Task Performance Analysis in Virtual Cloud Environment

Pooja Gupta[1] Anurag Kumar [2] Dr. Gopal Krishan [3]

[1]*Assistant Professor, Uttaranchal University, Dehradun, State: Uttarakhand, India*

[2]*Department of Computer Science and Engineering, GRD Institute of Management and Technology, Dehraun.. State: Uttarakhand, India.*

[3]*Department of Computer Service Centre, IIT Delhi, India*

Abstract: **Cloud computing based applications are beneficial for businesses of all sizes and industries as they don't have to invest a huge amount on initial setup. This way, businesses can opt for Cloud services and can implement innovative ideas. But evaluating the performance of provisioning (e.g. CPU scheduling and resource allocation) policies in a real Cloud computing environment for different application techniques is challenging because clouds show dynamic demands, workloads, supply patterns, VM sizes, and resources (hardware, software, and network). User's requests and services requirements are heterogeneous and dynamic. Applications models have unpredictable performance, workloads, and dynamic scaling requirements. So a demand for a Simulation toolkit for Cloud is there. Cloudsim is self-contained simulation framework that provides simulation and modeling of Cloud-based application in lesser time with lesser efforts. In this paper we tried to simulate the task performance of a cloudlet using one data center, one VM. We also developed a Graphical User Interface to dynamically change the simulation parameters and show simulation results.**

Keywords: **CloudSim, VM, DataCenter, Provisioning policy, Virtualization**

I. INTRODUCTION

1.1 Cloud Computing

Cloud computing is nothing but using services online, internet integration, virtualization, online business platforms and using pay-as-use model.

According to M. Klems [1] Cloud computing provides scalability on demand in a matter of few seconds. Dynamic provisioning of resources can be done while running the application without any downtime. This way over-utilization and under-utilization of resources can be avoided.

R. Buyya [2] says Cloud computing is a 'parallel and distributed computing' where physical machines consist of a collection of inter-linked and virtualized machines These machine are dynamically provisioned (that means resource like memory, processing units etc. can be added/removed at run time), and physical machine are considered as consolidated computing resources fulfilling QoS and SLAs between the service provider and the users. Virtualization is the backbone of cloud computing. It helps in providing on the fly and on demand configuration of resources to run different cloud applications [3]. Using virtualization, Cloud computing provides software, hardware and platform based services on subscription based model and hence reducing wastage of resources. In virtualization, computing resources are in the form of virtual machine [4]. A software layer, known as VM monitor/hypervisor manages the access to physical resources and maintains isolation between all Virtual Machines at all times. VMM controls resources allocated to VM [4].

Many VMs can run simultaneously on the same PM. A virtual machine can experience a dynamic workload and hence, resources can be over-utilized (hotspot) or under-utilized (cold spot) [5]. Due to variation in resource demands, dynamic resource provisioning techniques are required [6]. For example, the Data Center in any part of the globe may be over-utilized in daytime (from 9 AM to 9PM) but same Data Center may be under-utilized during night. Hence it is required to develop some dynamic resource provisioning schemes, for applications to achieve QoS and avoid SLA violation [7]. Evaluating the performance of provisioning policies like scheduling and allocation of resources in an actual Cloud under service models and different conditions is not feasible because of the following reasons [7]:

(i) Applications in cloud experience variation in demand & supply patterns,
(ii) Consumers have different and challenging QoS requirements.

Users can not go for repeatable experiments with varying demands of resources in real cloud infrastructures. They have to pay in real currency. Buyya et al proposed, a simulation tool, called CloudSim as a solution to the mentioned problem. According to them, CloudSim is a general tool that can be used for modeling and simulation of cloud environment [6]. CloudSim supports modeling analysis of different components of cloud computing. It facilitates cloud service providers to measure the performance of the applications executing on the behalf of the consumers and can come up with a most suitable model. In this paper, we have performed a simulation for a case study: Simulation Results for Virtual Cloud Environment with one Data Center, one Virtual Machine and one cloudlet using Time-shared provisioning policy. We also created a user-interface to dynamically change the properties of Host, VM and cloudlet. We have used this simulation tool to show the variation in cloudlet execution Finished time with different MIPS and with different cloud lengths.

