Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 92 (2016) 38733-38737

ISSN: 2229-712X

Analysis of Static Load Balancing Algorithms for Cloud Computing

Abhay Kumar Agarwal* and Atul Raj

*Assistant Professor, Computer Science & Engineering Department, Kamla Nehru Institute of Technology , Uttar Pradesh, India.

ARTICLE INFO

Article history:

Received: 30 January 2016;
Received in revised form:
29 February 2016;
Accepted: 3 March 2016;

Keywords

Datacenter,
Cloud,
Distributed servers,
Load balancing.

ABSTRACT

A cloud consists of several elements such as clients, cloudlets, datacenter, distributed servers etc. While Cloud computing is a term, which involves virtualization, distributed computing, networking and software and web services. Centrally, the issue is to establish an effective load balancing in cloud. There are two types of load balancing algorithms available one is Static and other is Dynamic. In this paper we have studied three different Static load balancing algorithms i.e. Round Robin, Throttled, and Optimal. To compare each algorithm we have prepared a set-up and implemented each algorithm. Finally, an analysis has been done about the three implemented algorithms to find the best algorithm in terms of overall response time of cloudlet and Datacenter Processing time of cloudlet.

1. Introduction

Load balancing is the process of distributing the load among various nodes in a distributed system to achieve optimal resource utilization for maximizing throughput and minimizing overall response time. It helps avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. In cloud computing load balancing are used for balancing the load on virtual machine and cloud resources. When request generated by users are received by cloud hosting environment, the load balancer (load balancing algorithm) distribute the load over various cloud server so that all server should be utilized efficiently (no server is under loaded and over loaded) .So load balancing can be defined as method managing request coming from users so that all the server is utilized in such manner that no server is under loaded or overloaded.

In cloud computing there are two types of load balancing algorithms: (1) Dynamic load balancing algorithm and (2) Static load balancing algorithm. Further each type of load balancing algorithm is classified into many more. Talking about the Static algorithms it includes: Round Robin, Throttled, and Optimal. This paper presents the comparison of these algorithms. The comparison is done by implementing all these algorithms using the same parameters like: hardware; number of user base, number of Datacenters, Simulation Duration, Number of Regions, Operating System, software etc. The comparison is done for overall response time of cloudlet and Datacenter Processing time of cloudlet.

The other sections of the paper are: section 2 discusses about the cloud environment that includes cloud components, section 3 explains each of the three static load balancing algorithms there drawbacks are also discussed, in section 4 detail regarding the experiment set up is given, section 5 presents the comparison and the analysis, finally the conclusion with future scope is presented.

2. Cloud Environment

2.1 Cloud Component

A Cloud system consists of 3 major components [1] that are clients, datacenter, and distributed servers. All the three components are connected through internet for message passing. Each element has a definite purpose and plays a specific role.

Clients

End users interact with the clients to manage information related to the cloud. Clients are of three types as given below:

Mobile: Windows Mobile Smartphone, smart phones.

Thin: Such clients does not do any computation rather server do all the works for them. They only display the information. They also don’t have any internal memory.

Thick: These use different kind of browsers like Internet Explorer, Mozilla Firefox, Google Chrome etc. to connect to the Internet cloud.

Datacenter

Datacenter is a collection of servers that host different applications. An end user connects to the datacenter to subscribe different applications.

Distributed Server

A distributed server is a part of cloud which is present throughout the Internet hosting different applications. However the user feels that it is using the application from its own machine while using the application from the cloud.

3. Static Load Balancing Algorithms

The static load balancing algorithm requires prior knowledge of system resources, so that the decision of shifting of the load does not depend on the present state of system. Static algorithm performs better if prior knowledge of server is mentioned.

The various types of static load balancing algorithms are :

- Round Robin Scheduling Algorithm.
- Optimal Scheduling Algorithm.

Tele:
E-mail address: abhay.knit08@gmail.com

- Throttled Algorithm.

3.1 Round Robin Scheduling Algorithm

One of the simplest techniques for load balancing in cloud is called Round Robin, in which all processes are divided amid all available processors [2]. The allocation order of processes is maintained locally which is independent of the allocation from the remote processor as shown in figure 1. In this technique, the request is sent to the node having minimum number of connections, because of which at some point of time, few nodes may be heavily loaded and other remains idle.

