International Journal of Cloud Computing and Services Science (IJ-CLOSER)
Vol.2, No.3, June 2013, pp. 170~183
ISSN: 2089-3337

Optimization of QoS parameters through flexible Resource Scheduling in Finite Population Cloud Environment

Veena Goswami*, Sudhansu Shekhar Patra*, G.B.Mund**
* School of Computer Application, KIIT University
** School of Computer Engineering, KIIT University

Article Info

Article history:

Received Mar 15th, 2013
Revised Apr 20th, 2013
Accepted May 20th, 2013

Keyword:

Cloud computing
performance
scheduling
private cloud
Quality of Services

ABSTRACT

Cloud computing renders more ability to existing internet technologies and web cluster to fit the emerging business needs by accessing distributed computing resources. It supports processing large data utilizing clusters of commodity computers to control the next generation data centers and empowered application service providers for deploying applications depending on user Quality of Service (QoS) requirements. This would demand tools and mechanisms for analyzing the performance of the cloud system. In this paper, we present two scheduling policies along with an analytical resource prediction model for each policy for private cloud system. Queuing models are employed to provide exact performance measures of such systems. Various performance measures under various load, network time- delay and buffer size of both the systems indicate that the proposed provisioning technique help the cloud operators to tune the resources accordingly to match the offerings with requirements.

Corresponding Author:

Veena Goswami,
School of Computer Application,
KIIT University,
Bhubaneswar 751024, India.
Email:veena_goswami@yahoo.com

1. INTRODUCTION

Cloud computing is a way through which organizations can increase their resource capabilities dynamically without investing in new infrastructure. It expands the existing IT capabilities. Cloud computing is a epitome for organizations to meet the rising business needs by accessing the distributed resources over the internet. It has been considered the new computing paradigm that would change the way computing resources have been purchased and used. The cloud computing is now commonly known as the fifth utility following electricity, water, gas and telephony [1].

With the evolution of cloud computing, the organizations computing resources has shifted from capital to operational cost. The service consumers are bearing only for the services consumed instead of the hardware or software resources. Through cloud system the service consumers have the flexibility to use the distributed resources such as network devices,computing resources and storage system distributed over wide distances to create a unique environment for themselves. Cloud computing as a promising computing paradigm has broadly attracted significant attentions of researchers from both industry [2] - [5] and academia [6] - [9]. Surveys on cloud computing services are reported in [10] and [11]. A comprehensive overview of the techniques and approaches in the fields of energy efficiency for data centers and large-scale multimedia services has been discussed in [12]. Using a self-devising method cloud formation can be achieved from

small to large and dynamic complex process. Resource management method can form resource groups and dynamically optimize the organizational structure of resources in conformity with resource changes in cloud computing [13]. This brings the heterogeneous geographic distributing and idle computer resources together effectively, which can provide the cloud computing environment a large number of available computing resources, and achieve the optimal scheduling and efficient utilization of these resources. The future of resource management in cloud computing may shift to user-centric resource sharing and competition. To help cloud customers, integrated budget and deadline conditions build a reasonable balance between the two conditions [14].

A private cloud configuration is attractive to organizations to leverage the scalability benefits of a shared virtual infrastructure that have a known application workload footprint in terms of CPU, memory, and storage utilization, and that have potentially stringent security considerations. It effectively eliminate the provisioning constriction present in the datacenter virtualization model while maintaining cost savings as the same many-to-one relationship exists between VMs and physical hosts [15]. In the private cloud environment, data and processes are managed within the organization without the restrictions of network bandwidth, which can be of great benefit for security compliance. In addition, private cloud services offer the provider and the user greater control of the cloud infrastructure, because user access and the networks used are limited and designated. To create private cloud initial cost is expensive, but gets minimal at later phases of using it as a service.

Cloud computing involves the following three basic components:

- **Clients:** Clients may use desktop computers, laptops, tablet computers or other mobile devices to access internet data or services.
- **Datacenter:** The datacenter is a set of servers where the requested applications are hosted. Virtualization helps multiple virtual servers to run on one physical server. The number of virtual servers that can run on a physical server depends on its size and speed and on the nature of the applications running on the virtual server.
- **Distributed servers:** The structure of cloud computing allows cloud providers to host physical servers in disparate geographical locations without affecting the interaction of cloud end-users. This increases the flexibility and security options of the service provider. In case of any fault in a datacenter, a service will be still accessible through another distributed server. In addition, in case that cloud needs more hardware devices to support its workload, it is not necessary to attach more servers onto the primary datacenter but can be set up at another group of distributed servers and to be automatically embedded to the cloud.

Cloud Computing's technical, business, and policy issues play out across three layers:

- **The Infrastructure layer:** It encompasses the hardware, networks and operating systems responsible for managing fundamental resources such as data storage, computation and network bandwidth. A decisive element of the Cloud Infrastructure layer is the ability to virtualize the connection between physical resources and the services that consume them. There may be several virtual machines residing on a particular physical server, or there may be multiple physical servers running one particular virtual machine. Virtualization enables greater flexibility in managing the workloads, and to construct datacenters as providers can dynamically add, remove or modify hardware resources without having to reconfigure the services that depend on them.
- **The Platform layer:** This layer serves two purposes. It provides a set of common services, such as databases, messaging, and business rules engines, that are shared by applications. It also insulates application developers from the complexity of the underlying infrastructure through a set of higher level Application Programming Interfaces (APIs).
- **The Application layer:** It provides the mechanism through which users interact with the Cloud applications often through a web browser. In the Cloud datacenter the application layer is where the business logic for the application is run.

Cloud systems are receiving requests for different services and would in turn be evoking virtual devices for servicing them. It would be possible to model the incoming requests and the provisioning of services using statistical models as all these operations are random processes. Table 1 shows the cloud computing stack.

Table 1 The cloud computing stack

Level	Offered Services	Provider
Software as a Service(SaaS)	Applications (eg.,social networks,CRM)	Salesforce, Microsoft online services
Platform as a Service(PaaS)	Platform (eg.,programming languages,frameworks)	Google AppEngine, Microsoft Azure
Infrastructure as a Service(IaaS)	Infrastructure (eg.,computing servers,storage)	Amazon webservices,GoGrid,Rackspace

Cloud computing services can be broadly divided into three service models: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The service model describes the degree of service or control the cloud service provider offers and the degree of freedom the customer has. The cloud computing service models is depicted in Figure 1.

