Simulation Based Analysis of Infrastructural Framework for Large Scaled Applications in Cloud

Minu Bala
Assistant Professor, Department of Higher Education,
J&K Govt, Jammu, India
(E-mail: ind_minu@yahoo.com)

***Abstract*—** No doubt cloud computing has decreased the infrastructural problems of Application Developers by providing pay- per- use flexible elastic infrastructural services. But it is very difficult for the application developers to assess in advance how there application is going to perform in the real cloud environment. There are many important parameters like response time, processing time, cost etc. which need to be taken care of, before deploying an application on the real cloud environment. Simulation tool can be very helpful in assessing the performance of an application under different infrastructural configurations and can help in finding an optimal model for a particular application. Tool based simulation enables seamless modeling, simulation and experimentation of emerging cloud computing infrastructures and management service. Hence tool based simulation of cloud computing environment may help the users to access and deploy applications from anywhere in the world, on demand, at competitive cost, depending on the user quality of service requirement. The present study has been made using CloudAnalyst: A CloudSim-based tool for modeling and analysis of large scale cloud computing environments. The experimental results reveal that enhancement in the infrastructural resources increase the cost for cloud computing customers (i.e. application providers) but decreases the overall response time of the end users of the application.

Keywords*—Cloud Computing, IaaS, PaaS, SaaS, Virtual Machine, Load Balancing, Simulation.*

I. INTRODUCTION

The rapid development of processing and storage technologies and the success of the internet has enabled the realization of new computing model called Cloud Computing [1], in which resources are provided as general utilities that can be leased and released by users through the internet in an on-demand fashion. Clouds offer services[2] that can be grouped into three categories: Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS).

***Infrastructure as a Service*:** IaaS refers to on-demand provisioning of infrastructural resources, usually in terms of VMs. The cloud owner who offers IaaS is called an IaaS provider. Examples of IaaS providers include Amazon EC2, GoGrid etc.

***Platform as a Service*:** PaaS refers to providing platform layer resources, including operating system support and software development frameworks. Examples of PaaS providers include Google App Engine, Microsoft Windows Azure etc.

***Software as a Service*:** SaaS refers to providing on demand applications over the Internet. Examples of SaaS providers include Google Apps, Facebook etc. Cloud Computing can be divided into there parts: cloud computing providers, cloud computing customers and end users. Cloud service providers own the physical resources as datacenters and virtualization technology. Cloud computing customers use these resources to provide services to customers. And end users use these services. For example, any online newspaper uses Amazon EC2 for hosting their website. Here Amazon EC2 is the cloud computing provider and newspaper is the cloud customer. And newspaper readers are the end users. With cloud computing, new possibilities have been opening up on how applications can be built on the Internet. On one hand there are the cloud service providers who are willing to provide large scaled computing infrastructure at a cheaper price which is often defined on usage, eliminating the high initial cost of setting up an application deployment environment, and provide the infrastructure services in a very flexible manner which the users can scale up or down at will. On the other hand there are large scaled software systems such as social networking sites and e-commerce applications gaining popularity today which can benefit greatly by using such cloud services to minimize costs and improve service quality to the end users.

II. PROBLEM DEFINITION

The most difficult part in cloud computing is to deploy an application in real environment. It is very difficult to know the exact cost and its requirement until and unless we buy the service. Not only this, it is not known that whether it will support the existing application which is available on traditional datacenter or had to design a new application for the cloud computing environment. The response time, the processing time, cost etc. are some parameters which we need to take care of before deploying an application.

KEY FACTORS AFFECTING THE PERFORMANCE OF LARGE SCALED APPLICATIONS

A. Configuration of Datacenter

Datacenter is characterized by different hardware specifications like no. of hosts or servers in datacenter, no. of

processors in each host, processor speed, no. of VMs allocated to each host etc. All these parameters are very import as they directly affect the performance of an application on cloud.

