ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, *January 2015*

CloudSim-A Survey on VM Management Techniques

Seema Vahora[1], Ritesh Patel[2]

Student, U & P U. Patel Dept. of Computer Engineering, C.S.P.I.T., CHARUSAT, Changa, Gujarat., India[1]

Associate Professor, U & P U. Patel Dept. of Computer Engineering, C.S.P.I.T., CHARUSAT, Changa, Gujarat., India[2]

Abstract*:* With the initiation of internet in the 1990s to the present day facilities of universal computing, the internet has reformed the computing world in a drastic way. It has travelled from the concept of parallel computing to distributed computing to cluster computing to grid computing to utility computing to virtualization and recently to cloud computing, in future Internet of Things. Virtualization and utility computing can be stated as key concept of cloud. As cloud computing can be specified as a realization of utility computing. Although the idea of cloud computing has been around for quite some time, it is an evolving field of computer science. Since the evolution of cloud computing: Load balancing, energy management, VM migration, server consolidation, cost modelling and security issues are the popular research topic in this field. Deploying real cloud for testing or for commercial use is very costly. Cloud computing model have complex provisioning, composition, configuration, and deployment requirements. Evaluating the performance of Cloud provisioning policies, application workload models, and resources performance models in a repeatable and controllable manner under varying system and user configurations and requirements is difficult to accomplish. To overcome this challenge, cloud simulator is needed. In this paper basic of cloud simulator is discussed, and major focus is on cloudsim- a simulator for management of vm. The CloudSim toolkit supports both system and behaviour modeling of Cloud system components such as data centers, virtual machines (VMs) and resource provisioning policies. It implements generic application provisioning techniques that can be extended with ease and limited effort. Currently, it supports modeling and simulation of Cloud computing environments consisting of both single and inter-networked clouds (federation of clouds). In this paper how cloudsim work, its architectural design, highlighting important features and give brief overview of its functionalities is presented.

Keywords: Cloud computing; modelling and simulation; Virtual machines; VM management, millions of instruction per second (MIPS)

I. INTRODUCTION

Cloud computing supplies infrastructure, platform, and software as service based on pay- as-you-go-model to customers. These services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) in industries. The importance of these services was highlighted in a recent report from the University of Berkeley as: 'Cloud computing, the long-held dream of computing as a utility has the potential to transform a large part of the IT industry, making software even more attractive as a service' [1].

Clouds [11] aim to power the next-generation data centers as the enabling platform for dynamic and flexible application provisioning. This is facilitated by exposing data center's capabilities as a network of virtual services (e.g. hardware, database, user-interface, and application logic) so that users are able to access and deploy applications from anywhere in the Internet driven by the demand and Quality of Service (QoS) requirements [3]. Similarly, IT companies with inventive ideas for new application services are no longer required to make large capital outlays in the hardware and software infrastructures. By using clouds as the application hosting platform, IT companies are freed from the trivial task of setting up basic hardware and software infrastructures. Thus, they can focus more on innovation and creation of business values for their application services [1].

Clouds exhibit varying demands, supply patterns, system sizes, and resources (hardware, software, and network); users have heterogeneous, dynamic, and competing QoS requirements; and applications have varying performance, workload, and dynamic application scaling requirements. The use of real infrastructures, such as Amazon EC2 and Microsoft Azure, for benchmarking the application performance (throughput, cost benefits) under different environments (availability, workload patterns) is often constrained by the rigidity of the infrastructure. Hence, this makes the reproduction of results that can be relied upon, an extremely difficult undertaking. Further, it is tedious and time-consuming to re-configure benchmarking parameters across a massive-scale Cloud computing infrastructure over multiple test runs. Such limitations are caused by the conditions prevailing in the Cloud-based environments that are not in the control of developers of application services. Thus, it is not possible to perform benchmarking experiments in repeatable, dependable, and scalable environments using real-world Cloud environments [10]. A more viable alternative is the use of simulation tools. These tools open up the possibility of evaluating the hypothesis (application benchmarking study) in a controlled environment where one can easily reproduce results. Simulation-based approaches offer significant benefits to IT companies (or anyone who wants

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, *January 2015*

to offer his application services through clouds) by allowing them to: (i) test their services in repeatable and controllable environment; (ii) tune the system bottlenecks before deploying on real clouds; and (iii) experiment with different workload mix and resource performance scenarios on simulated infrastructures for developing and testing adaptive application provisioning techniques[10]The remainder of this paper is organized as follows: first, a general description about Cloud computing, existing models, and their layered design is presented. This section ends with a brief overview of existing state-of-the-art in distributed (grids, clouds) system simulation and modeling. Following that, comprehensive details related to the architecture of the CloudSim framework are presented. Section 4 presents the available vm management techniques in cloudsim. Section 5 presents implementation details of vm management techniques in cloudsim. Finally, the paper ends with brief conclusive remarks.