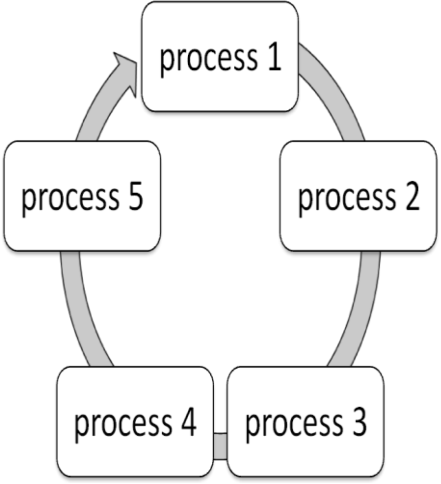

Fig 1. Round Robin load balancer

Round-robin is by far the simplest algorithm available to divide load among various nodes. It is generally the first choice when implementing a simple scheduler. One of the reasons for it being so easy is that the only information needed is a list of nodes [2].

However this is only when several key assumptions are true.

The nodes must be identical in capacity. Otherwise performance will degrade to the speed of slowest node in the cluster.

Two or more client connections must not start at the same time.

The jobs must be similar to achieve optimum load distribution among the various nodes. If a single node is more loaded than others it will become a bottleneck in the system [2].

Drawback of round robin: It simply works on time slicing. It allocates the load on various nodes on basis of time without considering the need of resource.

3.2 Active Monitoring Load Balancer (Optimal)

In equally spread current execution the random arrival of load in a cloud environment can cause of some server to be heavily loaded while other server is idle or only lightly loaded. Equally load distributing, improves performance. It does so by transferring load from heavily loaded server to lightly loaded servers. Efficient scheduling and resource allocation are critical characteristics of cloud computing on which the performance of the system is estimated [4]. The considered characteristics have an impact on cost optimization, which are obtained by improved response time and processing time.

In this algorithm the jobs are submitted by the clients to the computing system. As the submitted jobs arrive to the cloud they are stacked using a queue. The cloud manager estimates the job size and checks for the availability and capacity of the virtual machine. Once the job size and the available resource (virtual machine) size matched, the job scheduler allocates the identified resource to the job in queue as shown in figure 2.

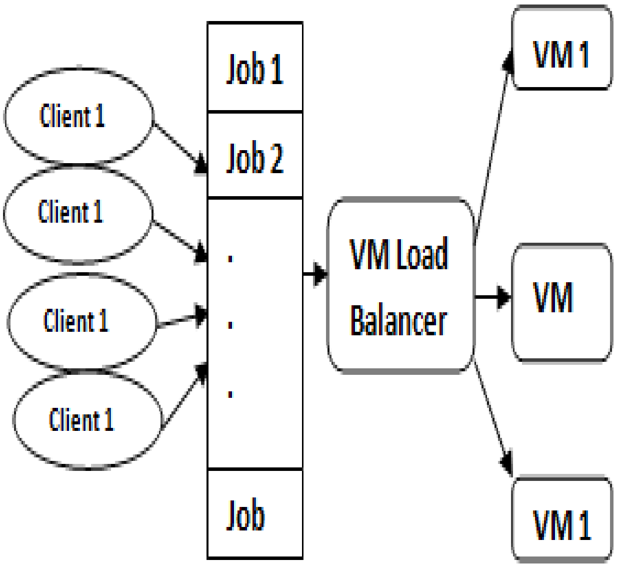

Fig 2. Active Monitoring Load Balancer

Unlike the round robin scheduling algorithm, there is no overhead of dividing the time slots to schedule the jobs in a periodic way [3]. The impact of the Active Monitoring Load Balancer (Optimal) algorithm is that there is an improvement in response time and the processing time. The jobs in Active Monitoring Load Balancer (Optimal) algorithms are spread equally, the complete computing system is load balanced and no virtual machines remain underutilized. Due to this advantage, there is considerable reduction in the virtual machine cost and the data transfer cost.