B. Instances of Application

The cloud computing customer can deploy its application on a single datacenter or can deploy its application in multiple instances in Web Farm. In Web Farm, datacenters can be homogenous or heterogeneous. The performance of an application is directly affected by no. of instances of the cloud computing service provider, which the customer opts.

C. Geographical Location of Datacenters and User Groups

The geographical location of service provider and the end users of the application are also important factors in large scaled applications. The overall performance of large scaled applications on the cloud is affected by different network issues and the location of the user groups. Amazon EC2 [3] provides the ability to place instances in multiple locations. It is currently available in nine regions: USEast (Northern Virginia), USWest (Oregon), USWest (Northern California), EU (Ireland), Asia Pacific (Singapore), Asia Pacific (Tokyo), Asia Pacific (Sydney), South America (Sao Paulo) and AWS GorCloud. By launching instances of an application from different regions, failure of an application from single region can be protected.

D. VM Allocation Policy

VM allocation policy depicts the method of assigning VMs to jobs. VM allocation policy could be time shared or space shared. In time shared policy, multiple jobs at a time can be assigned to VMs on time sharing basis where as in space shared policy only one job is assigned to a VM at a time. The other jobs in the system will be in the waiting list.

E. Service Broker Policy [4]

In cloud datacenter, Service Broker mechanism is responsible for sending requests coming from users belonging to different user groups present at different geographical locations across the globe to different datacenters in the cloud. The Service Broker Policy can be broadly categorized as following:

Closest Datacenter Policy

In this Policy, the Service Broker sends the request to closest datacenter in terms of Network Latency.

Performance Optimization Policy

In this policy, the service broker actively monitors all the datacenters and sends the request to the datacenter which gives best response time to the end user at the time it is queried.

Dynamic Configuration Policy

This policy handles the dynamic behavior of the environment. The service broker in this case is allotted an extra work of enhancing the application‘s instances as per the workload faced by the application. It adds or removes VMs dynamically in physical machines in cloud datacenter as per the workload faced by the applications at different instances of time.

F. VM Load Balancing Policy [5]

VM Load Balancing policy is responsible for balancing load among various VMs. The Datacenter Controller (DCC) module receives requests from users in the form of cloudlets which are further routed the to the appropriate VM using the VM Load Balancing Policy for processing.

Round Robin Policy

In Round Robin Policy, the requests of the clients are handled in a circular manner on first come first bases. The users' requests are directly handled by the Datacenter Controller module which further routes the requests to the Load Balancer (LB). The LB is an entity which has full information about all available VMs and next VM to which next incoming job is to be assigned. The incoming jobs are assigned to VMs in a circular way.

Throttled Policy

In throttled policy only one job is allowed to be assigned to each VM at a time. The next job is assigned to VM only on the successful completion of first job on VM. The LB maintains an index table which records full information (Status – Busy/Free) about all VMs in the environment. It tells whether a particular VM is Busy or Free. The incoming request coming from the clients are forwarded by the DCC to the LB for further assignment of VMs. The LB sees its index table starting from the beginning to its end for any free VM. It sends the id of first free VM found in its index table to the Datacenter Controller for assignment of job. The DCC sends back the acknowledgement signal and accordingly LB updates the status of VM as Busy in its index table. If LB does not find any free VM, it sends a null signal to DCC which then keeps the user request in waiting queue. As some VM finishes its job, the LB sends a signal to DCC about the free VM and DCC assigns the waiting job the VM. The LB updates its index table on allocation of jobs to VMs and de-allocation of jobs from VMs.

Active Monitoring Policy

Load is evenly distributed among all VMs in different physical machines in a cloud datacenter. A load monitoring mechanism continuously keeps an eye on the workload faced by each VM in the system. In this policy, load is equally distributed among all the VMs by actively monitoring the load on all the VMs. The LB keeps track of number of jobs assigned to each VM in the system. Whenever LB gets a signal from DCC for the allocation of VM for the new job, it first parses the index table from top until the least loaded VM is found and returns the VM which is least loaded. If there is more than one found, it uses first come first serve (FCFS) basis to choose the least loaded and returns the id to DCC. The DCC notifies the Load Balancer about the new allocation. It updates the index table by increasing the allocation count by 1 for that VM. When any VM completes the job assigned to it, it signals to DCC which further signals to LB and it accordingly updates the index table by decreasing the allocation/ load count of that particular VM by 1.