II. Related Work

In the past decade, Grids [7] have evolved as the infrastructure for delivering high-performance services for compute- and data-intensive scientific applications. To support research, development, and testing of new Grid components, policies, and middleware, several Grid simulators, such as GridSim [4], SimGrid [8], OptorSim [5], and GangSim [6], have been proposed. SimGrid is a generic framework for simulation of distributed applications on Grid platforms. Similarly, GangSim is a Grid simulation toolkit that provides support for modeling of Grid-based virtual organizations and resources. On the other hand, GridSim is an event-driven simulation toolkit for heterogeneous Grid resources. It supports comprehensive modeling of grid entities, users, machines, and network, including network traffic.
Although the aforementioned toolkits are capable of modeling and simulating the Grid application management behaviors (execution, provisioning, discovery, and monitoring), none of them are able to clearly isolate the multi-layer service abstractions (SaaS, PaaS, and IaaS) differentiation required by Cloud computing environments. As Cloud computing main aim to deliver services on subscription-basis in a pay-as-you- go model to SaaS providers. Therefore, Cloud environment modeling and simulation toolkits must provide support for economic entities, such as Cloud brokers for enabling real-time trading of services between customers and providers. Among the currently available simulators discussed in this paper no simulators offer support for simulation of virtualized infrastructures, neither have they provided tools for modeling data-center type of environments that can consist of hundred-of-thousands of computing servers.
As Cloud computing R&D is still in the infancy stage [1], a number of key issues need detailed exploration. Topics of interest consist of cost-effective and also energy-efficient approaches for provisioning of virtualized resources to end-user's requests, inter-cloud negotiations, and federation of clouds. To support and accelerate the research related to Cloud computing systems, applications and services, it is essential that the required software tools are designed and developed to support researchers and industrial developers.

III. CloudSim Architecture

Figure 1 shows the multi-layered design of the CloudSim software framework and its architectural components. The CloudSim simulation layer provides support for modeling and simulation of virtualized Cloud-based data center environments including dedicated management interfaces for VMs, memory, storage, and bandwidth.

The fundamental issues, such as provisioning of hosts to VMs, managing application execution, and monitoring dynamic system state, are handled by this layer. A Cloud provider, who wants to study the efficiency of different policies in allocating its hosts to VMs (VM provisioning), would need to implement his strategies at this layer. Such implementation can be done by programmatically extending the core VM provisioning functionality. There is a clear distinction at this layer related to provisioning of hosts to VMs.

A Cloud host can be concurrently allocated to a set of VMs that execute applications based on SaaS provider's defined QoS levels. This layer also exposes the functionalities that a Cloud application developer can extend to perform complex workload profiling and application performance study. The top-most layer in the CloudSim stack is the User Code that exposes basic entities for hosts (number of machines, their specification, and so on), applications (number of tasks and their requirements), VMs, number of users and their application types, and broker scheduling policies.

By extending the basic entities given at this layer, a Cloud application developer can perform the following activities: (i) generate a mix of workload request distributions, application configurations; (ii) model Cloud availability scenarios and perform robust tests based on the custom configurations; and (iii) implement custom application provisioning techniques for clouds and their federation.

Overview of CloudSim functionalities: [12]

- Support for modeling and simulation of wide range cloud computing data centers which contain more number of host and vm.
- Support for modeling and simulation of virtualized server hosts, with customizable policies for management of vm on host and resources utilization of host is improved.
- Support for modeling and simulation of power-aware computational resources
- Support for modeling and simulation of inter connected clouds.
- Support for dynamic insertion of simulation components, stop and resume of simulation