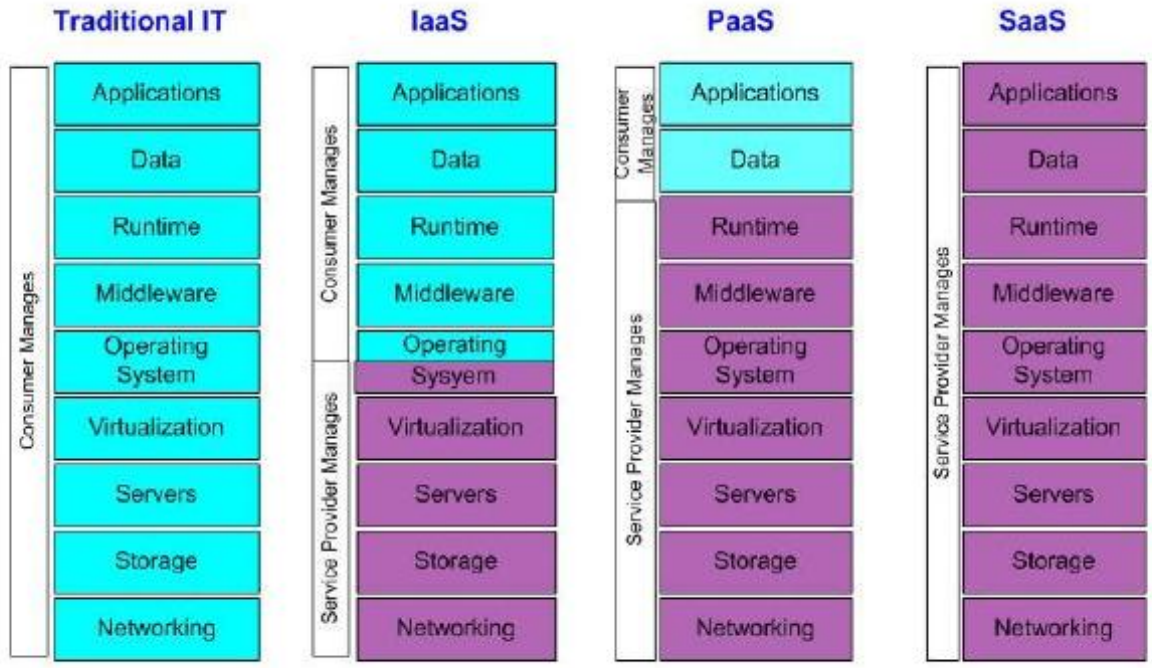

Fig 1: Cloud computing service models

The cloud computing deployment models are:

- **Public cloud:** Cloud infrastructure is hosted at vendor's premises. Public cloud is that model of cloud which allows the user to access the cloud facilities using the interim layer as web browser. Thus, all the services and infrastructure of cloud provider is available through internet. It also reduces the capital expenses as the cost is distributed and shared across a very large group of individuals and businesses [16].
- **Private Cloud:** Private Cloud is that model whose infrastructure is dedicated to a particular organization and is setup within an organization's internal datacenter. Private clouds are of two types: on-premise private clouds [17] and externally hosted private clouds. Externally hosted private clouds are also exclusively used by one organization, but are hosted by a third party specializing in cloud infrastructure. Externally hosted private clouds are cheaper than on-premise private clouds. The private cloud is ideal in situations when a company has a unique product or service that needs to be kept under strict control.

In situations when there is a frequent need to fiddle with the infrastructure, a private cloud is the best choice. A Virtual Private Cloud (VPC) is a private cloud existing within a shared or public cloud (i.e. the intercloud). Amazon web services launched Amazon Virtual Private Cloud on 2009-08-26, which allows the Amazon Elastic Compute Cloud service to be connected to legacy infrastructure over an Internet Protocol Security (IPsec) virtual private network connection [18].

- **Hybrid Cloud:** A hybrid cloud is a private cloud linked to one or more external cloud services, centrally managed, provisioned as a single unit, and circumscribed by a secure network. It provides virtual IT solutions through a mix of both public [19] and private clouds. Hybrid Clouds provide more secure control of the data and applications and allows various parties to access information over the internet. It also has an open architecture that allows interfaces with other management systems. The approach of temporarily renting capacity to handle spikes in load is known as "cloud-bursting" [20]. The cloud deployment models and characteristics are shown in Figure 2 and Table 2, respectively.
-