III. RELATED WORK

There are many simulation techniques to investigate behavior of large scaled distributed systems, as well as tools to support the research work. Some of these simulators are GridSim[6], GangSim[7], CloudSim[8], CloudAnalyst[9]. GridSim toolkit was developed to address the problem of performance evaluation of real large scaled distributed environments (typically Grid systems), CloudSim enables modeling, simulation and experimenting on cloud computing infrastructures.It is built on top of GridSim toolkit. CloudAnalyst is developed using CloudSim toolkit, leveraging the features of the original framework and extending some of the features of cloudSim. The researchers have used these simulation tools for modeling the behavior of newly designed policies, infrastructural models etc.

Jingsong Wang et [10] al. have used simulations to assess the stability and capacity of cloud computing systems using their own simulation program. The logical stability of the cloud under various configurations has been assessed and the correctness of the simulation is verified by the theoretical results of M/M/1 queuing system.

R. Jeyarani et [11] al. suggested two- level scheduler which concentrates on optimizing the system throughput by maximizing the over all utilization of resources and guaranteeing expansion in performance of the application. The study has been done using CloudSim and it (cloudsim toolkit) has also been extended by implementing a novel high-level meta-scheduler.

IV. PROPOSED STUDY

Quantifying the performance of different provisioning policies in a real cloud computing environment for different application models under variant conditions is extremely challenging due to the rigidity of the real infrastructure. Further, it is tedious and time consuming to reconfigure benchmarking parameters across massive scale cloud computing infrastructure over multiple test runs. So, the proposed study is to assess the optimal infrastructural model for a cloud computing customer i.e. application developer for launching its application on real cloud environment. The study will be made for different cloud scenarios by varying no. of datacenters, hosts/servers, VMs, service broker policies, load balancing policies etc. during each simulation run. By simulation we can understand the real environment of cloud computing and after having optimal results, we can deploy our application in cloud computing environment.

V. SIMULATION

The simulation tool provides the repeatable and controlled environment to setup the data center configuration, cloud configuration and internet characteristics for the cloud tasks.

A. Simulation Tools

CloudAnalyst is a GUI based simulator for modeling and analysis of large scaled applications. It is built on top of CloudSim toolkit, by extending CloudSim functionality with the introduction of concepts that model Internet and Internet Application behaviors. Through GUI, its different components like Userbase, Internet, Service Broker Policy, Internet Cloudlet, Datacenter Controller and VM Load Balancing Policies can be configured differently for different cloud scenarios.

B. Simulation Setup

Large Scaled Applications that could be benefited from the cloud computing are Social Networking Applications, e-Commerce Applications, Online Education Applications etc. For the present study, a Social Networking Application – Facebook has been considered. The approximate users of Facebook[12] distributed across the globe as on 31-03-2012 are as under:

Table 1. Registered users of FB as on 31-03-2012

Geographic Region in order of size	**Registered users**
Europe	232,835,740
Asia	195,034,380
North America	173,284,940
South America	173284940
Central America	41,332,940
Africa	40,205,580
Middle East	20,247900
Oceania/Australia	13,597380
Caribbean,the	6355320

For the simulation purpose, the whole globe has been divided into six regions as R0, R1, R2, R3, R4 and R5. And grouping of registered users of FB as Userbases is done as under:

Table 2. Grouping of Regions & Userbases

Regions	**Names of Geographic Regions**	**Userbase**	**Registered users**
R2	Europe	UB3	232835740
R3	Asia, Middle East	UB4	215282280
R0	North America, Central America, Caribbean,the	UB1	220973200
R1	South America	UB2	112531100
R4	Africa	UB5	40205580
R5	Oceania/Australia	UB6	13,597380

In order to bring the simulation framework more close to the real environment, the network behavior of the simulation model as been

configured as per the network characteristics of Amazon EC2. It has been assumed that only 5% of the registered users remain online during peak hours and 1/10th of peak hour users will remain online during off-peak hours. It is further assumed that each user makes a new request after every 5 minutes when online.