Paper is organized as follows: Section 1 gives the brief introduction of cloud computing, virtualization and problem facing in implementing different experiments in real cloud environment. Section 2 presents the existing research on cloud simulation, introduction to cloudSim, its features and brief review of the key components of CloudSim. Section 3 explains modeling of Cloud and virtual machine. It also explains resource provisioning policies like time-shared and space-shared. Section 4 presents a set of experiments conducted successfully using CloudSim. Finally, section 5 gives a brief conclusion and future work.

II. LITERATURE SURVEY

From the last 10 years Grids computing is used to provide infrastructure for high-performance services in the area of data-oriented scientific applications. A lot of research has been done in the area of grid computing. Many Grid simulators e.g. GridSim [8], simGrid [9], optorSim [10] and GangSim [11] are proposed by different researchers. But none of them is capable of working in multi-layer Cloud environment [6]. In 2010, two IT giants Yahoo and HP collaborated together to propose the development of a global Cloud computing testbed, called Open Cirrus, this Open Cirrus is going to support a federation of data centers located in 10 organizations. But developing this type of experimental setup is very expensive and it is not simple to conduct tests in repeatable manner as resource availability is dynamic [6]. In 2009 Bryan M. A. El-Refaey and M. A. Rizkaa propose their work on Virtual Systems Workload Characterization [12] which introduced an indication of the major requirements and uniqueness of virtual machines performance and workload characterization. Buyya et al [6] propose CloudSim: It is a simulation tool that supports modelling and simulation of Cloud computing platform. According to the authors the CloudSim toolkit supports modelling and creation of one or more virtual machines (VMs) on a simulated node of a Data Center and their mapping to suitable PMs according to the requirement of VMs like memory, storage, processing power [7].

Another simulator, iCanCloud [13] is also used for conducting experiments in cloud computing. It provides a trade-off model between cost and performance for every cloud application that is running on a specific machine. This model is useful in calculating cost per user [14].

Some authors [15] argue that setting up experiments in real cloud is too costly, and simulation tools are good alternatives. They compare most powerful tools for cloud simulation like CloudSim, CloudAnalyst, CloudReports, CloudExp, GreenCloud, and iCanCloud and conclude that iCanCloud is a superior platform as it provides GUI support to the users.

[16] Defines CloudSim and reviewed all it's available variants such as GreenCloud, CloudAnalyst (an extension of CloudSim), EMUSIM, Network CloudSim, and MDCSim. They compare all CloudSim Variants with different parameters like platform, networking and language. Finally, they conclude that the CloudSim is probably the best among the all simulators they reviewed.

Cloud computing is not fully explored area and a lot of research is going in cloud computing, hence new ideas, techniques and capabilities are being introduced on regular basis. This extensible feature of cloud computing, makes designing of customized cloud model, a very challenging task. Definitely a simulation tool is required to evaluate the cloud configuration for actual deployment. CloudSim is the first preference of researcher to experiment the application for quality of service identification [16]. CloudSim, allows the researchers and application developers to concentrate on system design issues that they want to investigate, without bothering about low level details related to Cloud-infrastructures and services.

2.1 CloudSim features: [6] defines the following novel features of CloudSim.

(i) Modeling and simulation of Cloud, Data Centers, Virtual Machines, Hosts on a single physical computing system.
(ii) A self-contained modeling framework for modeling and simulation of cloud environment, service brokers, resource provisioning and allocation policies.
(iii) It provides a virtualization engine which helps in creation and management of many independent and co-hosted virtualized service on same data center.
(iv) Provides aids to network simulation in simulated system nodes.
(v) CloudSim provides flexibility to switch between resource allocations algorithms (space-shared and time-shared).
(vi) Both public and private domain of federated cloud that interconnects resources can be simulated. It is an important point to study in concern with Cloud-Burst and automatic application scaling. These features of CloudSim accelerate the development of new provisioning algorithms for the Cloud [17].

