International Journal of Civil Engineering and Technology (IJCIET)
Volume 9, Issue 7, July 2018, pp. 476–483, Article ID: IJCIET_09_07_049
Available online at http://www.iaeme.com/ijciet/issues.asp?JType=IJCIET&VType=9&IType=7
ISSN Print: 0976-6308 and ISSN Online: 0976-6316

© **IAEME** Publication **Scopus** Indexed

A COMPARATIVE ANALYSIS ABOUT LOAD BALANCING ALGORITHMS USING CLOUD SIMULATOR

Anusooya G, Vijayakumar V, Punitha.K, Sathyarajasekaran.K, Bharathiraja.S
School of Computing Science and Engineering,
VIT University, Chennai Campus, Tamil Nadu, India

ABSTRACT

Cloud load balancing is the process of distributing workloads across multiple computing resources in a cloud environment. Load distribution in cloud computing systems is more challenging than in other systems. The purpose of the paper is to address the issue of optimal task dispatching on multiple Virtual Machines (VM's) with efficient power management. The prime goal is to address the load distribution in multiple VM's and to propose an algorithm to maintain a minimized task response time and minimized power consumption using CloudSim cloud simulator.

Key words: Cloud Computing, load balancing, Virtual Machines (VM's), CloudSim, Cloudlet, Broker, Datacenter.

Cite this Article: Anusooya G, Vijayakumar V, Punitha.K, Sathyarajasekaran.K and Bharathiraja.S, A Comparative Analysis About Load Balancing Algorithms Using Cloud Simulator, International Journal of Civil Engineering and Technology, 9(7), 2018, pp. 476–483.
http://www.iaeme.com/ijciet/issues.asp?JType=IJCIET&VType=9&IType=7

1. INTRODUCTION

Load balancing in the cloud environment can be defined as the process of distributing or sharing the workload amongst various computer resources (servers, computers etc.). In respect to cloud computing, load balancing is the most challenging or daring task. It is challenging in the way – energy consumption, power management and to decrease the average response time. [Li Keqin (2017)]

In cloud computing utilization of resources and proper energy consumption along with proper load balancing is considered as an important aspect [Sallami N. M. Al (2013)]. Above mentioned features utilize the resources in a such a manner that will decrease the carbon emition, energy utilization and results in achieving green computing. [Kaur Tarandeep and Chana Inderveer (2015)]

In this paper we will propose an algorithm which will achieve minimized energy consumption and average power consumption on multiple Virtual Machines (VM's) and will show a comparative analysis between various load balancing algorithms like First Come First Serve(FCFS), Shortest Job First(SJF), Opportunistic Load Balancing and General Prioritized Load balancing algorithm [Mishra Nitin Kumar and Mishra Nishchol (2015)]. The above stated algorithm has been studied and applied to check the performance of each algorithm using CloudSim 3.0 and find out performance of the algorithms through the overall execution time taken by each algorithm to complete the task.

2. CLOUDSIM 3.0: CLOUD SIMULATOR TOOLKIT

CloudSim is just a framework which is used for simulating the cloud infrastructure and services. CloudSim cannot be implemented as a standalone simulator tool. It has a .java extension, so it requires an Java IDE to be implemented. The IDE can be either NetBeans or Eclipse. In this paper we implemented CloudSim using Eclipse IDE. Before implementing CloudSim here are the following prerequisites: -

- Eclipse IDE for Java Developers.
- CloudSim 3.0.3
- Apache Commons Math 3.6.1

Following steps show how to implement CloudSim simulator in Eclipse IDE: -

- Unzip CloudSim 3.0.3 and Apache Commons Math in same folder. You just need commons-math3-3.6.1.jar and delete the rest of the apache common jar files.

Figure1 CloudSim Extracted folder

- Now run Eclipse IDE and create a New Java Project.