The following parameters haven kept fixed for all simulation scenarios.

Table 3. Parameters fixed for simulations

Parameter	Value
Simulation duration	60 min.
Requests per user per hr.	60
Data Size per request per hr.	100 bytes
User Grouping factor in Userbases	10000
Request Grouping factor in Datacenters	1000
Executable instruction length per request	500 bytes

C. Scenario Setup & Observations

The scenarios considered in this experimental work are depicted in table 4. For each cloud scenario, simulation runs have been made using CloudAnalyst simulator and results obtained are depicted in table 5, 6 &7. The parameters which have been observed during each simulation run are Overall Response Time of the datacenter and the datacenter cost. The datacenter cost (in $s) is composed of two elements: cost of VMs and data transfer cost

Table 4. Description of simulation scenarios

S.No.	No. of Data centers	No. of Hosts	No. of VMs	Broker Policy
1.	1	40	150,125, 100,75,50, 25,10	Optimize Response Time
2.	1	40	150,125, 100,75,50, 25,10	Reconfigure Dynamically the load
3.	1	20	150,125, 100,75,50, 25,10	Optimize Response Time
4.	1	20	150,125, 100,75,50, 25,10	Reconfigure Dynamically the load
5.	1	20	150,125, 100,75,50, 25,10	Optimize Response Time
6.	1	20	150,125, 100,75,50, 25,10	Reconfigure Dynamically the load

Table 5. Overall Response Time of a datacenter and datacenter cost when no. of host/servers is 10

No. of VMs	Broker Policy: Optimize Response Time		Broker Policy: Refigure Dynamic with load	
	Overall Avg. Response Time	Cost incurred in $s	Overall Avg. Response Time	Cost incurred in $s
150	1085.08	2389.89	1085.17	2389.89
125	1060.63	2387.44	1060.76	2387.39
100	1048.18	2376.94	1049.44	2376.99
75	1025.60	2373.43	1236.09	380.22
50	1010.60	2379.91	1230.65	379.11
25	1468.23	2377.4	2415.44	135.68
10	3811.83	2375.89	7583.33	93.13

Table 6. Overall Response Time of a datacenter and datacenter cost when no. of host/servers is 20

No. of VMs	Broker Policy: Optimize Response Time		Broker Policy: Refigure Dynamic with load	
	Overall Avg. Response Time	Cost incurred in $s	Overall Avg. Response Time	Cost incurred in $s
150	670.17	2421.34	670.51	2421.28
125	660.31	2418.83	660.68	2418.78
100	653.11	2416.32	653.96	2416.37
75	666.01	2413.81	725.72	498.37
50	850.55	2411.3	1000.12	385.92
25	1453.90	2408.79	2406.30	136.82
10	3785.73	2407.29	7609.29	93.70

Table 7. Overall Response Time of a datacenter and datacenter cost when no. of host/servers is 40

No. of VMs	Broker Policy: Optimize Response Time		Broker Policy: Refigure Dynamic with load	
	Overall Avg. Response Time	Cost incurred in $s	Overall Avg. Response Time	Cost incurred in $s
150	509.41	2421.34	509.62	2421.28
125	537.07	2418.83	537.35	2418.78
100	583.58	2416.32	582.64	2416.37
75	666.06	2413.81	725.30	506.54
50	850.52	2411.27	999.93	385.92
25	1453.96	2408.79	2405.89	136.82
10	3785.72	2407.29	7609.62	93.70

D. Graphical Representation and Analysis

The graphical representations of experimental observations mentioned in table 5 are depicted in Fig. 1 &2.