The idea behind this tool based simulation is to run the cloudlet on a cloud computing environment with high quality of service that simulation time and cloudlet finish time should remain the minimum.

2.2 Review of topics related to CloudSim and virtualization:

2.2.1 Virtualization: Virtualization is important for provisioning requirements of a cloud-computing solution. Virtualization facilitates a logical view of resources used to instantiate virtual machines (VMs). It is the role of VM monitor/VMM/hypervisor to control and multiplex the access to physical resources and maintain isolation between all Virtual Machines at all times. It is hypervisor who provides access to physical resources and supervises the access to resources such as memory allocated to VM, CPU allocated to a VM, and these access could be over-utilized or under-utilized [3].

2.2.2 Hypervisor: Hypervisor (e.g. Xen, KVM) is a software that manages the VM-PM mapping. It also manipulates the access to resources. It is a software layer called virtual machine monitor (VMM) that resides above the hardware and maps the virtual machine to physical resources. Each VM runs on either unmodified (full virtualization) or little modified (para-virtualization) type of operating systems used in virtualization. Full virtualization runs on bare metal whereas in para-virtualization, OS is modified to support virtualization. Hypervisors controls different functions for the hosting VMs, these operations are creation, deletion, restart, suspend and migration [4].

2.2.3 Cloud Information Service (CIS): CIS usually scan the user request and direct them to most appropriate cloud service providers. It does the user request and cloud provider match making at database level. Discovery of available resources for allocation is the main responsibility of CIS. Datacenter Broker is also responsible for the same work and it is the main part of scheduling in simulated environment [18, 19, 20].

2.2.4 Regions: It represents the geographical areas in which CSPs provides services to their customers. There are six regions that relates to six continents in the world [19].

2.2.5 Datacenter: Data center consists of hosts or physical nodes. It models the infrastructure services provided by various cloud service providers. It consists of a set of computing hosts (servers) that are either heterogeneous or homogeneous; it depends on their hardware configurations [5, 12].

2.2.6 Broker: Datacenter broker is a module of cloud computing environment that manages the datacenter. It allocates the cloudlets to suitable VMs. It decides which data center should be chosen to provide services to the user requests [5, 12].

2.2.7 Virtual Machine: Each VM running on a PM has some resources attached (e.g. CPU, memory I/O devices). While instantiating VM, each VM is allocated these resources. To manage the use of these resources across the VM is the responsibility of the hypervisors. Initial provisioning levels of a VM depend upon resource-usage profiles of applications or it depends on number of resources required to meet the load requirements. Due to dynamic workload of VMs, two undesirable conditions can occur:

- Hot spot: it means resources crunch. Not enough resources are provisioned to meet demand.
- Cold spot: it means resources are not utilized efficiently.

2.2.8 Cloudlet: It is a set of user request or tasks. A cloudlet is sent from the user for processing to the DC [21]. Its main properties are cloudlet ID (unique Id), cloudlet length (complexity of task) and arrival time etc. The cloudlet length of cloudlets should not be less than one; it should be either greater or equal to one [22]. It has some information attached to it like application ID, name of the user base that is the source to which the responses have to be sent back, and the size of the request execution commands, and input and output files. It models the cloud-based application services. The complexity of an application is defined in terms of its computational requirements. Complexity of an application is directly proportional to its computational requirements.

2.2.9 VMM allocation policy: It models VM allocation provisioning policies. Which VM should be hosted by which host that depends on the processing requirements of the VMs and according to that only a suitable host is selected.

2.2.10 VM scheduler: It represents the models for scheduling policies like time or space shared; these policies are used to allocate processing elements to VMs.