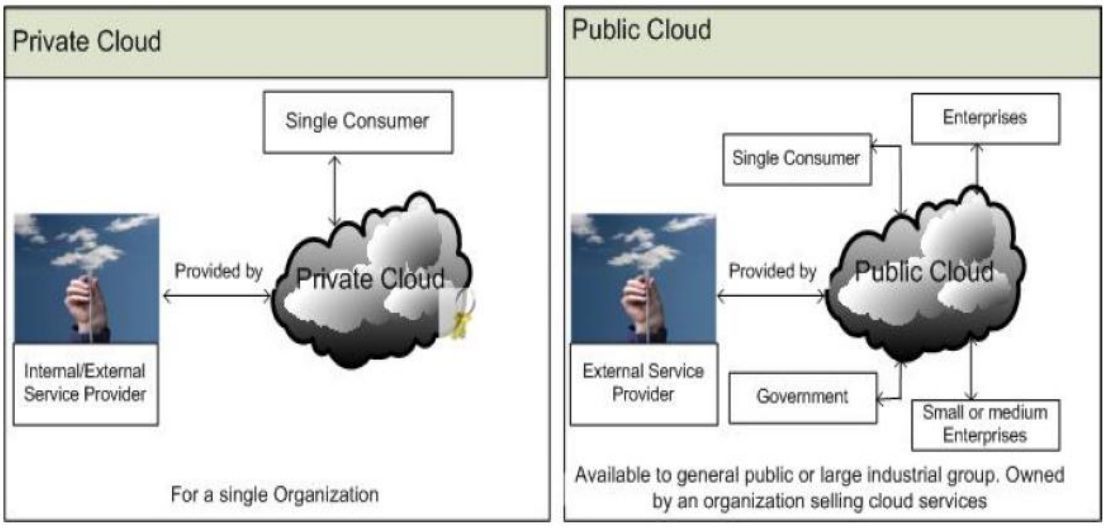

Fig 2 (a) Private Cloud Fig 2 (b) Public Cloud

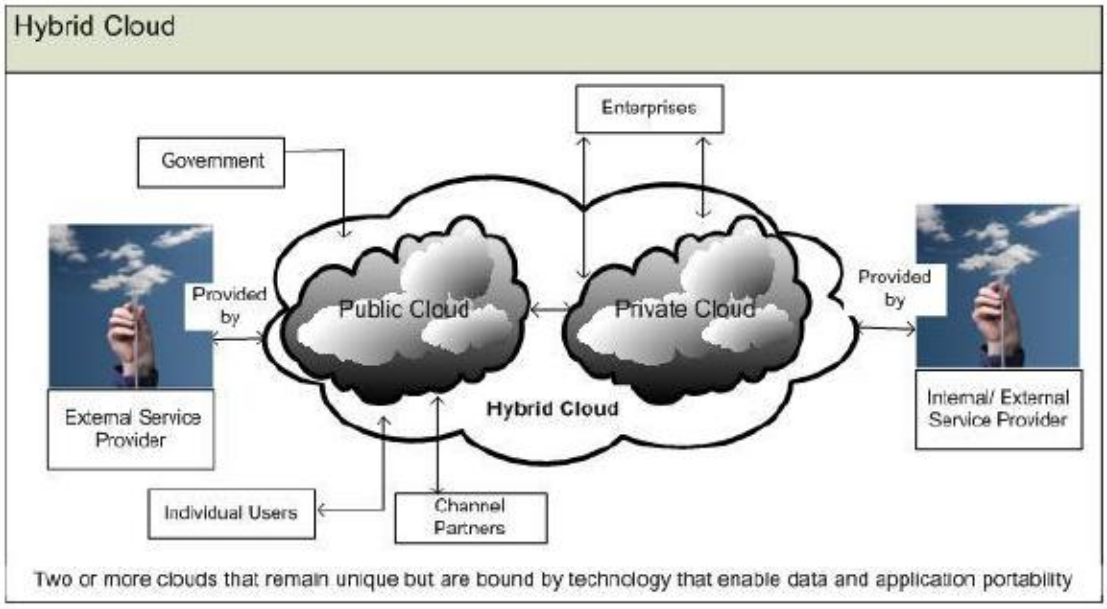

Fig 2 (c) Hybrid Cloud

Table 2: Cloud deployment Models

Deployment Model	Managed by	Owner of Infrastructure	Dedicated Hardware
Public	Cloud Service Provider	Cloud Service Provider	No
Private,External	Cloud Service Provider	Cloud Service Provider	Yes
Private,Internal	Internal Organization	Internal Organization	Yes
Hybrid	Mixed	Mixed	Depends on contract with cloud service provider

In this paper, we present two scheduling policies for a private cloud computing system. The rest of the paper is organized as follows. Section 2 gives the system model for partial allocation with waiting and analyzes the system. Section 3 presents the system model for partial allocation with blocking and analyzes the system. Computational experiences with a variety of numerical results in the form of graphs and table are discussed in Section 4. Finally, Section 5 concludes the paper.

2. PARTIAL ALLOCATION WITH WAITING (POLICY 1)

When a web application is deployed on a cloud, the cloud controller as the portal of a cloud establishes a queue to hold the client requests. A client requests for services through a cloud intermediary known as a cloud coordinator. When a client sends a request to the web application on the cloud, the cloud coordinator identifies the request and forwards the request to the appropriate web application. The service will be put on a queue depending on the number of service requests arriving to the application per unit time. If the number of requests is small and the system is fast, the request will be serviced without delay. On the otherhand, if the number of requests is large, the requests will have to wait for a long time, depending on the speed of the system and the availability of resources.

Policy 1 scheduling is a partial cloud approach with the option to wait. Requests are granted a complete or partial allocation, queueing the unsatisfied client requests for resource availability. We adopt M/M/c/K/N queuing system for the performance analysis of the model. The input source is limited, that is, the number of client request to the cloud system is of finite source N. For the value c< K <N we have delay-loss system, that is customers can arrive into the system until the number of customers in the system is K - 1 but then they must return to the source because the system is full. The service times are independent and exponentially distributed and the mean service time is 1/μ. The client requests are generated from N population according to quasi-random arrival process.

The one-dimensional process of the number of client requests at the service center is a birth-death process on a finite state space {0, 1,…, K} [21]. The quasi-random arrival process is a state-dependent birth process with the following rate

$$\lambda_i = (N - i), 0 \le i \le K - 1,$$

and the death process with the rate

$$\mu_i = \begin{cases} i\mu, 1 \le i \le c, \\ c\mu, c < i \le K. \end{cases}$$

We define P_i as the steady state probability that there are i client requests in the system. The steady-state distribution for finite buffer multi server queueing system with c homogeneous VMs are given by

$$P_i = \begin{cases} \binom{N}{i}\rho^i P_0, 0 \le i < c, \\ \frac{\binom{N}{i} i!\rho^i}{c!c^{i-c}} P_0, c \le i \le K, \end{cases} \quad (1)$$

where $\rho = \lambda/\mu$. Using normalizing condition $\sum_{i=0}^{K} P_i = 1$, we have

$$P_0 = \left[\sum_{i=0}^{c-1} \binom{N}{i} \rho^i + \sum_{i=c}^{K} \binom{N}{i} \frac{i!\rho^i}{c^{i-c}c!}\right]^{-1}. \quad (2)$$

Computational aspects:

In most cloud computing system when the number of client request is large, numerical difficulties arises in the direct use of the steady state probabilities. We show numerically stable methods of computation that avoids the computation of factorials and large power of loads. We establish steady state probabilities based

on recursive relations [22]. Taking $\varphi_i = P_i/P_0$ and using the relation of the birth-death process, recursive procedure operates as follows.

$$\varphi_0 = 1$$

$$\varphi_i = \left(\frac{N-i+1}{i}\right)\rho\varphi_{i-1}\ , 0 \le i \le c-1,$$

$$\varphi_i = \left(\frac{N-i+1}{c}\right)\rho\varphi_{i-1}\ , c \le i \le K.$$

Since, we know $P_0 = 1 - \sum_{i=1}^{K} P_i$, dividing both sides by P_0, we have

$1 = \frac{1}{P_0} - \sum_{i=1}^{K}\frac{P_i}{P_0} = \frac{1}{P_0} - \sum_{i=1}^{K}\varphi_i$. Hence $P_0 = [1 + \sum_{i=1}^{K}\varphi_i]^{-1}$ and $P_i = \varphi_i P_0$.
This computation is more stable than the direct use of (1).

2.2 Performance indices

Performance evaluation is an important aspect of cloud computing which is of crucial interest for both cloud providers and cloud customers. The effectiveness and utility of queueing model can be depicted and estimated by means of its performance indices. We obtain various performance measures in terms of steady state probabilities.

