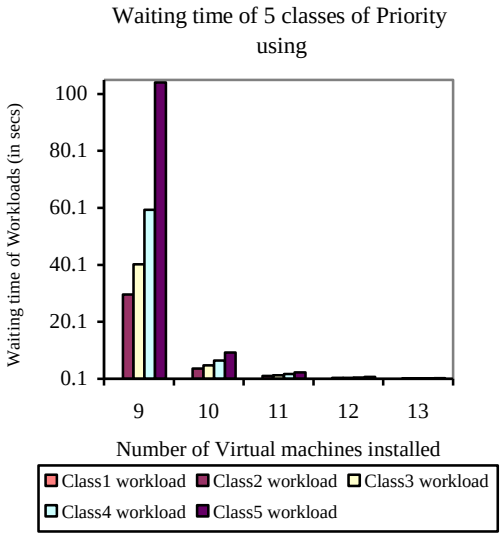

Figure 3. Waiting time of 5 priority classes with varying number of virtual machines.

6.3. Category 3: M/M/S System with Non Preemptive Priority Discipline with Same Different Service Rate

In cloud datacenter, there will be multiple high end servers with different service rate. High Numeric intensive jobs require servers of high service rate. So, the cloud providers must allocate physical or virtual servers based on the type of the workloads. In category 3 the system evaluates the waiting time of workloads with different service rate.

Table 6. Performance analysis Of M/M/S non-preemptive multi server model with different service rate.

Different service rate Vs Waiting time (sec)	μ = 1	μ = 2
Utilization Rate ρ (%)	93	46
Waiting time of Type1 job (W1)	0.210	0.0001
Waiting time of Type2 job (W2)	3.365	0.0013
Waiting time of Type3 job (W3)	5.288	0.0016
Waiting time of Type4 job (W4)	9.254	0.0018
Waiting time of Type5 job (W5)	18.509	0.0020

Table 6 shows the reduction in the waiting time when the service rate is doubled. The utilization rate is also decreased from 93 % to 46 % if the service rate is doubled. The graphical form is shown in Figure 4.

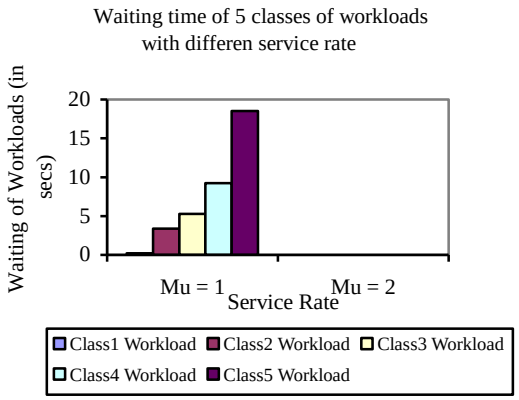

Figure 4. Waiting time of 5 priority classes with varying number of server rate.

6.4. Delay Tolerant based M/M/S Non Preemptive Priority Queue

The system considers maximum waiting time for each category of workload. The Table 7 shows the performance analysis of M/M/S Non-Preemptive Multi Server Model with λ_1=2 workloads/sec, λ_2=14 workloads/sec, λ_3=13 workloads/sec, λ_4=12 workloads/sec, and λ_5=9 workloads/sec and service rate μ=6 as shown in the Figure 5. On comparing the performance analysis shown in Table 5 and Table 7, for the same number of servers if the type arrival rate is Type 1 job is reduced from 12 workloads per second to 2 workloads per second, the waiting time is greatly reduced. So, no need to increase the number of virtual instances if the system reduces the arrival rate without degrading the quality of service of high priority jobs. The system has the advantage of reducing the number of virtual machines in order to ensure power management.

Table 7. Performance analysis of delay tolerant based M/M/S Non-preemptive multi server model with same service rate.