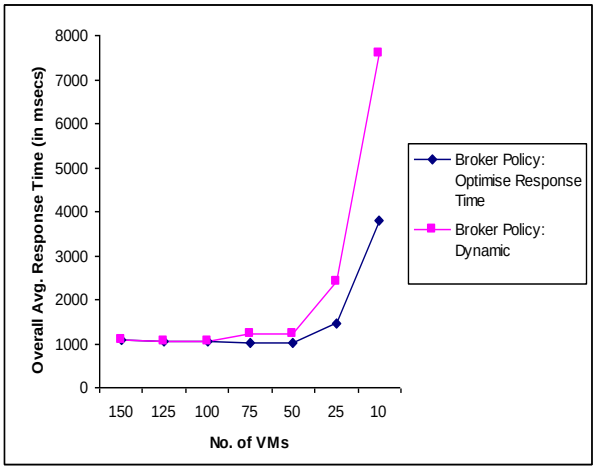

Fig.1 Overall Avg. Response Time of a Datacenter at different no. of VMs using Optimize Response Time & Dynamic reconfiguration Policy at Broker Level and Throttled LB Policy at VM level

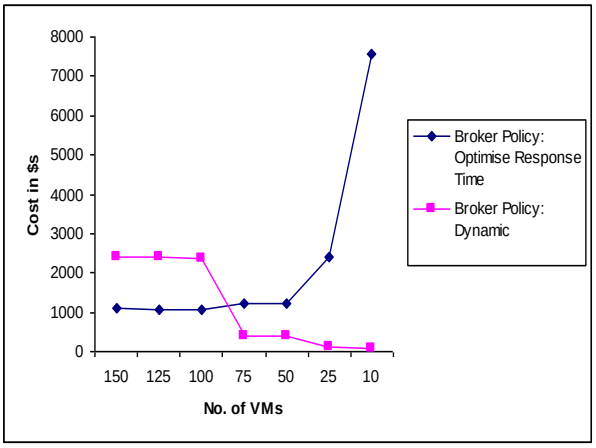

Fig.2 Datacenter Cost incurred at different no. of VMs using Optimize Response Time & Dynamic reconfiguration Policy at Broker Level and Throttled LB Policy at VM level.

From the graph it is clear that

a) No. of VMs in between 50 - 75 is the best slot for the above considered parameters as the response time is almost minimum and stable in this slot .

b) The response time is minimum when the service broker policy is Optimize Response Time.

c) The response time increases with the decrease in no. of VMs. from 50. The response time increases more drastically in case of dynamic service broker policy.

d) The response time again increases slowly with the increase in VM from 75.

e) Datacenter cost is also optimal in 50 –70 slot of VMs.

The graphical representation of experimental observations mentioned in table 6 are depicted in Fig. 5 & 6.

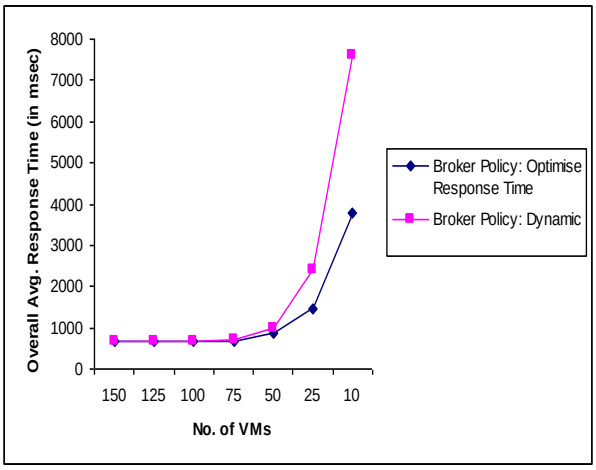

Fig.3 Overall Avg. Response Time of a Datacenter at different no. of VMs using Optimize Response Time & Dynamic reconfiguration Policy at Broker Level and Throttled LB Policy at VM level