The expected number of client requests in the system (L_s) and expected number of client requests in the queue (L_q) are respectively, given as

$$L_s = \sum_{i=0}^{K} iP_i \quad = NP_0\left[\sum_{i=1}^{c-1}\binom{N-1}{i-1}\rho^i + \sum_{i=c}^{K-1}\binom{N-1}{i-1}\frac{i!\,\rho^i}{c!\,c^{i-c}}\right]$$

$$L_q = \sum_{i=c+1}^{K}(i-c)P_i = L_s - c + \sum_{i=0}^{c-1}(c-i)P_i.$$

The mean number of busy VMs ($\overline{c}$) and the mean number of requests in the source ($\overline{m}$) are respectively, given as

$$\overline{c} = \sum_{i=1}^{c-1} iP_i + c\,, \quad \overline{m} = N - L_s.$$

The utilization of the server (U_s) and the utilization of sources (U_t) respectively, can be calculated as

$$U_s = \frac{\overline{c}}{c}, \qquad U_t = \frac{\overline{m}}{N}.$$

The mean number of idle servers ($\overline{S}$) can be derived as $\overline{S} = c - \overline{c}$.

The effective arrival client request rate $\overline{\lambda}$ into the system is different from the overall arrival client request rate and is given by

$$\overline{\lambda} = \lambda\sum_{i=0}^{K-1}(N-i)\,P_i = \lambda[N - L_s - (N-K)P_K].$$

The mean waiting time (W_q) and response time (W_s) using Little's formula can be derived as $W_q = \frac{L_q}{\overline{\lambda}}$ and $W_s = \frac{L_s}{\overline{\lambda}}$, respectively.

Using the Bayes' rule it is easy to see that for the probability of blocking, we have

$$P_B(N,c,K) = \frac{(N-K)P_K(N,c,K)}{\sum_{i=0}^{K}(N-i)P_i(N,c,K)} = P_K(N-1,c,K).$$

In particular, if $K = N$, then $\overline{\lambda} = \lambda(N - L_s) = \mu\overline{c}$, $W_s = \frac{L_s}{\lambda(N-L_s)}$, $W_q = \frac{L_q}{\lambda(N-L_s)}$ and $P_B = 0$, as it was expected.

3. PARTIAL ALLOCATION WITH BLOCKING (POLICY 2)

Policy 2 scheduling is a partial cloud approach that provides an option for a complete or partial allocation when the waiting buffer is full new client requests are denied (blocked). We adopt M/M/c/c/N queuing system for the performance analysis of the above mentioned model. The client requests are generated from N population according to quasi-random arrival process.

The one-dimensional process of the number of client requests at the service center is a birth-death process on a finite state space {0,1,….,c}. As before it is easy to see that the number of client requests in the system is a birth-death process with rates

$$\lambda_i = (N - i)\lambda, 0 \le i < c - 1,$$
$$\mu_i = i\mu, \quad 1 \le i \le c.$$

We define P_i as the steady state probability that there are i client requests in the system. By substituting the birth and death rates the state probabilities P_i can be obtained as

$$P_i = \binom{N}{i} \rho^i P_0, \quad 1 \le i \le c, \qquad (3)$$

where $\rho = \frac{\lambda}{\mu}$ is called the offered load per idle source. Using the normalization condition, we get

$$P_0 = \left[\sum_{i=0}^{c} \binom{N}{i} \rho^i\right]^{-1}.$$

Hence, $P_i = \frac{\binom{N}{i}\rho^i}{\sum_{j=0}^{c}\binom{N}{j}\rho^j}, \quad i = 0,1, \dots, c. \qquad (4)$

Remark: If we take $N\lambda = \xi$ constant and let $N \to \infty$, then

$$\binom{N}{i}\rho^i \frac{N(N-1)\dots(N-i+1)}{N^i} \frac{(N\rho)^i}{i!} \to \frac{\left(\lambda/\mu\right)^i}{i!}.$$

Thus, the Engset distribution of (4) converges to the Erlang distribution. This is shown in table 3.

Table 3: Performance indices for Engset loss system with c=6, μ=1, Nλ fixed

		N=6	N=18	N=60	N=300	N=600	N==900
	P_c	0.000008499	0.000206246	0.000398040	0.000486721	0.000498745	0.000502801
Nλ=1							
	P_B	0.000000000	0.000145135	0.000364204	0.000478576	0.000494580	0.000500004
	L_s	0.857143000	0.947238000	0.983254000	0.996202000	0.997843000	0.998391000
	P_c	0.004096000	0.062357800	0.099381100	0.113531000	0.115342000	0.115948000
Nλ=4							
	P_B	0.000000000	0.050345000	0.094840300	0.112577000	0.114863000	0.115628000
	L_s	2.400000000	3.136670000	3.414590000	3.508180000	3.519780000	3.523640000
	P_c	0.015625000	0.000206246	0.000398040	0.000486721	0.000498745	0.000502801
Nλ=6							
	P_B	0.000000000	0.000145135	0.000364204	0.000478576	0.000494580	0.000500004
	L_s	0.900000000	0.947238000	0.983254000	0.996202000	0.997843000	0.998391000

3.2 Performance indices

The various performance measures for policy-2 are as follows:

There are on average $N - L_s$ sources that are eligible to generate new requests with intensity λ/μ. Thus, the offered load (a) is expressed as

$$a = (N - L_s)\frac{\lambda}{\mu} = \rho \sum_{i-0}^{c}(N - i)\, P_i = \rho N \frac{\sum_{i=0}^{c}\binom{N-1}{i}\rho^i}{\sum_{j=0}^{c}\binom{N}{j}\rho^j}. \quad (5)$$

The carried load a_c, the expected number of servers that are occupied at given time, can be obtained as

$$a_c = \rho \sum_{i-0}^{c} N\, P_i = \rho N \frac{\sum_{i=0}^{c-1}\binom{N-1}{i}\rho^i}{\sum_{j=0}^{c}\binom{N}{j}\rho^j}. \quad (6)$$

From the equations (5) and (6) we can yield request congestion as

$$P_B(N, c) = 1 - \frac{a_c}{a} = \frac{(N - c)P_c(N, c)}{\sum_{i=0}^{c}(N - i)\, P_i(N, c)} = P_c(N - 1, c)$$

The expected number of client requests in the system (L_s) is given as

$$L_s = \sum_{i=0}^{c} iP_i = \frac{N\rho}{1+\rho}\sum_{i=1}^{c}\binom{N-1}{i-1}\left(\frac{\rho}{1+\rho}\right)^{i-1}\left(1 - \frac{\rho}{1+\rho}\right)^{N-i}.$$

The utilization of the system (U_c) and the mean number of busy servers ($\bar{c}$) can be calculated as

$$U_c = 1 - P_0, \quad \bar{c} = L_s.$$

The utilization of the server (U_s) and the utilization of the sources (U_t) are respectively, can be calculated as

$$U_s = \frac{\sum_{i=1}^{c} iP_i}{c} = \frac{\bar{c}}{c}, \quad U_t = \frac{N - L_s}{N}.$$

