Priority Alteration Scheme for Load balancing in Virtual Machines of Cloud Computing

Anitha K L[1], T.R. Gopalakrishnan Nair[2]
[1]Research Scholar, Bharathiar University, Coimbatore,India
Networks and Security Research Group, Advanced Reseach Centre, RRCE, Rajarajeswari Group of Institutions, Bangalore, India
(E-mail: anithakl07@gmail.com)

[2] Networks and Security Research Group, Advanced Reseach Centre, RRCE, Rajarajeswari Group of Institutions, Bangalore, India
Visiting Professor, NIAS, Bangalore, India
(E-mail: trgnair@yahoo.com)

***Abstract*—**Load balancing permits the organizations to handle the workloads or tasks demands by allocating the resources amongst multiple computers, servers or networks in a cloud computing environment. Anytime an Enterprise moving to the cloud greatly facilitates the deployment of applications across multiple geographically distributed data-centers. In such cases load balancing in cloud computing environment has become a very challenging and important research area. In this paper, we focus on cloud data center approaches and load balancing policies in cloud computing and we propose net priority scheduling for optimal task scheduling for virtual machine load balancing.

Keywords*—load balancing; virtualization; data center; scheduling.*

I. INTRODUCTION

Load Balancing is one of the key aspects of cloud computing environment. The load can be a memory, CPU capacity, network or delay load. Load balancing is a process of distributing loads from heavily loaded nodes to lightly loaded nodes thereby the workload can be shared among different nodes of distributed system. It helps to improve the resource utilization and for enhanced performance of the system. Consequently we can avoid the situation where nodes are either heavily loaded or under loaded in the network. Efficient load balancing scheme ensures efficient resource utilization by provisioning the resources to cloud user's on a demand driven basis and task scheduling in distributed environment. Load Balancing may even support prioritizing users by applying appropriate scheduling criteria. Cloud Computing utilizes the virtualization technology for dynamic supply of virtual computing and storage resources based on varying needs of the users. The key benefits of cloud computing include virtualized resources, hiding and abstraction of complexity and efficient use of distributed resources. The effective asset provisioning and booking of assets as well as tasks will guarantee that the resources are exceptionally available on demand; the assets are adequately used under any high/low load; vitality is saved when the usage of cloud assets is beneath certain limit; lessen expense of utilizing assets. The examples of the Cloud computing platform include Google App Engine [1], IBM blue Cloud [2], Microsoft Azure [3].

Load Balancing provides an efficient solution to various issues residing in the environment set-up and usage of cloud computing. Simulation set up is required to measure the efficiency and effectiveness of load balancing algorithms. CloudSim is one of the efficient tools that can be used for modeling of cloud [4]. CloudSim allows virtual machines to be managed by hosts which in turn are managed by data centers. CloudSim architecture consists of four entities: Data centers, Hosts, Virtual machines, Application as well as System Software. These entities allow the user to set up a cloud computing environment and measure the efficiency of the Load Balancing algorithms. Datacenters provides infrastructure level service to Cloud consumers. Host in the cloud are the physical servers which have pre-configured processing capabilities. It provides Software level services to the cloud consumers. Virtual machines allow development as well as deployment of custom application service models and are mapped to a host which matches their critical characteristics like storage, memory, processing, and software and availability requirements. Similar instance of virtual machine are mapped to same instance of host based upon availability. The system and application software's are executed on virtual machine on-demand.

The paper is organized as follows. Section II provides essential foundations which cover cloud data center with virtualization technology and resource utilization by efficient load balancing scheme. Section III discusses related work and describes the usage and functionalities of existing load balancing algorithms and policies. Simulation is carried out for scheduling the tasks and the implementation is done in Section IV and Section V concludes the paper.

II. VIRTUALIZATION

Cloud data center can be a distributed network which consists of distributed components such as processing, storage, servers and resources. Each resource has its own properties such as CPU, memory, network bandwidth and so on. In traditional approach of data center, the applications are tied to specific servers and storage sub systems that are often over-provisioned to deal with workload surges and unexpected failures, which are expensive to maintain with wasted energy and floor space, low resource utilization and significant management overheads [5]. With Virtualization technology, cloud data centers become more secure and flexible. In today's cloud data center, applications are loosely coupled to the underlying infrastructure so that the resources can be shared among themselves.