Number of Virtual machines Vs Waiting time in secs	9 Virtual Machines	10 Virtual Machines	11 Virtual Machines	12 Virtual Machines	13 Virtual Machines
Utilization Rate ρ (%)	92	83	75	69	64
Waiting time of Type1 job (W1)	0.049	0.014	0.006	0.002	0.001
Waiting time of Type2 job (W2)	24	3.05	0.8	0.33	0.13
Waiting time of Type3 job (W3)	32.8	4.03	1.15	0.42	0.17
Waiting time of Type4 job (W4)	49.9	5.7	1.55	0.54	0.21
Waiting time of Type5 job (W5)	96.0	9.33	2.30	0.76	0.28

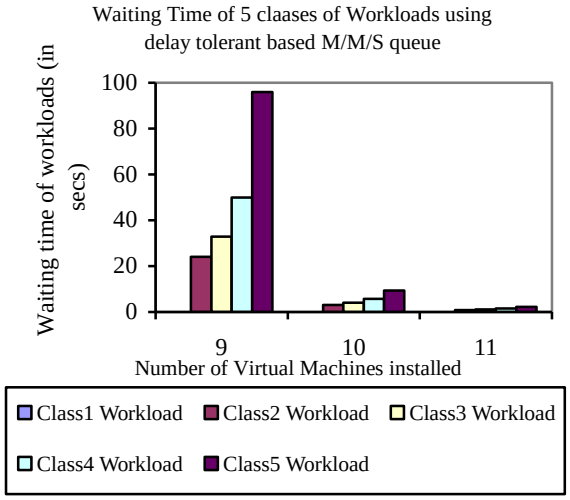

Figure 5. Waiting time of 5 priority classes using delay tolerant based non pre-emptive priority.

Performance analysis of M/M/S Non preemptive multiserver model using strict priority and delay tolerant model for priority 1 workload is shown in Figure 6. It shows the reduction in waiting time of priority1 workload when using delay tolerant based non pre-emptive priority queuing model.

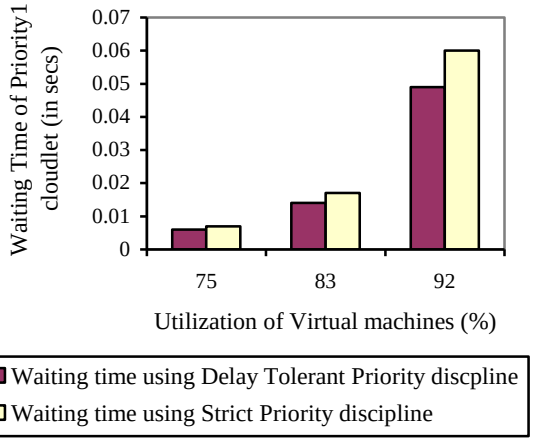

Figure 6. Comparison of waiting time of priority 1 workload using strict priority and delay tolerant based non preemptive priority.

7. Conclusions

In this paper, we have proposed a novel data center design using delay tolerant based priority queuing model for resource provisioning, by paying attention to individual customer attributes. Our results are introduced by the stochastic analysis, which yielded positive results for the operation of our model that exhibits a behaviour far superior to First In First Out (FIFO) and strict priority discipline. We have also obtained supportive results from the implementation that alleviate uncertainties from adopting numerous assumptions on the stochastic analysis section.

Therefore we can safely claim that our data center design has the potential to achieve higher customer satisfaction. Our next step is to enhance our evaluation

towards to extend the experiments by using the real cloud environment.

References

[1] Amazon EC2 Instance Types. Available from <https://aws.amazon.com/ec2 /instance-types/>, Last Visited, 2017.

[2] Calheiros N., Ranjan R., and Buyya R., "Virtual Machine Provisioning Based on Analytics Performance and Qos in Cloud Computing Environments," *in Proceedings of International Conference on Parallel Processing*, Taipei City, pp. 295-304, 2011.

[3] Chen H. and Li S., "A Queueing-Based Model for Performance Management on Cloud," *in Proceedings of* 6th *International Conference on Advanced Information Management and Service*, Seoul, pp. 83-88, 2010.

[4] Ellens W., Ivkovic M., Akkerboom J., Litjens R., and Berg H., "Performance of Cloud Computing Centers With Multiple Priority Classes," *in Proceedings of IEEE* 5th *International Conference on Cloud Computing*, Honolulu, pp. 245-252, 2012.