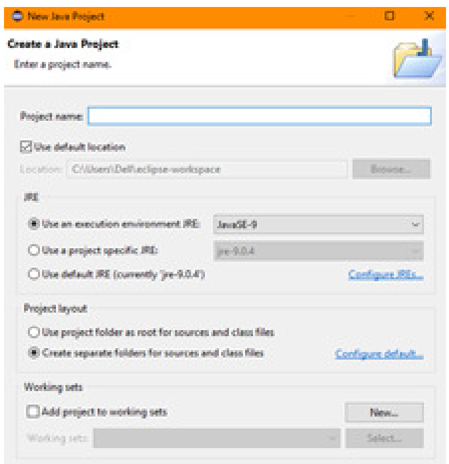

Figure 2 Create a new java project

- Now give project name as your choice and Untick the Use Default Location checkbox and browse for the CloudSim folder.

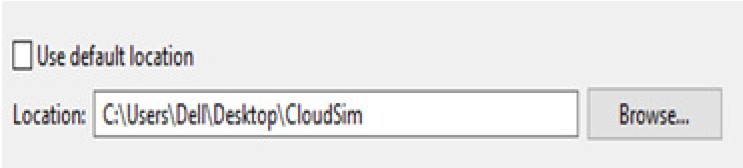

Figure 3 Select the CloudSim folder location for project workplace

- Just click Next and you will see that your Project will be created with your name you gave in the new project name option. And under that you will see CloudSim packages are there.

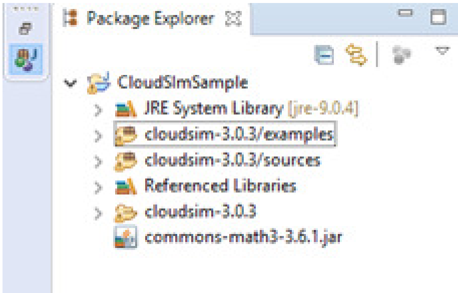

Figure 4 Your CloudSim is implemented in your project.

- CloudSim is now implemented in Eclipse IDE and you can start working on CloudSim cloud simulator.

2.1. CLOUDSIM TERMINOLOGY'S AND CONFIGURATION

- Cloudlets: These are related to the cloud-jobs or tasks.
- Users: These sent the tasks to the cloud.
- Datacenter: It is just the repository of the virtual servers that hots various applications.
- Broker: Broker is an intermediator which handles the incoming requests and distribute among various VM's.
- Virtual Machines (VM's): The System which actually process the cloudlet or the cloud job.

2.2. CLOUDSIM CONFIGURATION

The below table show the various configuration set-up of cloudsim like number of users and vm's along with cloudlets. It also discusses about the configuration of the system to create a datacenter.

Table 1 Simulation Parameters

SIMULATION PARAMETERS	
PARAMETERS	VALUES
Number of Users	1
Number of VM's	5
Number of Cloudlets	20

Table 2 Cloudlets Details

CLOUDLETS DETAILS	
PARAMETERS	VALUES
File Size	4000
Input Size	300
Output Size	300
PES	1

Table 3 VM Details

VM DETAILS	
PARAMETERS	VALUES
Name of VM	Xen
RAM	512
MIPS	250
BW	1000

Table 4 Datacenter Details

DATACENTER DETAILS	
PARAMETERS	VALUES
Number of Datacenters	1
RAM	102400
MIPS	102400
Storage	1000000

The above table has 4 sub tables that shows the CloudSim configuration which remains same for testing of the algorithms which were discussed in the Introduction section of the paper. [Maltare Jayprakash and Prajapat Balwant (2016)]

3. SIMULATION OF VARIOUS LOAD BALANCING ALGORITHMS

In this section we will be analyzing various load balancing algorithms such as FCFS, SJF and others [Santra Soumen and Dr. Mali Kalyani (2015)]. Following are the four load balancing algorithms which are simulated in the CloudSim Simulator and following outputs were obtained: -

FCFS (First Come First Serve)