Load balancing schemes depending on whether the system dynamics are important can be either static or dynamic [6]. The cloud provider installs homogeneous resources in static environment and heterogeneous resources in dynamic environment. The resources are not flexible in static and the cloud needs prior knowledge of the nodes, memory, processing power and the statistics of user requirements, where as in dynamic environment; the resources are flexible and can easily adapt to run time changes in load. The cloud computing revolutionize the way we interact with the resources via Internet [7]. Virtualization is the creation of virtual resources and is one of the effective ways to reduce the IT expenses. Cloud models used virtualization technology which helps in creating a single data centre or high power server to act as compound machines. It also depends on the hardware pattern of the data center or server in how may virtual machine they can be separated. To apply virtualization extra software is required [8]. Cloud computing, creates a virtual group of resources like networks, storage, processing units and memory to carry out the user's resource requirement and offers on demand hardware and software [9].

III. LOAD BALANCING

A. *Metrics for Load balancing*

a) Fault tolerance

Fault tolerance is the property that implements a system to continue working properly in the result of the failure of one or more nodes or components. Fault tolerance can be achieved by efficient load balancing techniques in the cloud computing environment. Random failures could be compensated by load balancing algorithms also.

b) Migration time

Migration time is the total time needed for a process to be relocated from one node to another node for execution in a cloud computing environment. Migration time should be minimized to achieve better performance of the cloud system.

c) Response time

Response time is the total time taken for getting a response once the request is initiated. Response time should be minimized for better performance by using load balancing techniques.

d) Resource utilization

Resource utilization increases the energy efficiency of the system. By means of efficient load balancing, optimum resource should be utilized.

e) Throughput

Throughput is a measure of the total number of tasks which has executed within a given span of time. To achieve high performance it is needed to have high throughput.

f) Scalability

Scalability is the capability of an algorithm to perform load balancing to persist to function well for any finite number of nodes when it is changed in size or volume with the purpose of meeting a user need.

g) Single Point of Failure

If a node fails, and will stop the entire system from working is called single point of failure (SPOF). We need to use load balancing schemes wherein there should not be any single point of failure.

B. *Load Balancing Algorithms*

(i) Vector Dot

Vector Dot [10] uses the environment having data centers with integrated server and storage virtualization. It uses dot product to distinguish node based on the requirement of items. It also handles hierarchical and multidimensional resource constraints and removes overloads on server, switch and storage.

(ii) Compare and Balance

Compare and Balance [11] uses Intra-Cloud environment. Adaptive live migration of VMs will be done based on sampling process and also balances the load amongst servers and reaches equilibrium fast.

(ii) Carton

Carton [12] uses a unifying framework for cloud control. It is simple, easy and very low computation and communication overhead. Load balancing can be performed to minimize the associated cost.

(iii) Scheduling Strategy on LB of VM resources

It uses cloud computing environment and by Genetic Algorithm, the historical data and current state of system will be maintained. It achieves best load balancing and reduce dynamic migration [13].

(iv) Biased Random Sampling

Biased Random Sampling [14] has large scale Cloud Systems. It uses random sampling of system domain. It also achieves load balancing across all system nodes.

Load balancing in the cloud needs new framework to adapt transformation. Load balancing plays an important role in improving the performance and maintaining stability [15].

C. Load Balancing Policies

The quality of service can be improved by maximizing throughput, minimizing the response time and sufficient capacity in the data center to offer more resources during peak traffic. Here we use three load balancing policies across virtual machines in a single data center to distribute the load and checks the performance time and cost.