III. IMPLEMENTIGN WITH CLOUDSIM

Following are the steps used in simulation of cloudsim:

1- The first step is to fix the number of users for an on-going simulation, i.e. broker count.
2- Second step is the initializing of simulation; it is done by instantiating the common parameters (current time, number of users).
3- In third step CIS instance is created.
4- After CIS instance is created, creation and registration of data center instance is done and Host characteristics: PE, RAM, and BW are initialized.
5- Creating physical machines (hosts) with their characteristics, e.g.PE, RAM, BW etc.
6- Creating data center broker instance. It deals with the communication between datacenter and submission of cloudlets and VM. This step is important in cloud simulation.
7- Next step is to create VMs and initialize their properties. VMs have same characteristics as Datacenters or Hosts.
8- Once virtual machines are created; they are submitted at data center broker. Till step 8 only basic infrastructures have been created for simulation.
9- Creating cloudlets (tasks) and specifying their characteristics like MIPS, BW. There may be single or more cloudlets.
10- Cloudlets are submitted to data center broker. Now broker has both Virtual Machines and Cloudlets.

11-Once every task is submitted and every instance is created, a call to start the simulation is sent, when there is an event to be executed.

12-When there is no event to be executed, a call to stop the simulation is sent.

3.1 Modeling the Cloud using CloudSim:

For simulating a cloud, we first need to model infrastructure and for this reason Data Center class of CloudSim can be extended. A DC can manage multiple hosts which in turn can handle many VMs allocated to it. A single host can be allocated to many VMs. There are many resource allocation policies. [23] Have categorized them as static resource allocation techniques and dynamic resource allocation techniques. VM allocation policies explain all VM related operations such as VM creation, VM destruction and VM migration [6]. VM migration help to successfully achieve various resource management objectives such as load balancing, power management, fault tolerance, system maintenance [23] and to reduce energy consumption. Some VM- PM mapping techniques are required to decide which virtual machine (VM) will be mapped onto which physical machine (PM) [16]. By allocating VMs to PM we are actually creating VM instances on PMs. Selection of most appropriate PM to host VMs is called virtual machine placement process. It is a crucial process of resource utilization and power efficiency of cloud computing environment [24]. Destination PM Selection to reduce waste of resources is an area of research itself.

CloudSim [6] helps in developing the customized user application which can be deployed in a VM instance. The core cloudlet objects are extensible and can be extended by users in order to implement their own application services. When these applications are defined and modeled they can be assigned to any VM by using provisioning policy. Provisioning of specific task based VM to some host on DC is the responsibility of VM allocation Controller component known as VmAllocationPolicy. This component uses First Come First Served policy to allocate VM to host.

There is another CloudSim policy (Host allocation policy) that is responsible for the allocation of CPUs to Vms using Host. There are two provisioning policies: Time-shared and Space-shared. A particular CPU is assigned to a specific VMs, is space-shared policy whereas in time-shared policy the computing power of the CPU is dynamically distributed among various Vms [6].

3.2 Modeling the VM allocation:

In VM allocation, processing power of a processor can be shared by two VMs. CloudSim supports both time-shared and space-shared policies at both levels [6]. For example, let's assume that, there is one host with two CPUs and this host receives request from two VMs. Each VM require two CPU to execute four tasks. Like tasks t1, t2, t3, and t4 to be hosted in VM1, whereas t5, t6, t7, and t8 to be hosted in VM2. Following cases explains the effects of different provisioning techniques on cloudlet execution:

Case a:

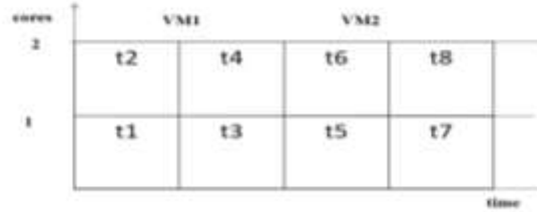

Figure 3.1: Case(a)

Policy Used: Same policy at both levels, Space-Shared at both VM and task unit levels

Drawback: Only one VM will run at a time. VM2 will start after VM1 finishes.

Benefit: Two tasks can run in parallel because each task needs one CPU.

So estimated finish time (eft) of a task using space-shared policy is:

$$eft(p) = est + \frac{rl}{capacity \times cores(p)}$$

Where est (p)=cloudlet estimated start time

rl= total no of instructions that cloudlet need to execute.

cores (p)= number of Pes.