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, *January 2015*

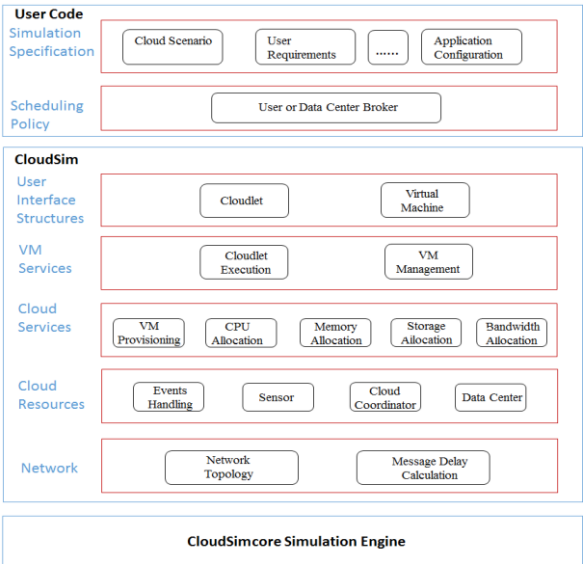

Fig. 1. Cloudsim Architecture [10]

IV. VM MANAGEMENT TECHNIQUES IN CLOUDSIM:

Cloudsim is most popular tool for cloud computing environment specifically for the purpose of server consolidation problem. Most of researchers are using cloudsim for evaluation of vm management algorithms and energy-efficient management of data centers. All VM management algorithms in cloudsim basically relay on analysis of historical data of the resource usage by Vms with respect to time and millions of instruction per second requested and allocated to Vms. All historical data are stored in form of file and file contains %cpu utilized by vm in one day. There are 288 data each are in interval of 5 minutes. Cloudsim is an event driven simulator. It is written in most popular object oriented language java. Because of OOP feature cloudsim modules can be easily extendable with the user's requirement. One of the drawback of cloudsim is lack of GUI. The problem of server consolidation can be divided into four parts [2].

A. Determining when host is overloaded- which required migration of one or more vm from one host to other host - (Host overload Detection).
B. Determining when host is under-loaded- which required all vm to be migrated from this host and switch the host to sleep mode - (Host under-load Detection).
C. Selection of VMs that should be migrated from an overload host - (VM Selection)
D. Finding new placement of VMs selected for migration from the overloaded and under-loaded hosts - (VM Placement).

A. *Host Overload Detection*

There are two types of approach used to find whether the host is overloaded or not. 1) Adaptive utilization threshold base and 2) non-threshold base algorithm.

First of all, some variables should be define before illustrating the implemented algorithms in the simulator (CloudSim). Assuming there are P hosts in data centers, denote the ith host as Li and the number of VMs on Li as Qi; denote the jth VM on host Li as Sij, the utilization of CPU for Sij in time frame t as uij,t (utilization must be between 0 and 1) and the maximum utilization of CPU for Sij as uij; furthermore, denote the number of time frames for each VM as p.

Median absolute deviations: Mad is adaptive utilization threshold base algorithms. Here we find upper threshold of data using MAD. Mad is of statistical dispersion of data. Here set of data is available in form of %cpu utilize by vm. For example suppose set of data is {X1,X2,……Xn}. Median is middle value of data if data is odd and if data is even median is average of two middle value. To find MAD of data first data is sorted in increasing order. For example X1 ≤ X2≤ ….≤Xn. and than find median of data as shown in (1)

$$Med = median(\{X1, X2, \ldots Xn\}) \ldots . 1)$$

After that take a absolute difference between each element of available data and median of data. Denote it as {Y1,Y2,…Yn} and sort it in increasing order.{Y1 ≤Y2 ≤… ≤Yn}. Finally MAD of data is calculated as shown in (2)

$$MAD = median(\{Y1, Y2, \ldots Yn\}) \ldots\ldots . 2)$$

Now , find upper threshold of data as shown in (3)

$$Tu = 1 - s\ MAD \ldots .. \ldots\ldots\ldots ..(3)$$

Where s is safety parameter (€R+), it is adjusted values for this method on the basis of experimental approach. For MAD its value is 2.5, If utilization of host exceeds upper threshold than host is overloaded otherwise not.

Interquartile range (IQR): It is also an adaptive utilization threshold based method. IQR is measure of statistical dispersion of data. It is also called as middle fifty and midspered. IQR equal to the difference between the upper and lower quartiles*:* IQR=Q3 _ Q1, where Q1 and Q3 are the 25thpercentile and 75th percentile respectively of a sorted set. Then the upper threshold for host Li is found by equation (4).

$$Tu = 1 - s\ IQR \ldots . (4)$$

Where s is safety parameter (€R+), as shown in method 1), here value for s is 1.5.
We can determine host is overloaded or not same as previous method.