As the name says the task which comes first is allotted to the virtual machine which processes it and gives the final output. This algorithm was put under simulation and following output was observed: -

```
.1: Broker: Sending cloudlet 0 to VM #0
.1: Broker: Sending cloudlet 1 to VM #1
.1: Broker: Sending cloudlet 2 to VM #2
.1: Broker: Sending cloudlet 3 to VM #3
.1: Broker: Sending cloudlet 4 to VM #4
.1: Broker: Sending cloudlet 5 to VM #0
.1: Broker: Sending cloudlet 6 to VM #1
.1: Broker: Sending cloudlet 7 to VM #2
.1: Broker: Sending cloudlet 8 to VM #3
.1: Broker: Sending cloudlet 9 to VM #4
.1: Broker: Sending cloudlet 10 to VM #0
.1: Broker: Sending cloudlet 11 to VM #1
.1: Broker: Sending cloudlet 12 to VM #2
.1: Broker: Sending cloudlet 13 to VM #3
.1: Broker: Sending cloudlet 14 to VM #4
.1: Broker: Sending cloudlet 15 to VM #0
.1: Broker: Sending cloudlet 16 to VM #1
.1: Broker: Sending cloudlet 17 to VM #2
.1: Broker: Sending cloudlet 18 to VM #3
.1: Broker: Sending cloudlet 19 to VM #4
```

Figure 5: The tasks were allotted to VM in the way they were received.

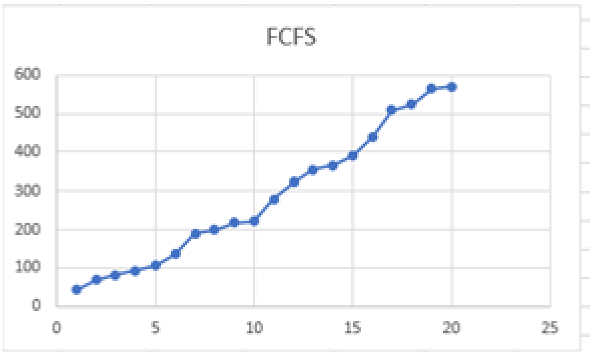

Figure 6: It shows the total execution time taken by FCFS algorithm

SJF (Shortest Job First)

In this algorithm the task or the cloudlet which have the least cloudlet size is allotted to the virtual machine. Following were the output observed: -

```
cloudletlist size = 10500
cloudletlist size = 17000
cloudletlist size = 23500
cloudletlist size = 20223
cloudletlist size = 26723
cloudletlist size = 23446
cloudletlist size = 29946
cloudletlist size = 26669
cloudletlist size = 33169
cloudletlist size = 29892
cloudletlist size = 36392
cloudletlist size = 33115
cloudletlist size = 39615
cloudletlist size = 36338
cloudletlist size = 42838
cloudletlist size = 39561
cloudletlist size = 46061
cloudletlist size = 42784
cloudletlist size = 49284
cloudletlist size = 46007
```

Figure 7: It shows the cloudlet size of 20 cloudlets

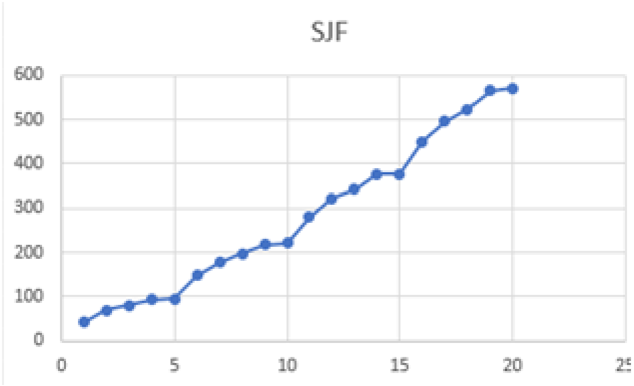

Figure 9: Total Execution Time of SJF

OLB (Opportunistic Load Balancing)