[5] Householder R., Arnold S., and Green R., "On Cloud-Based Oversubscription," *International Journal of Engineering Trends and Technology*, vol. 8, no. 8, pp. 425-431, 2014.

[6] Ivo A. and Resing J., *Queueing Theory*, Eindhoven University of Technology, 2002.

[7] Ivo A., 2003. M/M/1 with priorities. [online]. Available from World Wide Web: http://www.win.tue.nl/~iadan/sdp/ h4prior.pdf, Last Visited, 2013.

[8] Ivo A., Multi-Machine Systems. [online]. Available from World Wide Web: http://www.win.tue.nl/~ iadan/sdp/ h11.pdf, Last Visited, 2013.

[9] Jiang Y., Perng C., Li T., and Chang N., "Cloud Analytics for Capacity Planning and Instant VM Provisioning," *IEEE Transactions on Network and Service Management*, vol. 10, no. 3, pp. 312-325, 2013.

[10] Khazaei H., Misic J., Misic V., and Rashwand S., "Analysis of A Pool Management Scheme for Cloud Computing Centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 5, pp. 849-861, 2013.

[11] Khazaei H., Jelena Misic J., and Misic B., "Performance of Cloud Centers with High Degree of Virtualization under Batch Task Arrivals," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 12, pp. 2429 - 2438, 2013.

[12] Khazaei H., Misic J., and Misic V., *Cloud Computing: Methodology, System and Applications*, Taylor and Francis Group, 2012.

[13] Khazaei H., Misic J., and Misic V., "A Fine-Grained Performance Model of Cloud Computing Centers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 11, pp. 2138-2147, 2013.

[14] Khazaei H., Misic J., and Misic V., "Performance Analysis of Cloud Computing Centers Using M/G/M/M+R Queuing System," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 5, pp. 936-943, 2012.

[15] Mulia D., Sehga N., Sohoni S., Acken M., Stanberry C., and Fritz L., "Cloud Workload Characterization," *IETE Technical Review*, vol. 30, no. 5, pp. 382-397, 2013.

[16] Natsheh E., Jantan B., Khatun S., and Shamala S., "Fuzzy Active Queue Management for Congetion Control in wireless Ad-hoc," *The International Arab Journal of Information Technology*, vol. 4, no. 1, pp. 50-59, 2007.

[17] Rahman A., Liu X., and Kong F., "A Survey on Geographic Load Balancing Based Data Center Power Management In The Smart Grid Environment," *IEEE Communication Surveys and Tutorials*, vol. 16, no. 1, pp. 214-233, 2013.

[18] Roberttazzi T., *Computer Networks and Systems-Queuing Theory and Performance Evaluation*, Springer-Verlag, 2000.

[19] Sowjanya S., Praveen D., Satish K., and Rahiman A., "The Queueing Theory in Cloud Computing to Reduce the Waiting Time," *IJCSET*, vol. 1, no. 3, pp. 110-112, 2011.

[20] Stallings W., *Queueing analysis*, A Practical Guide to Computer Scientists. [online]. Available from World Wide Web, http://www.computersciencestudent.com/styled/QueueingAnalysis, Last Visited, 2013.

[21] Xia Y., Zhou M., Luo X., Zhu Q., Li J., and Huang Y., "Stochastic Modeling and Quality Evaluation of Infrastructure-as- A Service Clouds," *IEEE Transaction on Automation Science and Engineering*, vol. 12, no. 1, pp. 162-170, 2015.

Meera Annamalai working as an Associate Professor in the Department of Information Technology, Tagore Engineering College, Chennai, India. She has presented papers in various National and International conferences and published papers in reputed journals. Her research interest includes Cloud Computing and Distributed Databases.

Swamynathan Sankaranarayanan working as an Associate Professor of Department of Information Science and Technology, College of Engineering Campus, Anna University, Chennai, India. He has more than 20 years of teaching and research experience. He has carried out various funded projects. He has published more than 80 papers in reputed journals and conference proceedings. His research interest includes Distributed Computing, Semantic Web and Data Analytics.