Local Regression (LR): It is non threshold base algorithm. It means there is no upper threshold set. But based on past data predicted utilization of host in next time frame is determine. In linear algebra regression means to find a relation between two variables. These two variables are time and % cpu utilized by vm for each time interval. From large available data set we can draw a trend line between those data sets. Suppose trend line of the data set is y= a*x+ b, where a, b €R. How local regression algorithm work which is shown in below figure 2.

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, *January 2015*

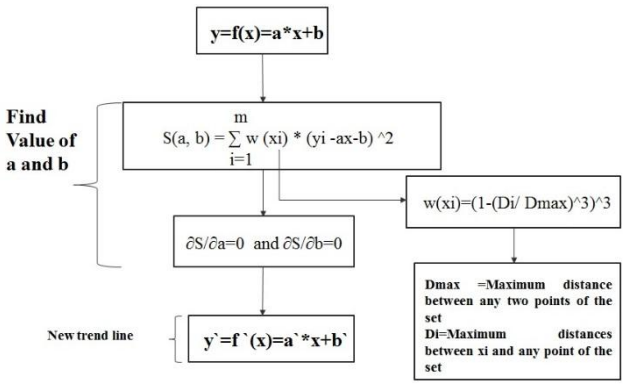

Fig. 2Working of LR

This trend line used to estimate the next observation f(xn+1). The predicted utilization of host in next time frame xn+1 can be calculated

$$Pu = s * f(xn+1) \ldots\ldots (5)$$

Where s is safety parameter (€R+), as shown in method 1), here value for s is 1.2. If Pu >1, Host is over-utilized otherwise not.

Robust local regression (LRR): Local regression is vulnerable to outliers that can be caused by heavy-tailed distributions or other distributions. To make it robust, Cleveland has presented Robust Local Regression. LRR is a little different from LR and an extended version of LR. How robust local regression algorithm work which is shown in below figure 3. This trend line used to estimate the next observation f(xn+1). The predicted utilization of host in next time frame xn+1 can be calculated same as local regression algorithm.

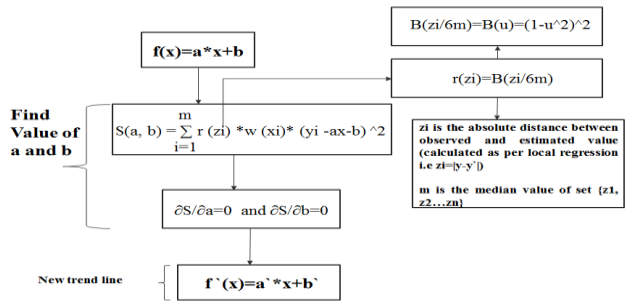

Fig. 2Working of LRR

Some important attributes from the above host overload detection algorithm is shown in given table I.

TABLE I
APPROACH FOR HOST OVERLOAD DETECTION

Algorithm	Approach for host overload detection	Safety Parameter
MAD(Median Absolute Deviation)	Adaptive utilization threshold	2.5
IQR(Interquartile Range)	Adaptive utilization threshold	1.5
LR(Local Regression)	Non- threshold	1.2
LRR(Robust Local Regression)	Non-threshold	1.2

B. *VM Selection*

If a host is overloaded, then some VMs should be migrated from it and make it not generate SLA violation. There are four policies [2] to migrate VMs from overloading hosts.

1) Minimum Utilization Policy: The Minimum Utilization Policy (MU) is a simple method to select VMs from overloading hosts. Among those Qi VMs on host Li, select the minimum utilization VM to migrate. If it is still overloaded, then repeat the step until the host considered being not overloaded.

2) The Random Choice Policy: The Random Choice Policy (RC) is another simple method to select VMs from overloading hosts. Among those Qi VMs on host Li, randomly select a VM to migrate. If it is still overloaded, then repeat the step until the host considered being not overloaded.

3) The Minimum Migration Time Policy: The Minimum Migration Time Policy (MMT) means to migrate a VM, which has the minimum migration time among Qi VMs on host Li, and repeat the step until the host considered being not overloaded. The migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host Li.