The Active Monitoring Load Balancer (Optimal) algorithm is mention below [3]:

```
[START]
Step1:- find the next available VM
Step2:-check for all current allocation count is less than max length of VM list allocate the VM
Step3:- if available VM is not allocated create a new one
Step 4:- count the active load on each VM
Step5:- return the id of those VM which is having least load
[END]
```

Drawback of Active Monitoring Load Balancer (Optimal): This algorithm checks all virtual machine in sequence for its availability to allocate resource on a free virtual machine.

3.3 Throttled Load Balancer

This algorithm is a static load balancing algorithm. In this algorithm we first check the index values of all the virtual machine in the system. The user request is sent to load balancer where it parses a table for the allocation of the resources in the system. It assigns the request to a particular load balancer which passes or responds reverse the request to the requester and updates the allocation policy [5]. After the successful allocation of the system the whole process for the de-allocation of the system also starts. This mechanism provides a higher amount of resource sharing and allocation in the system. The throttling threshold maintained generally is 1 however it can be modified easily.

This algorithm ensures that pre-defined numbers of cloudlets are allocated to a single VM at any given time. If there are more request groups are present then the number of available VM's at data centre than Throttled VM allocate incoming request as shown in figure 3. Otherwise queues until the next VM becomes available.

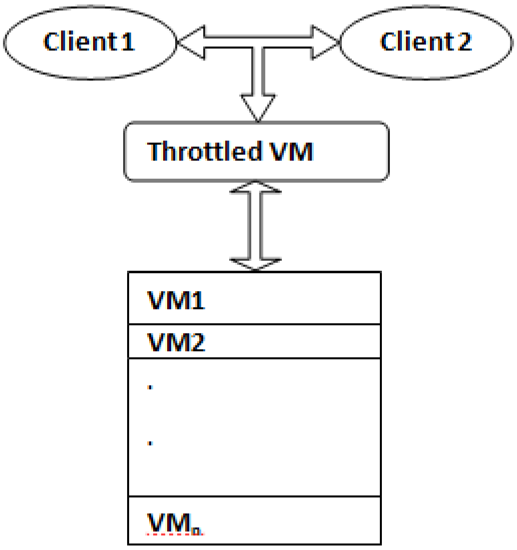

Fig 3. Throttled load balancer

The working of the throttled algorithm is understood in the following steps mention below [3]:

[START]
Step1:- find the last allocated VM.
Step2:-current allocation count is equal to last allocation +1.
Step3:-check for all current allocation count is less than max length of VM.
Step4:- if yes allocate request to current VM.
Step5:- No wait in queue go to step 3.
Step6:- return the id of those VM which is having least load.
[END].

Drawback of Throttled Load Balance: This algorithm is an improvement in Active Monitoring Load Balancer (Optimal) algorithm. This algorithm starts searching from last allocated virtual machine to nth virtual machine. But there is still a problem which is it do not utilises those virtual machine which become free after the execution of request.

The next section tells the detail about the experimental setup.

4. Experimental Set-Up

To implementation the different Static load balancing algorithms the common environment that includes hardware, software, simulator and other parameters considered are same for all the algorithms. The detailed about which are:

4.1 Hardware

For the implementation, hardware involves one computer system with the following specifications: processor Dual Core of 2.1 GHz, RAM of 2GB, Cache 2MB.

4.2 Software

The existing and proposed algorithm has been implemented on operating system Windows 7. The language used in implementing algorithms is Java (JDK 1.7) on IDE Eclipse Kepler 2014. Microsoft Excel and MatLab are used to draw graph for showing various results.

4.3 Simulator

CloudSim [6] is the most popular simulator tool available for cloud computing environment. It is an event driven simulator built upon the core engine of grid simulator GridSim [7] Java the most powerful object oriented programming language so it is used in CloudSim, because of its OOP feature, CloudSim modules can be easily extendable with the user's requirement. CloudSim has feature of modelling and creating a huge datacenter, unlimited number of virtual machines, introducing brokering policy and support the most important feature of cloud computing pay-as-you-go model. One of its unique features is federated policy, which is infrequently available to any other simulators.

Because of extendibility nature of CloudSim its popularity is increasing day by day. Due to the lack of many important features in new cloud simulators Yuxiang Shi proposed a method which minimize the uses of energy by "Linear Predicting Method" [14] (LPM) and "Flat Period Reservation-Reduced Method" [14] (FPRRM) that is made using CloudSim which minimizes utilization of energy in cloud [11]. G. Belalem had made an perspective to improve allocation of resource in CloudSim [12]. Y. Shi added file stripping and functions for data replication management in CloudSim. One of the drawbacks of CloudSim is lack Of GUI but can be achieved by using applets in java since it support java language.