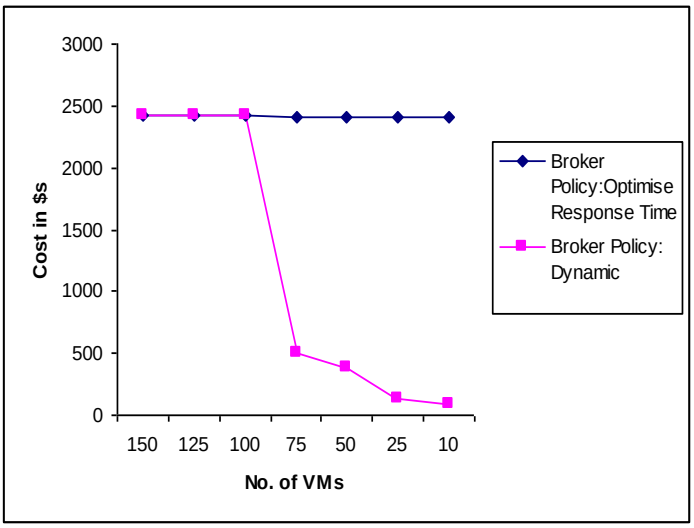

Fig.4 Datacenter Cost incurred at different no. of VMs using Optimize Response Time & Dynamic reconfiguration Policy at Broker Level and Throttled LB Policy at VM level.

From the graph it is clear that

a) No. of Vms in between 100 -75 is the best slot for the above considered parameters as the response time is almost minimum and stable in this slot.

b) The response time increases with the decrease in no. of VMs. from 75. The response time increases more drastically in case of dynamic service broker policy.

c) The response time again increases slowly with the increase in VM from 100.

d) Datacenter cost is also optimal in 100 -75 slot of VMs.

The graphical representation of experimental observations mentioned in table 7 are depicted in Fig. 5 &6.

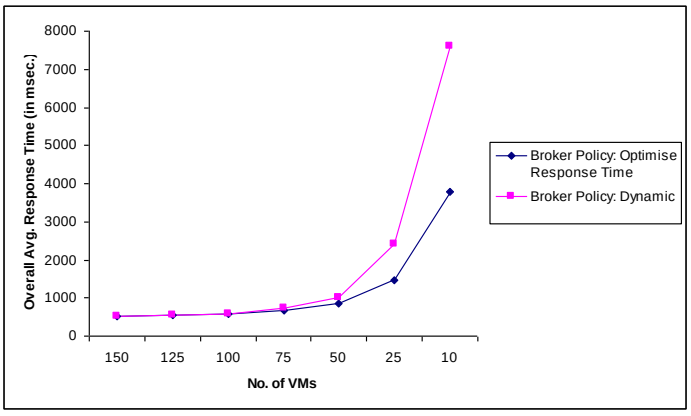

Fig.5 Overall Avg. Response Time of a Datacenter at different no. of VMs using Optimize Response Time & Dynamic reconfiguration Policy at Broker Level and Throttled LB Policy at VM level

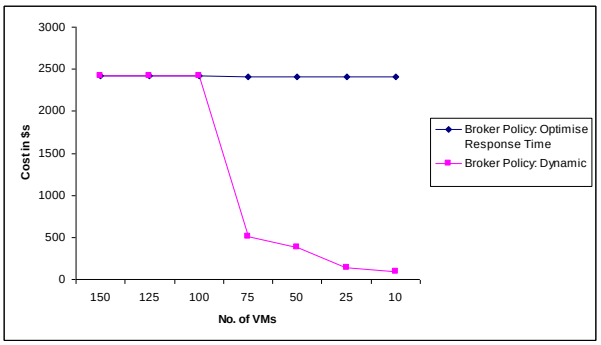

Fig.6 Datacenter Cost incurred at different no. of VMs using Optimize Response Time & Dynamic reconfiguration Policy at Broker Level and Throttled LB Policy at VM level.