In this algorithm the tasks are assigned in the first come first serve manner at the initial but later the tasks are assigned to the virtual machine is free i.e. the VM grab the opportunity and processes the task. Following were the outputs observed: -

```
0.1: Broker: Sending cloudlet 1 to VM #0
0.1: Broker: Sending cloudlet 2 to VM #1
0.1: Broker: Sending cloudlet 3 to VM #2
0.1: Broker: Sending cloudlet 4 to VM #3
0.1: Broker: Sending cloudlet 5 to VM #4
0.1: Broker: Sending cloudlet 6 to VM #0
0.1: Broker: Sending cloudlet 7 to VM #1
0.1: Broker: Sending cloudlet 8 to VM #3
0.1: Broker: Sending cloudlet 9 to VM #2
0.1: Broker: Sending cloudlet 10 to VM #4
0.1: Broker: Sending cloudlet 11 to VM #0
0.1: Broker: Sending cloudlet 12 to VM #3
0.1: Broker: Sending cloudlet 13 to VM #1
0.1: Broker: Sending cloudlet 14 to VM #4
0.1: Broker: Sending cloudlet 15 to VM #2
0.1: Broker: Sending cloudlet 16 to VM #0
0.1: Broker: Sending cloudlet 17 to VM #3
0.1: Broker: Sending cloudlet 18 to VM #1
0.1: Broker: Sending cloudlet 19 to VM #4
0.1: Broker: Sending cloudlet 20 to VM #2
```

Figure 10: It shows the Allocation fashion of OLB Algorithm

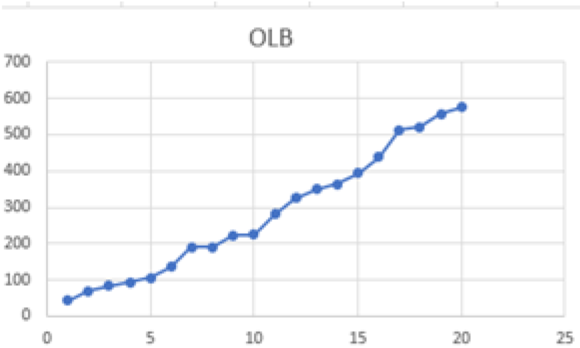

Figure 11: Total Execution time of OLB Algorithm

GP (Generalized Priority)

This is a priority-based algorithm in which the tasks are allocated to the VM on the basis of the priority of the cloudlet. Following were the output observed: -

```
Broker: Sending cloudlet 19 to VM #4
Broker: Sending cloudlet 17 to VM #2
Broker: Sending cloudlet 20 to VM #0
Broker: Sending cloudlet 15 to VM #1
Broker: Sending cloudlet 18 to VM #3
Broker: Sending cloudlet 13 to VM #4
Broker: Sending cloudlet 16 to VM #2
Broker: Sending cloudlet 11 to VM #0
Broker: Sending cloudlet 14 to VM #1
Broker: Sending cloudlet 9 to VM #3
Broker: Sending cloudlet 12 to VM #4
Broker: Sending cloudlet 7 to VM #2
Broker: Sending cloudlet 10 to VM #0
Broker: Sending cloudlet 5 to VM #1
Broker: Sending cloudlet 8 to VM #3
Broker: Sending cloudlet 3 to VM #4
Broker: Sending cloudlet 6 to VM #2
Broker: Sending cloudlet 4 to VM #0
Broker: Sending cloudlet 2 to VM #1
Broker: Sending cloudlet 1 to VM #3
```

Figure 12: Allocation of cloudlets to the VM on the priority-basis.

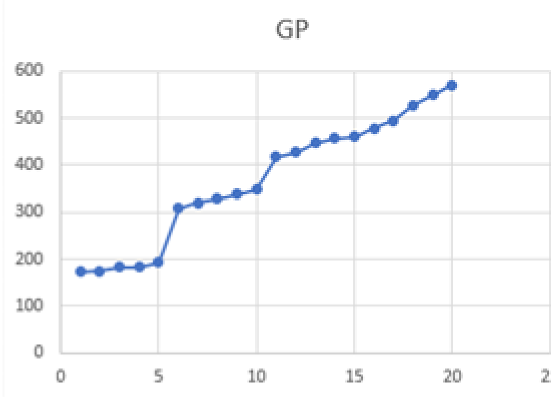

Figure 13: Execution time taken by GP Algorithm

4. ANALYSIS AND SIMULATION RESULT

Table 5: Simulation Results

Algorithms	Average Waiting Time (m/s)	Total Execution Time (m/s)
FFCS	153.33	570.52
SJF	152.03	570.73
OLB	153.28	575.25
GP	238.86	570.73

With the numerical figures we can clearly see that FCFS is faster and took less time to complete all the cloudlet execution. OLB took the most time to complete the execution. Also, we can see the cloudlet has to least wait in SJF and has to wait more in GP.