4) The Maximum Correlation Policy: The Maximum Correlation Policy (MC) means to migrate a VM Sij on host Li, whose utilization uij,t has the maximum correlation coefficient with the sum of the other VMs' on host Li, and repeat the step until the host considered being not overloaded. The correlation coefficient between the utilization of Sij and the sum of other VMs' on hosts Li can be expressed in equation (6).

$$\rho j = \frac{E[(\text{uij},t-1/p\sum_{t=1}^{p} uij,t)(\text{Uij},t-1/p\sum_{t=1}^{p} Uij,t)]}{\sqrt{S(\text{uij},t)} * \sqrt{S(\text{Uij},t)}} \quad \ldots\ldots\ldots (6)$$

Where Uij,t and S(uij,t) can be extended to (7) and (8).

$$\text{Uij},\ t = \sum_{q=1}^{j-1} uiq,t + \sum_{j+1}^{Qi} uiq,t \quad \ldots\ldots\ldots (7)$$

$$S(\text{uij},t) = E[(uij,t - 1/p \sum_{t=1}^{p} uij,t)\text{^}2] \quad \ldots\ldots\ldots (8)$$

S(Uij,t) has the same appearance with S(uij,t). For there are Qi VMs on host Li, so we can get Qi correlation coefficients. Then the maximum correlation coefficient can be calculated in the equation (9).

$$\rho\max = \max(\{\rho 1\text{^}2, \rho\, j\text{^}2, \ldots\ldots \rho\, Qi\text{^}2\}) \quad \ldots\ldots\ldots\ldots (9)$$

DOI 10.17148/IJARCCE.2015.4126

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, *January 2015*

C. *VM Placement*

Modified Best Fit Algorithm: [2]

1 **Input:** hostList, vmList **Output:** allocation of VMs
2 vmList.sortDecreasingUtilization()
3 foreach vm in vmList do
4 minPower MAX
5 allocatedHost NULL
6 foreach host in hostList do
7 if host has enough resources for vm then
8 power estimatePower(host, vm)
9 if power < minPower then
10 allocatedHost host
11 minPower power
12 if allocatedHost = NULL then
13 allocation.add(vm, allocatedHost)
14 return allocation

D. Host Under-load Detection [2]

For determining under-loaded hosts simple approach is used. First, all the overloaded hosts are found using the selected overload detection algorithm, and the VMs selected for migration are allocated to the destination hosts using vm placement algorithm. Then, the system finds the host with the minimum utilization compared to the other hosts, and tries to place the VMs from this host on other hosts keeping them not overloaded. If this can be accomplished, the VMs are set for migration to the determined target hosts, and the source host is switched to the sleep mode once all the migrations have been completed. If all the VMs from the source host cannot be placed on other hosts, the host is kept active. This process is reiterated for all hosts that have not been considered as being overloaded.

V. IMPLEMENTATION OF VM MANAGEMENT TECHNIQUES IN CLOUDSIM

As discussed earlier for managing VM in cloudsim there are four major part: host overload detection, host under-load detection, vm selection and vm placement. For implementation of VM management techniques in CloudSim some class should be modified which is shown in below table II. Other important class for Vm management technique is shown in table III.

In cloudsim some of functionalities are lacking, as cloudsim is going to be extended by researchers [12]. Currently cloudsim allows to implement infrastructure as a service (IAAS).

There is no user define SLA provided in cloudsim and calculation of SLA violation is based on available and requested MIPS.

There are no dynamic characteristics for host and vm defined. All vm management techniques is based on utilization of cpu. VM management and data center selection policy is combinely given in cloudsim.

TABLE III
IMPORTANT CLASSES FOR VM MANAGEMENT TECHNIQUES

VM Management Technique	Important Classes
Host Overload Detection	org.cloudbus.cloudsim.examples.power.RunnerAbstract.java, org.cloudbus.cloudsim.examples.power.planetlab.YourAllocation.java org.cloudbus.cloudsim.power.PowerVmAllocationPolicyMigrationYourAllocation.java
Host Under-load Detection	org.cloudbus.cloudsim.power.PowerVmAllocationPolicyMigrationAbstract.java
VM-Selection	org.cloudbus.cloudsim.examples.power.RunnerAbstract.java, org.cloudbus.cloudsim.power.PowerVmSelectionPolicyYourSelection.java
VM-Placement	org.cloudbus.cloudsim.power.PowerVmAllocationPolicyMigrationAbstract.java

TABLE III
OTHER IMPORTANT CLASSES

Class Name	Function
org.cloudbus.cloudsim.examples.power.Helper.java	No of host shutdown, SLA calculation
org.cloudbus.cloudsim.examples.power.Constants.java	Host and VM Characteristics
org.cloudbus.cloudsim.util.MathUtil.java	All mathematical calculation related to vm management technique
org.cloudbus.cloudsim.examples.power.planetlab.PlanetLabConstants.java	Total no of host

VI. CONCLUSION

Cloud computing is new era of computing utilities which provide utilities as a service like pay as u go model. Because of cloud computing IT services are growing faster and its complexity is reduces. For efficiently managing Cloud infrastructures Cloud technologies focus on novel methods and policies.