The blocking probability or call congestion that is the probability that a client request finds the system is full at his arrival, by the help of Bayes' theorem can be computed as

$$P_B(N, c) = \frac{(N-c)P_c(N,c)}{\sum_{i=0}^{c}(N-i)P_i(N,c)} = P_c(N - 1, c).$$

Let $E(N, c, \rho)$ denote the blocking probability, that is $E(N, c, \rho)$ $= P_c(N - 1, c)$ which is called Engset's loss formula. This can be solved numerically by the following recursion:

$$E(N, c, \rho) = \frac{\binom{N-1}{c}\rho^c}{\sum_{i=0}^{c}\binom{N-1}{i}\rho^i} = \frac{\binom{N-1}{c-1}\frac{N-c}{c}\rho^c}{\sum_{i=0}^{c-1}\binom{N-1}{i}\rho^i + \binom{N-1}{c-1}\frac{N-c}{c}\rho^c} = = \frac{\frac{N-c}{c}\rho E(N,c-1,\rho)}{1+\frac{N-c}{c}\rho E(N,c-1,\rho)} = \frac{(N-c)\rho E(N,c-1,\rho)}{c+(N-c)\rho E(N,c-1,\rho)}.$$

The initial value is $E(N, 1, \rho) = P_1(N - 1,1) = \frac{(N-1)\rho}{1+(N-1)\rho}$.

The effective arrival client request rate $\bar{\lambda}$ into the system is thus different from the overall arrival client request rate and is given by

$$\bar{\lambda} = \lambda \sum_{i=0}^{c-1}(N - i)\, P_i.$$

The mean response time (W_s) using Little's rule can be given as $W_s = \frac{L_s}{\bar{\lambda}}$.

In particular, when c = N, it is easy to see that $L_s = \frac{N\rho}{1+\rho}$ and thus $U_s = \frac{\rho}{1+\rho}$,

$\bar{m} = \frac{N}{1+\rho}$, $U_t = \frac{1}{1+\rho}$ and $P_B = 0$ which was expected.

4. NUMERICAL ILLUSTRATION

In this section, we illustrate the numerical tractability that shed a light on the performance aspect of cloud computing which is of crucial interest for both cloud providers and cloud customers.The graphs for policy-1 are shown in Figure 3 to figure 7.

Figure 3 depicts the effect of λ on the expected number of client requests in the system ($L_{s)}$ for various buffer sizes. The parameters are taken as N = 30, c=5 and μ = 10.0. It is seen that as λ increases L_s increases monotonically. For fixed λ , L_s decreases as the buffer size increases in the system. Figure 4 plots the impact of λ on the average waiting time in the buffer for different VMs. It can be observed that the waiting time in the buffer W_q increases with the increase in λ. As the number of VMs increases the average waiting time decreases which indicate that the server is available in the system more frequently, clearing the accumulated requests and as a result W_q decreases.

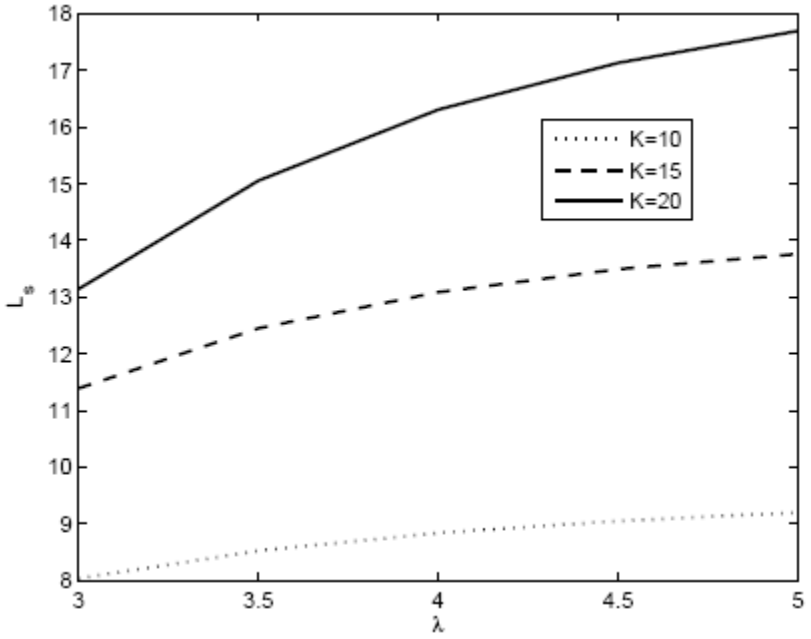

Figure 3 Impact of L_s on λ

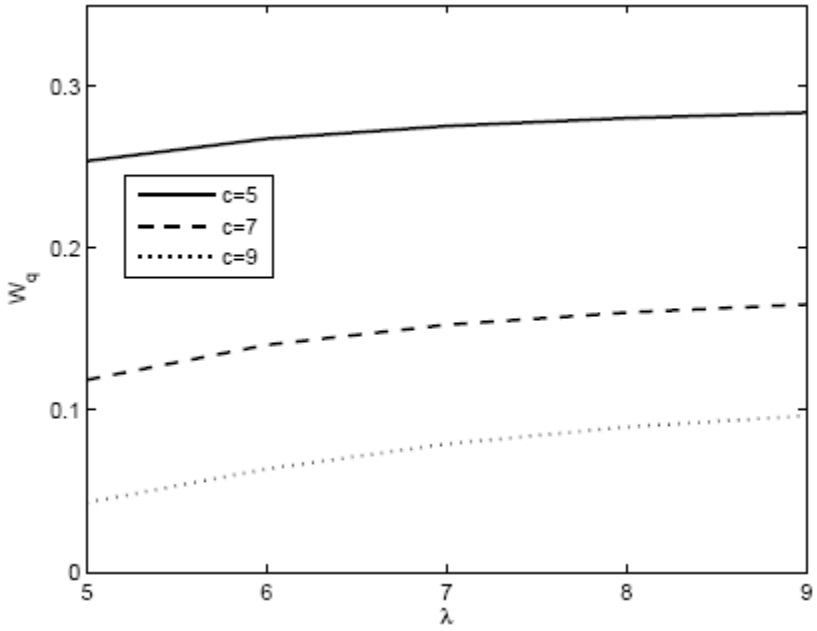

Figure 4 Impact of W_q on λ

Figure 5 shows the impact of mean number of idle server's ($\overline{S}$) on the buffer size in the cloud system for various ρ. It can be observed that as ρ increases the mean number of idle servers' decreases. As expected for fixed ρ, the mean number of idle servers decreases as the buffer size in the cloud system increases. The effect of population size on the utilization of sources for various numbers of VMs c in the cloud system is

presented in Figure 6. We see that for all the cases as population size increases, the utilization of sources also increases for a fixed number of VMs. One can also observe that as number of VMs c increases the utilization of sources also increases.