Case b:

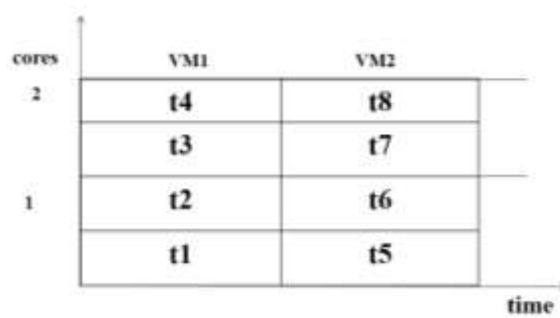

Figure 3.2: Case (b)

Policy Used: Different policy at different levels, Space –shared policy for VM allocation and time-shared policy within the VMs.

All tasks assigned to a VM will dynamically context switch until they finish. So estimated finish time of a task using time-shared policy is:

$$eft(p) = ct + \frac{rl}{capacity \times cores(p)}$$

Where ct= current simulation time

Case c:

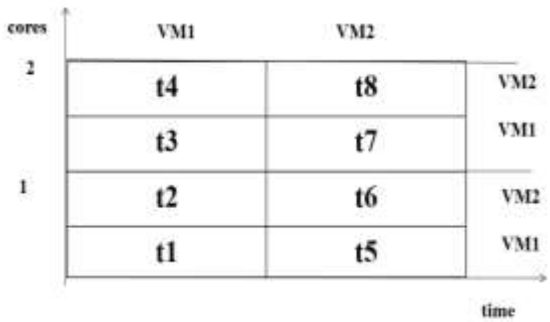

Figure 3.3: Case (c)

Policy used: This case is reversed of case (b), here, different policies used at different levels, Time–shared policy for VM allocation whereas Space-shared policy within the VMs. In this, VM receives a time-slice on each processing units. That again distributes the time slices among the task units on a space-shared policy.

Drawback: only one task will be using CPU at any given time.

Case d:

Figure 3.4: Case(d)

Policy used: Time-shared allocation is applied to both VMs and tasks.

Benefit: No waiting time in queues associated with task units.

IV. METHODOLOGY AND IMPLEMENTATION

To evaluate the performance of cloud, results were simulated in Window 8 basic (32-bit), i3 processor, 2.3 GHz of speed with memory of 3 GB and the language used is Java (net beans). Code of simulation is tested on one data center with one host and run on one Cloudlet.

4.1 Simulation Experiment:

We tried to perform a simulation experiments using CloudSim, we designed a Graphical user interface to dynamically change the properties of DC, VM and Cloudlet. Following tables provides the details of IaaS, PasS and SaaS properties that we used in our simulation.

S.No:	Property name	Description
1	Total Host	Total number of hosts on Data center
2	HostID	It is host id on which the VM will run
3	RAM	Memory Allocated to DC
4	Storage (MB)	Storage capacity
5	BW (Mbps)	Network Bandwidth
6	Architecture	X86
7	OS	Operating system used
8	Time Zone	Region

Table 4.1: IaaS Properties/Data center Parameters

S.No	Property Name	Description
1	Total VM	Total no. of VM used
2	VM ID	Unique Id of the VM
3	MIPS	Millions of Instruction per second
4	Size (MB)	Amount of Storage assigned to VM
5	RAM	Amount RAM assigned to VM
6	BW (Mbps)	Amount Bandwidth assigned to VM
7	Pes Number	No. of processing units allocated to VM
8	VMM Name	Virtual Machine Monitor/Hypervisor

Table 4.2: PaaS Properties/VM Parameters

S.No	Property Name	Description
1	Total Cloudlet (Gridlet)	No. of cloudlet
2	Cloudlet ID	The unique id of cloudlet,
3	Length	the length or size of this cloudlet to be executed in powerDatacenter
4	Filesize(Byte)	the file size (in Byte) of this cloudlet BEFORE submitting to a PowerDatacenter.
5	Outputsize(Byte)	the file size (in Byte) of this cloudlet AFTER finish execution by a PowerDatacenter.

Table 4.3: SaaS Properties/Cloudlet Parameters

4.2 Implementation:

We set the Host, VM and Cloudlet properties using the user interface we designed. Following tables show these properties.