4.4 Other Parameters

Number of User Base:	31
Number of Datacenter:	5
Simulation Duration:	60 min
Number of Region taken:	Globally divided in 6 region
Operating System:	Linux

Virtual Machine Manager: Xen (Para virtualization)

The next section presents the comparison and analysis of the results that are obtained on implementing the three static algorithms.

5. Comparison and Analysis

The comparison of the three static algorithms has been done on the basis of response time of Cloudlet and Datacenter processing time. The comparison has been done by calculating minimum time maximum time and average time for both the cases.

5.1 Response Time of Cloudlet

On analysing the results shown in table 1 about the overall response time of cloudlet (request from user base) obtained by simulating each algorithms it is found that Throttled load balancing is performs better than other two (Round Robin and optimal) static load balancing algorithms.

Table 1. Comparative analysis of overall response time of various load.

Static Time→ Algorithm↓	**Overall Response Time**		
	AVERAGE(ms)	***MINIMUM (ms)***	***MAXIMUM (ms)***
Round Robin	351.22	36.86	17755.03
Throttled	240.28	36.36	16395.53
Optimal	352.69	35.63	18052.26

The table 1 shows the average value, the minimum value and the maximum value of overall response time for each algorithm. The graphs are plotted using the data given in the table 1. Separate graph for average value, minimum value and the maximum values are plotted. It can be analysed from the graphs that the best performing algorithm is Throttled Algorithm.

Average value of response time of cloudlet

The chart 1 shows the overall response time in average case for different load balancing algorithms.

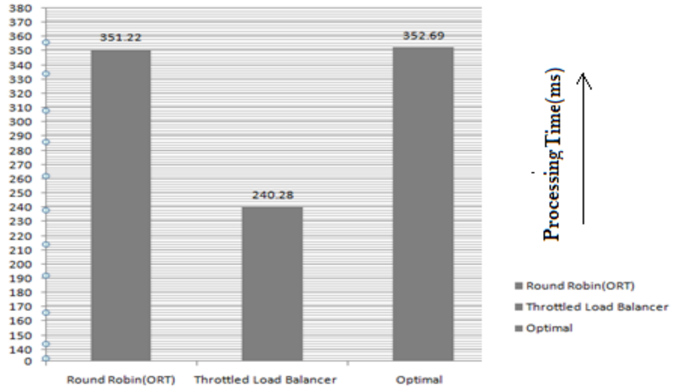

Chart 1. Showing average case comparison of overall response time of cloudlet of different load balancing algorithms in cloud computing.

Minimum value of response time of cloudlet

The chart 2 depicts overall response time in minimum case for different load balancing algorithms.

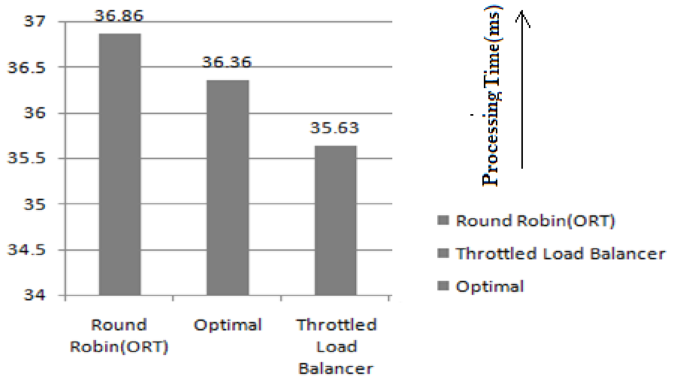

Chart 2. Showing minimum case comparison of overall response time of cloudlet of different load balancing algorithms in cloud computing.

Maximum value of response time of cloudlet

The chart 3 depicts overall response time in maximum case for different load balancing algorithms.