1. Round Robin

Round Robin is one of the simplest scheduling policies that use the concept of time slices. Here the time is divided into multiple slices and each node is given a particular time interval to perform its operations. The service provider allocates the resources to the user on the basis of time slice given. If the time slice given is extremely large then Round Robin scheduling policy follows FCFS scheduling. If the time quantum is too small then it selects the load on random basis and lead to a situation where some nodes are heavily loaded. However, there is an additional load on the scheduler to decide the size of quantum [16] and it has longer average waiting time, high turnaround time and low throughput.

2. Equally Spread Current Execution (ESCE)

In Equally Spread Current Execution load balancing policy, the load balancer distributes load to all virtual machines connected in the data center. Here the tasks are equally spread, takes less time, increases throughput. The proper utilization of virtual machines results in a load balanced system. Here the load balancer sustains an index table of Virtual Machines and the number of requests currently assigned to the Virtual Machine (VM) [17]. If any request comes from the data center to allocate a new VM, the least loaded VM is identified from the index table. The data center communicates the requests to the allocated VM and revises the index table by increasing the allocation count. When the task completes, the load balancer revises the index table by decreasing the allocation count; however there is an additional computation overhead to scan the queue repeatedly.

3. Throttled

In Throttled load balancing policy, the load balancer sustains an index table of virtual machines and their states (Busy/Available). The data center receives a new request from client/server to find an appropriate virtual machine to perform the recommended task. The data centre queries the load balancer for allocation of the VM. The load balancer parses the index table from top until the first available VM is found or the index table is parsed fully. If the VM is found, it sends the VM id to the data centre. Further, the data centre acknowledges the load balancer of the new allocation and the data centre updates the index table accordingly. While processing the request of client, if appropriate VM is not found, the load balancer returns -1 to the data centre [17]. Once the task completes the load balancer de-allocates the same VM whose id is already communicated. The throughput of the computing model can be estimated as the total number of jobs executed within a time span without considering the virtual machine allocation time and destruction time.

4. Load balancing Min-Min (LBMM)

Load balancing Min-Min (LBMM) [18] is a dynamic load balancing algorithm. This method makes use of Opportunistic load balancing algorithm wherein it keeps each node busy in cloud exclusive of considering the execution time of a node thereby causing bottleneck in the system. This problem is solved by the LBMM three layer architecture in which, the request manager in the first layer receives the task and assigns to service manager in the second level. Here the requests will be divided into various sub tasks. Service manager will be assigning the sub tasks for execution to the service node.

IV. SIMULATION

We use cloudsim [19] for the simulation set up and the simulation is carried out for FCFS (First Come First Serve) scheduling and SJF (Shortest Job First) scheduling. We use 5 virtual machines (VMs) and 20 cloudlets with a datacenter for the experimental setup. Table 1 and table 2 shows FCFS and SJF scheduling with 5 virtual machines.

VM Parameters: 512 MB Ram, 1000 Image size, 250 Mips, 1000 band width, 1 CPU, Xen VMM.

Cloudlet parameters: length 1000, filesize 400, outputsize 400

A. FCFS Scheduling

TABLE I FIRST COME FIRST SERVE SCHEDULING

Cloudlet ID	VM ID	Time	Start Time	Finish Time
0	0	41.83	0.1	41.93
1	1	68.55	0.1	68.65
3	3	82.2	0.1	82.3
2	2	92.88	0.1	92.98
4	4	104.79	0.1	104.89
5	0	93.41	41.93	135.34
6	1	120.75	68.65	189.39
7	2	105.41	92.98	198.39
8	3	134.83	82.3	217.13
9	4	117.22	104.89	222.11
10	0	144.99	135.34	280.33
11	1	133.52	189.39	322.92
12	2	156.58	198.39	354.97
13	3	147.71	217.13	364.85
14	4	167.99	222.11	390.1
15	0	157.61	280.33	437.94
16	1	185.73	322.92	508.64
17	2	169.1	354.97	524.07
18	3	200.34	364.85	565.18
19	4	180.42	390.1	570.52