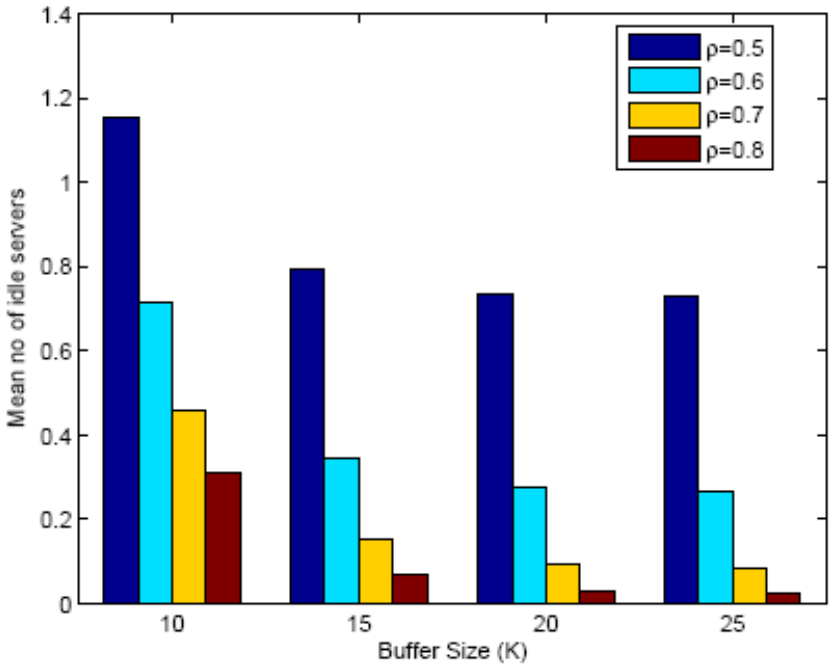

Figure 5 Impact of $\overline{S}$ on *K*.

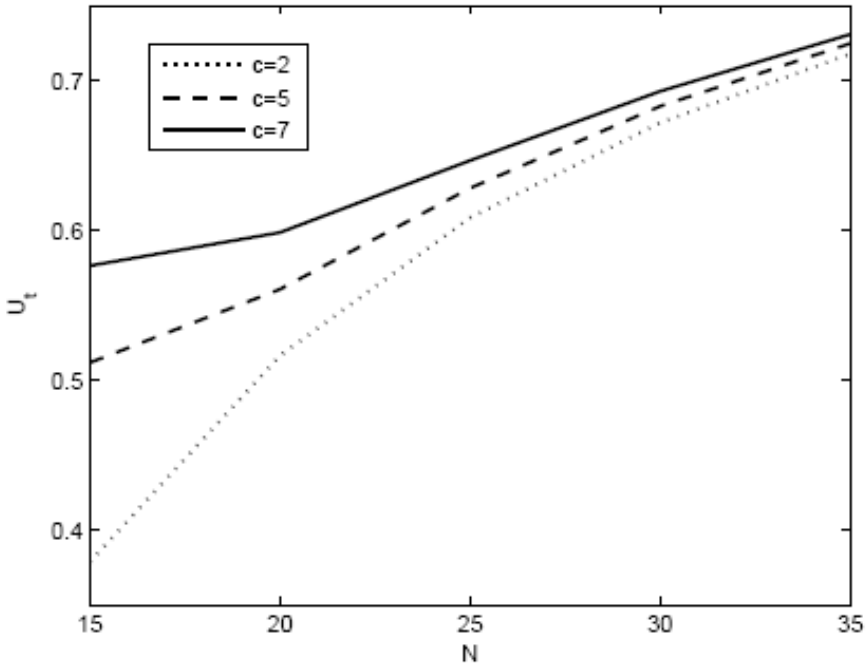

Figure 6 Impact of N on U_t.

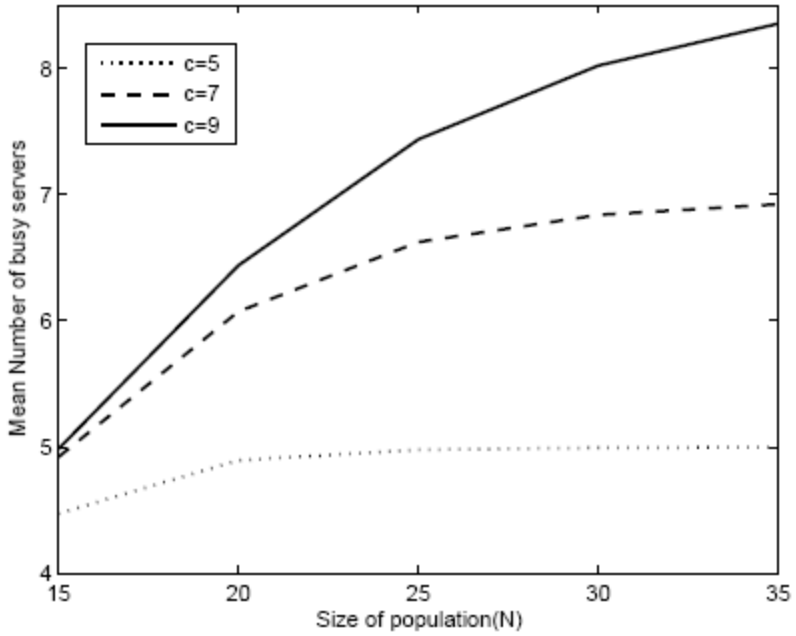

Figure 7 Effect of N on $\bar{c}$.

Figure 7 illustrates the impact of the population size on the mean number of busy servers for various virtual machines. It can be observed that when population size increases the mean number of busy servers in the system increases. For a fixed population size, the mean number of busy servers increases as the number of VMs c increases. The graphs are presented in figures 8 to 10 for policy-2. Figure 8 depicts the effect of λ on the expected number of client requests in the system Ls for various VMs c. The parameters are taken as N = 30 and μ = 10.0. It is seen that as λ increases Ls increases monotonically. For fixed λ, Ls decreases as c increases in the system. Figure 9 illustrates dependence of server utilization on ρ varying from 0.5 to 0.9 and c varying from 6 to 14. We observed that for fixed c, as ρ increases server utilization U_s increases. Further with fixed ρ, the server utilization U_s decreases as c increases.