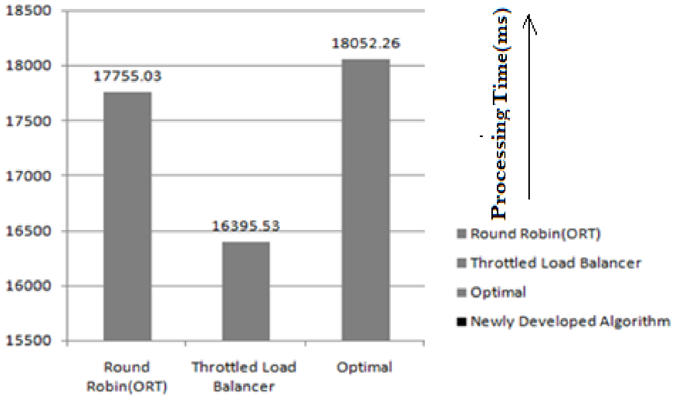

Chart 3. Showing maximum case comparison of overall response time of cloudlet of different load balancing algorithms in cloud computing

5.2 Datacenter Processing Time

On analysing the results shown in table 2 about the Datacenter processing time obtained by simulating each algorithm it is found that Throttled Load Balancing Algorithm performs better than the other two (Round Robin and optimal) static load balancing algorithms in cloud computing.

The table 2 shows the average value, the minimum value and the maximum value of Datacenter processing time for each algorithm. The graphs are plotted using the data given in the table 1. Separate graph for average value, minimum value and the maximum values are plotted. It can be analysed from the graphs that the best performing algorithm is Throttled Algorithm.

Table 2. Comparative analysis of datacenter processing time of various load balancing algorithm in cloud computing

Static Time→ Algorithm↓	Datacenter Processing Time		
	AVERAGE(ms)	*MINIMUM (ms)*	*MAXIMUM (ms)*
Round Robin	245.44	0.01	17701.55
Throttled	138.83	0.01	16338.52
Optimal	246.99	0.01	17999

The table 2 shows the average value, the minimum value and the maximum value of Datacenter processing time for each algorithm. The graphs are plotted using the data given in the table 1. Separate graph for average value, minimum value and the maximum values are plotted. It can be analysed from the graphs that the best performing algorithm is Throttled Algorithm.

Average Datacenter Processing Time

The chart 4 shows the overall response time in average case for different load balancing algorithms.

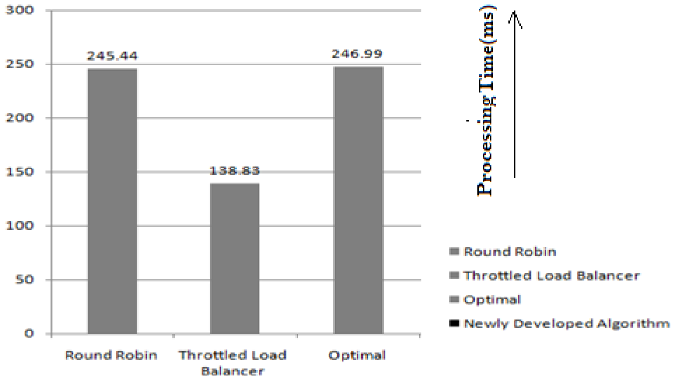

Chart 4. showing average case comparison of Datacenter Processing time of cloudlet of different load balancing algorithms in cloud computing.

Minimum Datacenter Processing Time

The chart 5 depicts overall response time in minimum case for different load balancing algorithms.

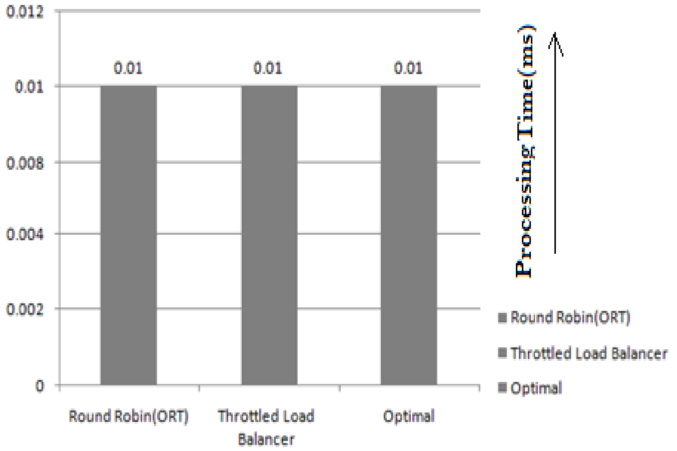

Chart 5. showing average case comparison of Datacenter Processing time of cloudlet of different load balancing algorithms in cloud computing.