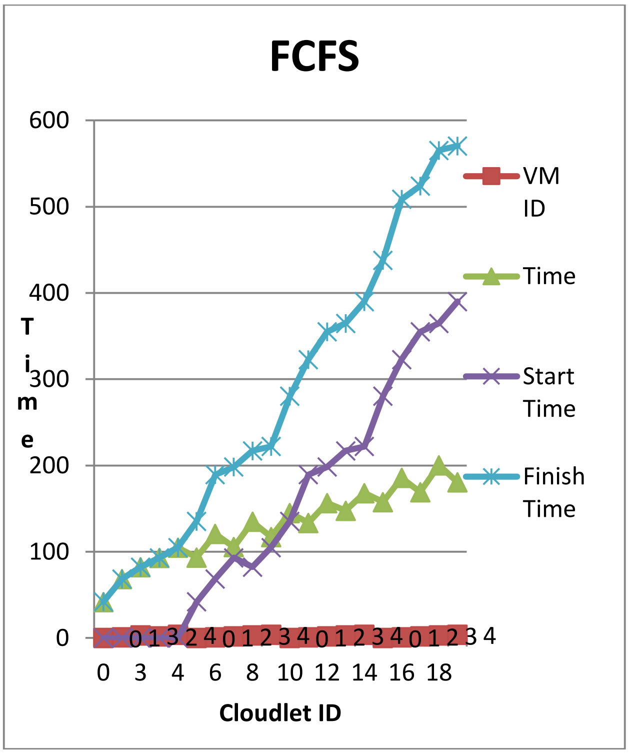

Fig. 1. FCFS Scheduling

B. SJF Scheduling

TABLE II SHORTEST JOB FIRST SCHEDULING

Cloudlet ID	VM ID	Time	Start Time	Finish Time
0	0	41.83	0.1	41.93
1	1	68.55	0.1	68.65
3	2	79.93	0.1	80.03
2	4	92.15	0.1	92.25
5	3	95.31	0.1	95.41
7	0	106.25	41.93	148.18
4	1	107.75	68.65	176.4
9	2	118.15	80.03	198.18
6	3	121.73	95.41	217.13
11	4	129.86	92.25	222.11
8	0	132.14	148.18	280.33
13	1	146.52	176.4	322.92
10	2	143.84	198.18	342.02
12	4	155.35	222.11	377.46
15	3	160.82	217.13	377.95
17	0	170.45	280.33	450.78
14	1	172.73	322.92	495.65
19	2	181.84	342.02	523.86
16	3	187.24	377.95	565.19
18	4	193.27	377.46	570.73

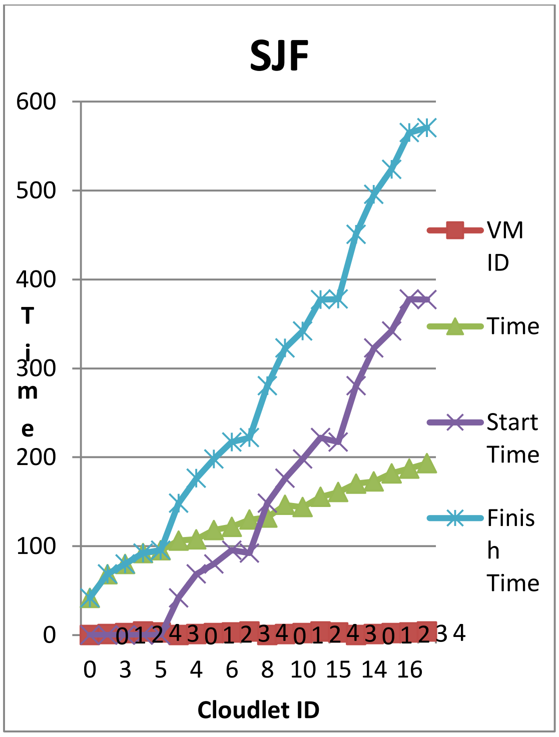

Fig. 2. SJF Scheduling

Fig. 1 and Fig. 2 shows the implementation of FCFS and SJF scheduling with start time and finish time for the incoming requests to be executed. High priority will be given for first incoming task in FCFS scheduling. And in SJF scheduling shortest task will be executed first by giving high priority.