From the graph it is clear that

a) No. of Vms in between 100 -75 is the best state for the above considered fixed parameters as the response time is almost minimum and stable in this slot .
b) The response time increases with the decrease in no. of VMs. from 75. The response time increases more drastically in case of dynamic service broker policy.
c) The response time again increases slowly with the increase in VM from 100.
d) Datacenter cost is also optimal in 100 -75 slot of VMs

The overall observation from the graphs and tables is that scenario no 2 i.e. a datacenter center with 40 hosts and 75 VMs, dynamic broker policy is the optimal infrastructural framework for an application that we have considered.

VI. CONCLUSION

It is clear from the above results that expansion in the infrastructural resources improves the results. Overall processing time decreases as we increase no. of virtual machines in a datacenter but it adds up to the cost. In cloud based application, cost is composed of VM cost and data transfer cost. It has already been analyzed in the previous work[13] that Geographical location of datacenter and Userbase affects the services. Bringing the services closer to the users improves the quality of service. Service quality can be further improved by application of different load balancing tactics at the application level and at the VM level. Thus the overall response time of the application to the end users can be optimized by making a right combination of the above mentioned elements. Moreover such type of simulation work helps to generate a valuable insight for Application Designers in identifying the optimal configuration for their application. The cloud computing customers can analyze the performance of their application in different scenarios using different policies and can assess the best for their application.

VII. REFERENCES

[1] Vaquero, Luis M., Luis Rodero-Merino, Juan Caceres, and Maik Lindner. "A break in the clouds: towards a cloud definition." *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, pp. 50-55, 2008.

[2] Buyya, Rajkumar, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility." *Future Generation computer systems, vol.* 25, no. 6, pp 599-616, 2009.

[3] Amazon Elastic Compute Cloud (EC2), http://www.aws.amazon.com/ec2 / [18 April 2010].

[4] Buyya, Rajkumar, Rajiv Ranjan, and Rodrigo N. Calheiros. "Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities." In *High Performance Computing & Simulation, 2009. HPCS'09. International Conference on*, pp. 1-11. IEEE, 2009.

[5] Jasmin James and Dr. Bhupendra Verma, "Efficient VM load balancing algorithm for a cloud computing environment", International Journal on Computer Science and Engineering (IJCSE), vol. 4, no. 9, pp.1658-1663, Sep 2012.

[6] R. Buyya and M. Murshed, "GridSim: a toolkit for the modeling and simulation of distributed resource management and scheduling for Grid computing", Concurrency and Computation: Practice and Experience, vol. 14, issue (13-15), pp. 1175-1220, 2002

[7] Dumitrescu, Catalin L., and Ian Foster. "GangSim: a simulator for grid scheduling studies." In *Cluster Computing and the Grid, 2005. CCGrid 2005. IEEE International Symposium on*, vol. 2, pp. 1151-1158. IEEE, 2005.

[8] http://www.cloudbus.org/cloudsim/

[9] Wickremasinghe, Bhathiya, Rodrigo N. Calheiros, and Rajkumar Buyya. "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications." In *Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on*, pp. 446-452. IEEE, 2010.

[10] Jingsong Wang and Michael N. Huhns. "Using simulations to assess the stability and capacity of cloud computing systems." In *Proceedings of the 48th Annual Southeast Regional Conference*, pp. 74-81. ACM, 2010.

[11] Jeyarani, R., R. Vasanth Ram, and N. Nagaveni. "Design and Implementation of an Efficient Two-Level Scheduler for Cloud Computing Environment." In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, pp. 585-586. IEEE Computer Society, 2010.

[12] www.internetworldststs.com/facebook.htm

[13] Bala, Minu and Devanand. "Performance Evaluation of Large Scaled Applications using Different Load Balancing Tactics in Cloud Computing." *International Journal of Computer Applications, vol.* 76, no. 14 (2013).

Minu Bala is working as an Assistant Professor in Computer Applications with Higher Education Department, Jammu and Kashmir Government, Jammu, India. She completed her MCA from University of Jammu in 1999 and is Post Graduate in Mathematics also. Presently, she is pursuing her Ph.D. from University of Jammu under Teacher Fellowship Programme, sanctioned by University Grants Commission, New Delhi. Her field of research is cloud computing.