Maximum Datacenter Processing Time

The chart 6 depicts overall response time in maximum case for different load balancing algorithms.

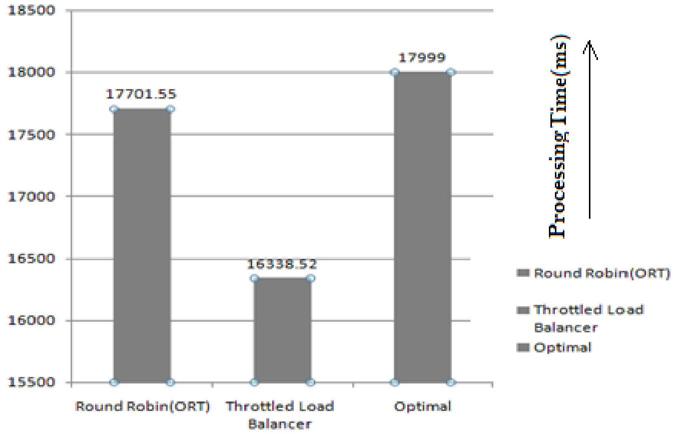

Chart 6. showing average case comparison of Datacenter Processing time of cloudlet of different load balancing algorithms in cloud computing

6. Conclusion and Future Scope

In this paper we have implemented the three Static load balancing algorithms for cloud computing. It is found on the basis of results obtained on implementing that the Throttled algorithm is best among the three static load balancing algorithms discussed. It is best in terms of response time of cloudlets and datacenter processing time. The average, minimum and maximum value of each comes out to be less for Throttled as compared with other algorithms i.e. Round Robin and Optimal.

In future one can make more efficient algorithm for load balancing which perform better that Throttled so the load balancing become more efficient in terms of datacenter processing time and response time.

References

Soumya Ray and Ajanta De Sarkar, “Execution analysis of load balancing algorithms in cloud computing environment,” International Journal on Cloud Computing: Services and Architecture (IJCCSA),Vol.2, No.5, October 2012.

Wei-Tek Tsai*, Xin Sun, Janaka Balasooriya “Service-Oriented Cloud Computing Architecture” Computer society 2010.

Pooja Samal1 and Pranati Mishra2, “Analysis of Variants in Round Robin Algorithms for Load Balancing in Cloud Computing”, (IJCSIT) International Journals of Computer Science and Information Technologies, Volume 4 (3), 2013, pg. number 416- 419.

Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid,” Availability and Load Balancing in Cloud Computing” IPCSIT vol.14 (2011).

Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing, 2010 IEEE 24th International Con-ference on Advanced Information Networking and Applications Workshops.

R. N. Calheiros, R Ranjan, A Beloglazov1, C A. F. De Rose, R. 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.

R. Buyya,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 2002.

Garg, S. K., & Buyya, R. (2011, December). “NetworkCloudSim: modelling parallel applications in cloud simulations.” In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on (pp. 105-113).

R. N. Calheiros, M .A. S. Netto, C. A. F. De Rose, and R. Buyya,“EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications,” Software-Practice and Experience, 2012.

F. Fittkau, S. Frey, W. Hasselbring, “CDOSim: Simulating Cloud Deployment Options for Software Migration Support”, IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems (MESOCA), 2012.

Y. Shi, X. Jiang, K. Ye, “An Energy-Efficient Scheme for Cloud Resource Provisioning Based on CloudSim”, IEEE International Conference on Cluster Computing, 2011.

G. Belalem, F. Z. Tayeb, and W. Zaoui, “Approaches to improve the resource management in the simulator CloudSim,” The International Conference on Information Computing and Applications, 2010.

S. Long, Y. Zhao, "A toolkit for modeling and simulating cloud data storage: an extension to CloudSim", International Conference on Control Engineering and Communication Technology (ICCECT), 2012.

Yuxiang Shi, Xiaohong Jiang,” An Energy-Efficient Scheme for Cloud Resource Provisioning Based on CloudSim”, IEEE International Conference on Cluster Computing,2011