But to test these newly developed methods and policies, researchers need tools that allow them to evaluate the hypothesis prior to a real deployment in an environment, where one can repeat tests. Simulation-based approaches in estimating Cloud computing systems and application behaviours offer substantial advantages, as they allow Cloud developers: (i) to test the performance of different service delivery policies in a repeatable and controllable environment free of cost; and (ii) to change the performance bottlenecks before real-world deployment on commercial Clouds.

In this paper we have presented cloudsim-toolkit for vm management techniques. Cloudsim basically used for managing vm at different condition. These condition are host overload, host under load, vm selection and vm placement. Choosing appropriate vm at each condition which interns reduces energy consumption, no of vm migration.

DOI 10.17148/IJARCCE.2015.4126

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 1, *January 2015*

ACKNOWLEDGMENT

I would like to thank Dr. Rajkumar Buyya Professor of Computer Science and Software Engineering and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Rodrigo N. Calheiros research fellow at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia, Anton Beloglazov is a staff researcher at IBM Research.
I would like thank my guide, Prof. Ritesh Patel Associate Professor, U & P U. Patel Department of Computer Engineering, C.S.P.I.T., CHARUSAT, Changa, Gujarat, Indiafor all his diligence, guidance, encouragement.

REFERENCES

[1] Armbrust M, Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. *Communications of the ACM* 2010; **53**(4):50–58.
[2] Anton Beloglazov*, Rajkumar Buyya : " Optimal Online Deterministic Algorithms And Adaptive Heuristics For Energy And Performance Efficient Dynamic Consolidation Of Virtual Machines In Cloud Data Centers ": Online In Wiley Inter-science.
[3] Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems 2009; 25(6):599–616.
[4] Buyya R, Murshed M. GridSim: A toolkit for the modeling and simulation of distributed resource management andscheduling for grid computing. *Concurrency and Computation Practice and Experience* 2002; **14**(13–15):1175–1220.
[5] Bell W, Cameron D, Capozza L, Millar P, Stockinger K, Zini F. Simulation of dynamic Grid replication strategies in OptorSim. *Proceedings of the Third International Workshop on Grid Computing (GRID)*, Baltimore, U.S.A.
[6] Dumitrescu CL, Foster I. GangSim: A simulator for grid scheduling studies. *Proceedings of the IEEE InternationalSymposium on Cluster Computing and the Grid*, Cardiff, U.K., 2005; 1151–1158.
[7] Foster I, Kesselman C (eds.). *The Grid*: *Blueprint for a New Computing Infrastructure*. Morgan Kaufmann:
[8] Legrand A, Marchal L, Casanova H. Scheduling distributed applications: The SimGrid simulation framework.*Proceedings of the Third IEEE/ACM International Symposium on Cluster Computing and the Grid*, Tokyo, Japan,2003; 138–145.
[9] Quiroz A, Kim H, Parashar M, Gnanasambandam N, Sharma N. Towards autonomic workload provisioning for enterprise grids and clouds. Proceedings of the 10th IEEE/ACM International Conference on Grid Computing (Grid 2009), Banf, AB, Canada, 13–15 October 2009. IEEE Computer Society: Silver Spring, MD, 2009; 50–57
[10] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose, Rajkumar Buyya :"CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Published online 24 August 2010 in Wiley Online Library
[11] Weiss A. Computing in the clouds. NetWorker 2007; 11(4):16–25
[12] http://www.cloudbus.org/cloudsim/

BIOGRAPHIES

Miss Seema Vahora obtained her Bachelor's degree in computer engineering from DDU, nadiad, Gujarat in 2008 andpursuing the Masters Degree in computer Engineering from CHARUSAT,changa, Gujarat. Her research interests include Cloud Computing,and networking.

Mr. Ritesh Patel obtained his Bachelor's degree in computer engineering from Ganpat university, mehsana, Gujarat in 2002and Masters Degree in computer Engineering from DDU, nadiad , Gujarat. In 2004 and pursuing PHD in area of cloud computing from CHARUSAT,changa. Currently he is working as Associate Professor in Computer Engineering Department at Charusat Charotar university of Science & Technology, changa, Gujarat. His research interests include next generation network, cloud computing, parallel computing and advanced computer architecture.