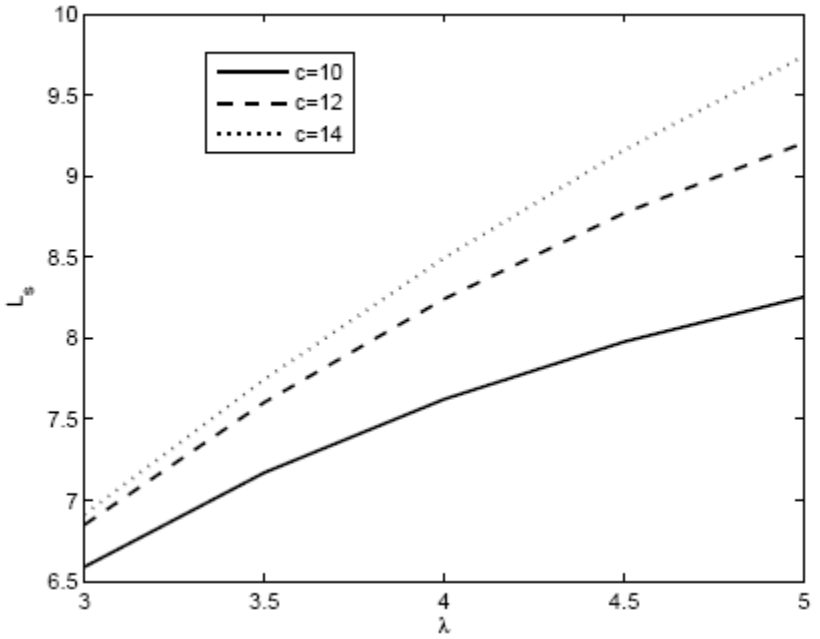

Figure 8 Effect of λ on L_s

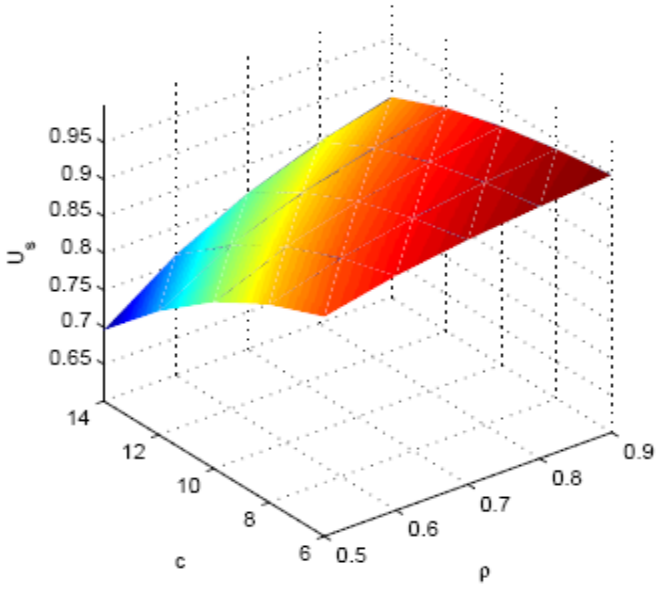

Figure 9 Server utilization U_s for various values of ρ and c

Figure 10 depicts the effect of probability of blocking (PBL) on ρ for various VMs c. It is seen that as ρ increases PBL increases monotonically. For fixed ρ, blocking probability decreases as c increases in the system.

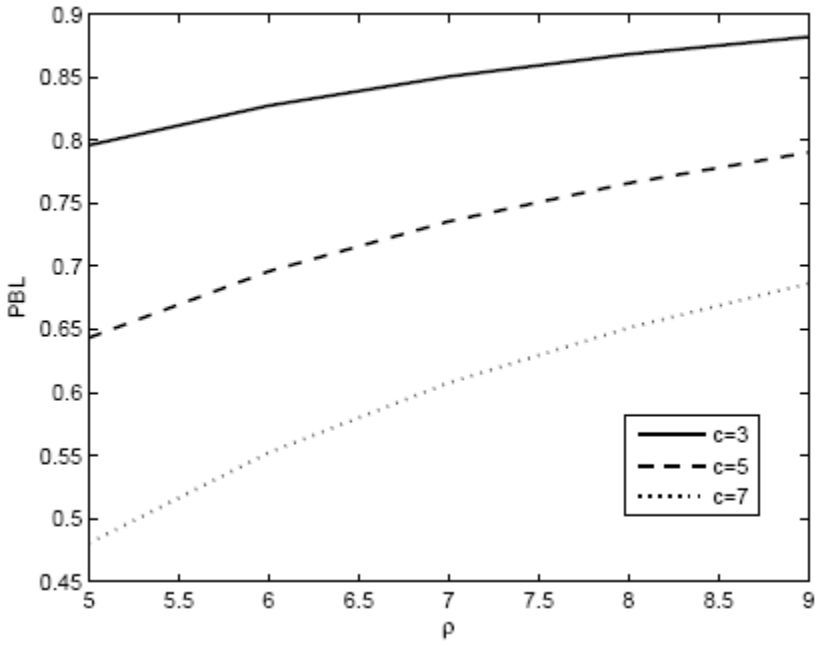

Figure 10 Impact of ρ on PBL.

Figure 11 shows the impact of ρ on L_s for both the policies. It can be seen that L_s increases as ρ increases. But, L_s is less in policy 2 as compared to policy 1. Utilization of the sources U_t versus N is depicted in Figure 12 for both the policies. It can be observed that policy 2 utilizes the resources optimally.

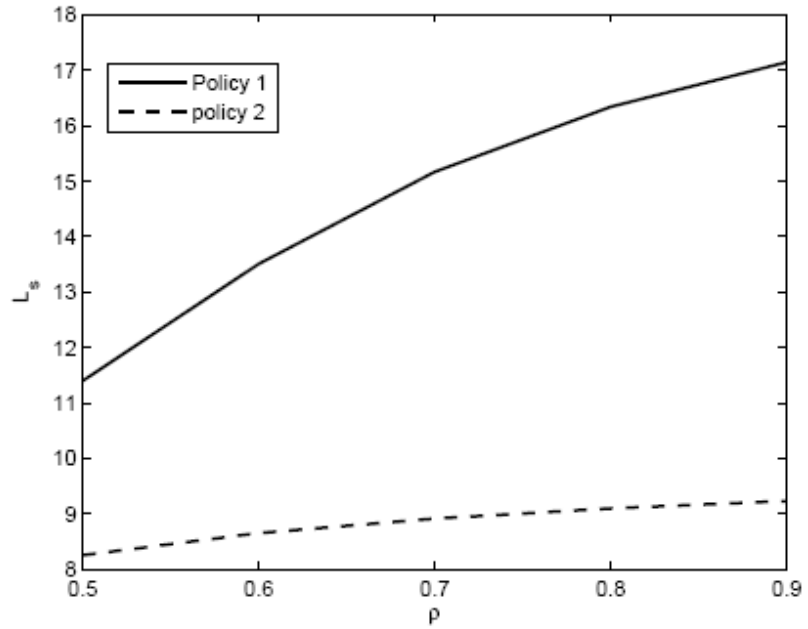

Figure 11 Impact of ρ on L_s for different policies.