So, with the above result we can analyze that FCFS is the most efficient algorithm when it comes to execution time and SJF is efficient when it comes to average waiting time. The least efficient is OLB in case of total execution time and GP the worst in case of average waiting time.

5. CONCLUSION AND FUTURE WORK

In this paper we came across four load balancing algorithm which are FCFS, SJF, OLB and GP. We implemented these algorithms using CloudSim simulator and observed various outputs. Later we found the results and analyzed the result to find out the efficient algorithm among the four. In future work we will propose a load balancing algorithm which will take least execution and least average waiting time which will result in decreased power consumption and reduced response time which is the main goal to be achieved.

In the future we will implement an algorithm which will take the least execution time and in which the cloudlet has to least wait. With these parameters we will achieve minimized power consumption and optimal task dispatching.

REFERENCES

[1] Anjali, Grover J., Singh M., Singh C., and Sethi H. (2015): *A new approach for dynamic load balancing in cloud computing*, Proc. Nat. Conf. Adv. Eng. Technol. Manage., pp. 30–36.

[2] Beloglazoy A., Abawajv J., and Buyya R (2012): Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Gen. Comput. Syst., vol. 28, pp. 755–768

[3] Beloglazov A., Buyya R., Lee Y. C., and Zomaya A (2011): A taxonomy and survey of energy-efficient data centers and cloud computing systems, Adv. Comput., vol. 82, pp. 47–111.

[4] Kaur Tarandeep and Chana Inderveer (2015): Energy efficiency techniques in cloud computing: A survey and taxonomy.,ACM Comput. Surv. 48, 2, Article 22.

[5] Li Keqin (2017): Optimal Task Dispatching on Multiple Heterogeneous Multiserver Systems with Dynamic Speed and Power Management, IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, VOL. 2, NO. 2

[6] Maltare Jayprakash and Prajapat Balwant (2016): Dynamic Load Balancing in Cloud Computing using CloudSim, International Journal of Computer Applications (0975 – 8887) Volume 148 – No.5

[7] Mishra Nitin Kumar and Mishra Nishchol (2015) :Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing-A Systematic Review , International Journal of Computer Applications (0975 –8887) Volume 131 –No.18

[8] Purba Komal and Bhagat Nitin (2014): A Review on Load Balancing Algorithm in Cloud Computing, SSRG International Journal of Computer Science and Engineering (SSRG-IJCSE)

[9] Sallami N. M. Al (2013): Load balancing in green cloud computation, Proc. World Congress Eng., Lecture Notes in Engineering and Computer Science, vol. 2205, pp. 798–802.

[10] Santra Soumen and Dr. Mali Kalyani (2015); A New Approach to Survey on Load Balancing in VM in Cloud Computing: using CloudSim, IEEE International Conference on Computer, Communication and Control.

[11] Gangu Dharmaraju, J. Divya Lalitha Sri and P. Satya Sruthi, A Cloud Computing Resolution in Medical Care Institutions for Patient's Data Collection. International Journal of Computer Engineering and Technology, 7(6), 2016, pp. 83–90.

[12] Dr. V. Goutham and M. Tejaswini, A Denial of Service Strategy To Orchestrate Stealthy Attack Patterns In Cloud Computing, International Journal of Computer Engineering and Technology, 7(3), 2016, pp. 179–186.

[13] Damodar Tiwari, Shailendra Singh and Sanjeev Sharma, A Prediction Based Multi- Phases Live Migration Approach to Minimize the Number of Transferred Pages, in Cloud Computing Environment, International Journal of Computer Engineering & Technology, 9(3), 2018, pp. 23–31.

[14] A G P Kujur and Ajay Oraon, Cloud Computing and Security: A Review. International Journal of Advanced Research in Engineering and Technology, 7(6), 2016, pp 01–06.

[15] Hamad Balhareth, Cloud Computing Strategy and Adoption in Higher Education: The Case of Saudi Arabia. International Journal of Information Technology & Management Information System 9(1), 2018, pp. 30–38