S.No	Host Property	Value
1.	Total Host	1
2.	Host Id	0
3.	RAM	1024 MB
4.	Storage	10000
5.	BW	1000
6.	Architecture	X86
7.	OS	Linux
8.	Time Zone	10.0

Table 4.4: Host Properties

S.No	VM Property	Value
1.	Total VM	1
2.	VM Id	0
3.	MIPS	130
4.	Size (MB)	1000
5.	RAM	512
6.	BW	100
7.	Pes No.	1
8.	VMM	Xen

Table 4.5: VM Properties

S.No	Cloudlet Property	Value
1.	Total Cloudlet	1
2.	Cloudlet Id	0
3.	length	1700
4.	File Size	200
5.	Output	200

Table 4.6: Cloudlet Properties

We observed the total finished time of each cloudlet by varying the cloud lengths from 100 to 1000. Similarly, we change MIPS from 100 to 1000 and observed finished time

We plotted two graphs, first for comparison analysis between increasing MIPS and finished time, second for different cloudlet length and finished time.

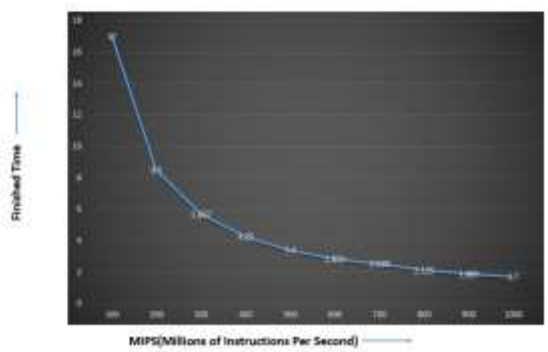

Figure 4.1 Comparative Analysis using graph between MIPS and Finished time

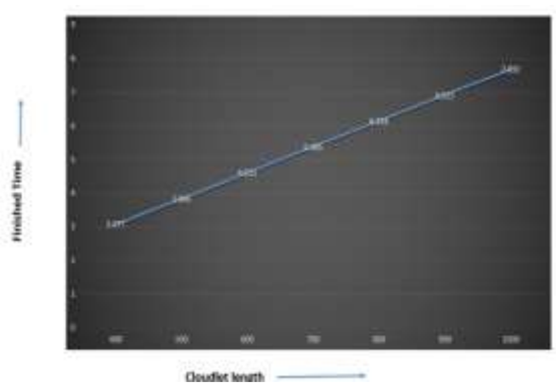

Figure 4.2: Comparative Analysis using graph between Cloudlet length and Finished time

IV. CONCLUSION

We have performed a simulation for a case study: Simulation Results for Virtual Cloud Environment with one Data Center, one Virtual Machine and one cloudlet using Time-shared provisioning policy. We also created a user-interface to dynamically change the properties of Host, VM and cloudlet. We have used this simulation tool to show the variation in cloudlet execution Finished time with different MIPS. As the MIPS is increasing, finished time is decreasing. It concludes that finished time of a cloudlet is inversely proportional to the MIPS. Similarly, with variation in Cloudlet length, finished time is also varying. With increasing Cloudlet length, finished time is also increasing. This concludes that cloudlet length and finished time are directly proportional.

Future Scope: Our experiment was limited to only one virtual machine and one cloudlet only. In future, we shall try to experiment with more virtual machine and cloudlets. We can also experiment with Space-shared provisioning using CloudSim.

REFERENCES

[1]. J. Geelan,"Twenty-one experts define cloud computing",Cloud Computing Journal, vol. 2, (2009),pp. 1-5.

[2]. Buyya, Rajkumar, Chee Shin Yeo, and Srikumar Venugopal. "Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities." In *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, pp. 5-13. Ieee, 2008.

[3]. Mishra, Mayank, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo. "Dynamic resource management using virtual machine migrations." *IEEE Communications Magazine* 50, no. 9 (2012).

[4]. Shaw, Subhadra Bose, and Anil Kumar Singh. "Use of proactive and reactive hotspot detection technique to reduce the number of virtual machine migration and energy consumption in cloud data center." *Computers & Electrical Engineering* 47 (2015): 241-254.