C. Net Priority (NeP) Scheduling

Let us consider the priority values for FCFS as A0 and SJF as A1. A0 and A1 will be estimated based on the resource properties taken. We will assume A0 as 0.6 and A1 as 0.4 for calculating the net priority. This is given based on an optimality giving importance to the arrival sequence. However A0 and A1 are tunable based on the current running scenario dashboard estimations and server management policies.

A linear combination of A0 and A1 into the net priority (NeP) can be calculated as

$$NeP = A0 * FCFS + A1 * SJF \qquad (1)$$

Fig. 3 shows the implementation of net priority scheduling.

TABLE III NET PRIORITY SCHEDULING

Cloudlet ID	VM ID	Time	Start Time	Finish Time
0	0	41.83	0.1	41.93
1	1	68.55	0.1	68.65
3	2	81.292	0.1	81.392
2	3	92.588	0.1	92.688
4	4	100.998	0.1	101.098
6	0	98.546	41.93	140.476
5	1	115.55	68.65	184.194
8	2	110.506	87.8	198.306
7	3	129.59	87.544	217.13
10	4	122.276	99.834	222.11
9	0	139.85	140.476	280.33
12	1	138.72	184.194	322.92
11	2	151.484	198.306	349.79
13	3	150.766	219.122	369.894
14	4	165.122	220.118	385.24
16	0	162.746	280.33	443.076
15	1	180.53	322.92	503.444
18	2	174.196	349.79	523.986
17	3	195.1	370.09	565.184
19	4	185.56	385.044	570.604

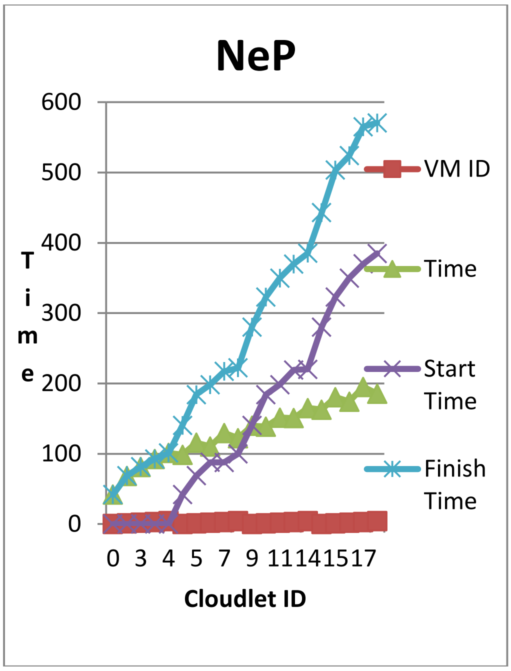

Fig. 3. NeP Scheduling

V. CONCLUSION

Load Balancing in cloud computing environment provides an efficient solution to achieve maximum utilization of resources. Virtualization transforms the data center into a flexible cloud infrastructure with the performance and reliability to run the applications on demand. As virtualization adoption increases across IT domains, they should deploy virtualization with automation across the data center. In this paper we have discussed some load balancing algorithms used in cloud computing. The performance of the system can be increased with efficient load balancing approaches. We used cloudsim tool to implement scheduling algorithms like FCFS and SJF. Here we propose net priority scheduling for scheduling the tasks. When the task scheduling becomes most appropriate and optimal all the virtual machines are fully occupied and task execution rates remains at maximality.

REFERENCES

[1] Google App Engine, 2013. [Online]. Available: http://code.google.com/intl/zh-CN/appengine/

[2] IBM blue cloud, 2013. [Online]. Available: http://www.ibm.com/grid/

[3] MicrosoftWindows Azure]. Available: http://www.microsoft.com/windowsazure

[4] Calheiros, R. N., Ranjan, R., Beloglazov, A., Rose, C. A. F. D. & Buyya, R. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. extended version of a keynote paper: R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. Proceedings of the Conference on High Performance Computing and Simulation (HPCS 2009) (pp. 21-24). IEEE Press, New York, USA, Leipzig, Germany, June, 2009.