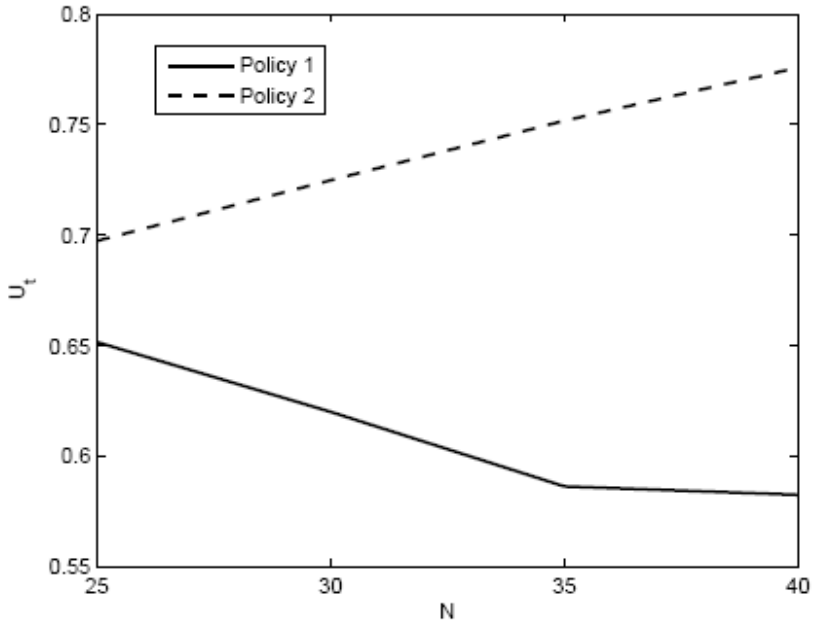

Figure 12 Effect of N on U_t for different policies.

5. CONCLUSION

In this paper, we have proposed two scheduling policies along with an analytical resource prediction model for private cloud system. Finite source finite buffer multiple server queueing model is applied to provide performance metrics of a private cloud computing system. We have developed a recursive method using the birth-death process, to obtain the steady-state system length distributions.Various performance measures such as utilization of the system, utilization of server, the expected number of client requests in the system, the expected number of client requests in the queue, mean number of idle servers, mean waiting time, response time and blocking probability are also carried out. The numerical computations under a range of parameters show that the proposed model improve the system performance and can optimize the organizational resources in cloud computing system .

REFERENCES

[1] R. Buyya, C. Yeo, S. Venugopal, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility,," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599– 616, 2009.

[2] [Online]. Amazon elastic compute cloud (amazon ec2), Available: http://aws.amazon.com/ec2/ ,2009

[3] [Online]. Microsoft windows azure ,Available: http://aws.microsoft.com/windowsazure/ ,2010

[4] [Online]. Google app engine, Available: http://www.code.google.com/appengine/ ,2011

[5] [Online]. Ibm smart business cloud computing,Available: http://www.ibm.com/ibm/cloud/ ,2010

[6] A. Ganapathi, H.Kuno, U.Daval, J.Wiener, A.Fox, M.Jordan, and D. Patterson, "Predicting multiple performance metrices for queries: Better decisions ended by machine learning," in *Proc. IEEE International Conference on Data Engineering (ICDE'09*, pp. 592–603, *)*, 2009.

[7] S. Venugopal, R. Buyya, and L. Winton, "A grid sevice broker for scheduling e-science applications on global data grids," *Concurrency and Computation: Practice and Experience - Middleware for Grid Computing*, vol. 18, no. 6, pp. 685–699, Feb. 2006.

[8] B. Li, J. Li, J. Huai, T. Wo, Q. Li, and L. Zhong, "Enacloud: An energysaving application live placement approach for cloud computing environments," in *Proc. IEEE International Conference on Cloud Computing (CLOUD'09)*, pp. 17–24, 2009

[9] J.H.Schiller, *Cloud Computing Principles and Paradigm*. New Jersey: John Wiley and Sons, 2011.

[10] G. Li, H. Sun, H. Gao, H. Yu, and Y. Cai, "A survey on wireless grids and clouds," in *Proc. IEEE International Conference on Grid and Cooperative Computing*, pp. 261–267,2009

[11] B. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *Proc. IEEE 5th International Joint Conference on INC, IMS and IDC*, pp. 44–51,2009.

[12] H. Yuan, C. Kuo, and I. Ahmad, "Energy efficiency in data centers and cloud-based multimedia services: An overview and future directions," in *Proc. IEEE International Conference on Green Computing*, pp.375–382,2010

[13] W. Lin and D. Qi, "Research on resource self-organizing model for cloud computing," in *Proc. IEEE International Conference on Internet Technology and Applications*, pp. 1–5,2010.

[14] F. Teng and F. Magoules, "Resource pricing and equilibrium allocation policy in cloud computing," in *Proc. IEEE International Conference on Computer and Information Technology (CIT 2010)*, pp. 195–202, 2010

[15] [Online]. RightScale White Paper. Designing private and hybrid clouds: Architectural best practices. Available: http://www.rightscale.com/lp/private-hybrid-cloud-white-paper.php ,2012

[16] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, and R. Katz, "Above the clouds:a berkeley view of cloud computing," *UC Berkeley Reliable Adaptive Distributed Systems Laboratory White Paper*, 2009.

[17] S. Ramgovind, M. Eloff, and E. Smith, "The management of security in cloud computing,," *Information Security for South Africa(ISSA)*, 2010.

[18] [Online]. Available: http://en.wikipedia.org/wiki/ .

[19] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual infrastructure management in private and hybrid clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14–22, 2009.

[20] P. T. Jaeger, J. Lin, J. M. Grimes, and S. N. Simmons, "Where is the cloud?geography, economics, environment, and jurisdiction in cloud computing," *First Monday*, vol. 14, pp. 4–5, 2009.

[21] H. Kobyashi and B. L. Mark, *System Modeling and Analysis*. New Jersey: Pearson Education, 2008.

[22] J.Sztrik, *Basic Queueing Theory*. University of Debrecen: Faculty of Informatics, 2011.