[5]. Nagesh, Bhavani B. "Resource Provisioning Techniques in Cloud Computing Environment-A Survey." *IJRCCT* 3, no. 3 (2014): 395-401.

[6]. Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." *Software: Practice and experience* 41, no. 1 (2011): 23-50.

[7]. 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.

[8]. Buyya, Rajkumar, and Manzur Murshed. "Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing." *Concurrency and computation: practice and experience* 14, no. 13- 15 (2002): 1175-1220.

[9]. 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.

[10]. Chappell D. Introducing the Azure services platform. White Paper, October 2008.

[11]. Dumitrescu CL, Foster I. GangSim: A simulator for grid scheduling studies. Proceedings of the IEEE International

Symposium on Cluster Computing and the Grid, Cardiff, U.K., 2005; 1151–1158.

[12]. El-Refaey, Mohamed A., and Mohamed Abu Rizkaa. "Virtual systems workload characterization: An overview." In *Enabling Technologies: Infrastructures for Collaborative Enterprises, 2009. WETICE'09. 18th IEEE International Workshops on*, pp. 72-77. IEEE, 2009.

[13]. Núñez, Alberto, Jose L. Vázquez-Poletti, Agustin C. Caminero, Gabriel G. Castañé, Jesus Carretero, and Ignacio M. Llorente. "iCanCloud: A flexible and scalable cloud infrastructure simulator." *Journal of Grid Computing* 10, no. 1 (2012): 185-209.

[14]. Sakellari, Georgia, and George Loukas. "A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing." *Simulation Modelling Practice and Theory* 39 (2013): 92-103.

[15]. Bahwaireth, Khadijah, Elhadj Benkhelifa, Yaser Jararweh, and Mohammad A. Tawalbeh. "Experimental comparison of simulation tools for efficient cloud and mobile cloud computing applications." *EURASIP Journal on Information Security* 2016, no. 1 (2016): 1-14.

[16]. Bhavani, B. H., and H. S. Guruprasad. "A comparative study on resource allocation policies in cloud computing environment." *Compusoft* 3, no. 6 (2014): 893.

[17]. Weblink: http://opensourceforu.com/2014/03/CloudSim-framework-modelling-simulating-cloud-environment/) accessed on 30/05/17.

[18]. W. Bhatia, R. Buyya, R. Ranjan CloudAnalyst: a CloudSim based visualmodeller for analysing cloud computing environments and applications, 24th IEEE International Conference on Advanced Information Networking and Applications (2010), pp. 446–452

[19]. G. Belalem, F.Z. Tayeb, W. Zaoui Approaches to improve the resources management in the simulator CloudSim R. Zhu, Y. Zhang, B. Liu, C. Liu (Eds.), ICICA 2010. LNCS, 6377, Springer, Heidelberg (2010), pp. 189–196.

[20]. Calheiros, Rodrigo N., Rajiv Ranjan, César AF De Rose, and Rajkumar Buyya. "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services."

[21]. Roy, Sudip, Sourav Banerjee, K. R. Chowdhury, and Utpal Biswas. "Development and analysis of a three phase cloudlet allocation algorithm." *Journal of King Saud University-Computer and Information Sciences* (2016).

[22]. Gulati, Ajay, and Ranjeev K. Chopra. "Dynamic round robin for load balancing in a cloud computing." *IJCSMC* 2, no. 6 (2013): 274-278.

[23]. Ahmad, Raja Wasim, Abdullah Gani, Siti Hafizah Ab Hamid, Muhammad Shiraz, Abdullah Yousafzai, and Feng Xia. "A survey on virtual machine migration and server consolidation frameworks for cloud data centers." *Journal of Network and Computer Applications* 52 (2015): 11-25.

[24]. Jamali, Shahram, Sepideh Malektaji, and Morteza Analoui. "An imperialist competitive algorithm for virtual machine placement in cloud computing." *Journal of Experimental & Theoretical Artificial Intelligence* 29, no. 3 (2017): 575-596.