[5] A. Singh, M. Korupolu, and D. Mohapatra, “Server-storage virtualization: Integration and load balancing in data centers,” in Proc. ACM/ IEEE Conf. Supercomput., 2008, pp. 1–12.

[6] N. G. Shivaratri, P. Krueger, and M. Singhal, Load distributing for locally distributed systems, Computer,vol. 25, no. 12, pp. 33-44, Dec. 1992.

[7] Rich Lee, Bingchiang Jeng ”Load Balancing Tactics In Cloud” International Conference On Cyber Enabled Distributed Computing And Knowledge Discovery, 2011

[8] A Survey on Open-source Cloud Computing Solutions Patrícia Takako Endo, Glauco Estácio Gonçalves, Judith Kelner.

[9] Sikder Sunbeam Islam, M. Baqer Mollah, M. Imanul Huq, M. Aman Ullah, “Cloud Computing for Future Generation of Computing Technology”, Proceedings of the 2nd IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems, Bangkok, Thailand, May, 2012, pp. 1-6.

[10] A.Singh, M.Korupolu, D.Mohapatra, “Server-storage virtualization:integration and load balancing in data centers,” Proceedings of the ACM/IEEE conference on Supercomputing (SC), pp. 978-1-4244-2835-9/08, 2008.

[11] Y. Zhao, W.Huang, “Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud,” Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC,Seoul, Republic of Korea, pp. 170-175, 2009.

[12] R.Stanojevic, R.Shorten, “Load balancing vs.distributed rate limiting: a unifying framework for cloud control,” Proceedings of IEEE ICC,Dresden, Germany, pp. 1-6, 2009.

[13] J.Hu,J.Gu, G.Sun, T.Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment,” Third International Symposium on Parallel Architectures,Algorithms and Programming (PAAP), pp. 89-96, 2010.

[14] M.Randles, D.Lamb, A.Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,” Proceedings of 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, pp. 551-556, 2010.

[15] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, Availability and load balancing in cloud computing, presented at the 2011 International Conference on Computer and Software Modeling, Singapore, 2011.

[16] Saroj Hiranwal , Dr. K.C. Roy, ”Adaptive Round Robin Scheduling Using Shortest Burst Approach Based On Smart Time Slice” International Journal Of Computer Science And Communication July-December 2011 ,Vol. 2, No. 2 , Pp. 319-323.

[17] Bhathiya Wickremasinghe ,Roderigo N. Calherios “Cloud Analyst: A Cloud-Sim-Based Visual Modeler For Analyzing Cloud Computing Environments And Applications”. Proc Of IEEE International Conference On Advance Information Networking nd Applications, 2010.

[18] Wang, S-C., K-Q. Yan, W-P. Liao and S-S. Wang, “Towards a load balancing in a three-level cloud computing network” in proc. 3rd International Conference on Computer Science and Information Technology (ICCSIT), IEEE, Vol. 1,pp:108-113, July 2010.

[19] R. Buyya, R. N. Calheiros, A. Beloglazov, S. Garg. Clousim: A Framework for modeling and simulation of cloud computing infrastructures and services, the cloud computing and distributed systems laboratory, University of Melbourne, www. cloudbus. org.

BIBLIOGRAPHY OF AUTHORS

	Anitha K L is a research scholar in the Department of Computer Science, Bharathiar University, Coimbatore, India. Anitha K L received post graduate degree in Master of Computer Applications and B.Sc. degree in Computer Science from the University of Kerala. Her research interests include cloud computing security, virtualization, networking and distributed computing.
	Dr. T.R. Gopalakrishnan Nair, a Fellow of Institution of Engineers, has 34 years of experience in professional field spread over Research, Industry and Education. Currently, he is the Rector for Rajarajeswari Group of Institutions in India. He was the RAMCO Endowed Chair in Technology in PM University, KSA. He holds degrees M.Tech. (I.I.Sc., India) and a Ph.D. in Computer Science. His areas of interest include Advanced networks, Cognitive Systems and Multidisciplinary studies including Brain and physical systems. He is a senior member of IEEE, ACM and few other professional bodies.