Simulation of Dynamic Load Balancing Algorithms

Dr.S. Suguna and R. Barani

Abstract--- Cloud computing is a new technology which uses virtual machines instead of physical machines to host, store and network different components. Load balancing is a methodology to distribute workload across multiple computers, or other resources over the network links to achieve optimal resource utilization, minimum data processing time, minimum average response time, and avoid overload. The load balancing is to be done properly to gain maximum throughput and performance of cloud. Here we have considered two load balancing algorithms Equally spread current Execution algorithm and Throttled load balancing algorithm and made performance analysis of these algorithms using cloudanalyst.

In Section I we gave brief introduction about load balancing in cloud computing. In section II we made a survey of Load balancing algorithms. We explained the two dynamic load balancing algorithms in section III. In section IV we introduced a simulation tool cloud Analyst and in next section V we analyzed the performance of algorithms.

Keywords--- Cloud Computing, Load Balancing, Equally Spread Current Execution, Throttled, Cloud Analyst, Performance Analysis

I. INTRODUCTION

CLOUD computing is one of the internet based service provider which allows users to access services on demand [1]. It provides pool of shared resources of information, software, databases and other devices according to the client request on “pay as you go” basis [2].

In cloud computing environment, the random arrival of tasks with random utilization of CPU service time requirements can load a specific resources heavily, while the other resources are idle or less loaded[2]. Hence resource Control or Load balancing is major challenging issue in cloud computing. Load Balancing is a methodology to distribute the workload across multiple computers or other resources over the network links to achieve optimal resource utilization, minimum data processing time and to avoid overload. Load balancing ensures that all the processors in the system as well as in the network do approximately the equal amount of work at any instant of time. Load balancing serves two important needs, primarily to promote availability of cloud resources and secondarily to promote performance. In order to balance the

requests of the resources it is important to recognize a few major goals of Load Balancing algorithms:

- a. Cost Effective- Primary aim is to achieve an overall improvement in system performance at a reasonable cost.
- b. Scalability and Flexibility- The distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.
- c. Priority- Prioritization of the resources or jobs need to be done on beforehand through the algorithm itself for better services to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their provision.[3]

II. SURVEY OF LOAD BALANCING ALGORITHMS

Load balancing [4] algorithm directly influences the effect of balancing the server workloads. Its main task is to decide how to choose the next server node and transfer a new connection request to it. Current main load balancing algorithm is divided into static algorithm and dynamic algorithm [5]. The static algorithm is easily carried into execution and takes less time, which doesn't refer to the states of the load nodes, but it can be only used in certain specific conditions. The common static algorithms are Round-Robin Scheduling Algorithm, Weighted Round-Robin Scheduling Algorithm, and Least-Connection Scheduling Algorithm etc. Of all, Round -Robin Scheduling Algorithm is the simplest one which could be most easily be carried out. However, it is only applicable to the circumstances in which all the nodes in cluster have the same processing ability. The dynamic algorithm like first come first serve is self-adaptive algorithm, which is better than static algorithm, and suitable for a great deal of requests which procreate different workloads, which would be unable to be forecasted [6]. M. Randles et al.,[7] have proposed comparison of static and dynamic load balancing algorithm for cloud computing. . Ram Prasad Pandhy et al.,[8] discussed on basic concepts of Cloud Computing and Load balancing and studied some existing load balancing algorithms, which can be applied to clouds. Dr.Hemantst al.,[9] in his paper described the features of a simulator to compare the performance of three dynamic load balancing algorithms. Cloud Analyst: A Cloud Sim-based Visual Modeler for Analyzing Cloud Computing Environments and Applications. Shu-Ching et al.,[10] suggested Load Balancing in a three-level Cloud Computing network, by using a scheduling algorithm which combines the features of Opportunistic Load Balancing (OLB) and Load Balance Min-Min (LBMM) which can utilize better executing efficiency and maintain load balancing of the system. Mayanka Katya et al.,[11] presented various load balancing schemes in different

Dr.S. Suguna, Assistant Professor, Department of Computer Science, Sri Meenakshi Govt. Arts College for Women(A), Madurai-2, Tamil Nadu, India. E-mail:kt.suguna@gmail.com

R. Barani, Assistant Professor, Department of Computer Science, Sri Sarada Niketan College of Science for Women, Karur-5. Tamil Nadu, India. E-mail:baraniraj77@gmail.com
DOI:10.9756/BIJSESC.8061

cloud environment based on requirements in Service Level Agreement(SLA). NayandeepSran et al.,[12] studied many scheduling algorithms like honeybee foraging algorithm, biased random sampling, active clustering, OLB+LBMM, Min-Min and made comparative study based on their metrics. J.Uma et al.,[13] made a methodological analysis of various load balancing algorithms. N.S.Raghava et al.,[14] analyzed various load balancing algorithms in cloud computing and compared to provide a gist of the latest approaches in the research area.

III. DYNAMIC LOAD BALANCING ALGORITHMS

3.1. Equally Spread Current Execution(ESCE)

Equally Spread current execution is a dynamic load balancing algorithm, which handles the process with priority. It determines the priority by checking the size of the process. This algorithm distributes the load randomly by first checking the size of the process and then transferring the load to a Virtual Machine, which is lightly loaded. The load balancer spreads the load on to different nodes, and hence, it is known as spread spectrum technique. The load balancer maintains a queue of the jobs that need to use and are currently using the services of the virtual machine. The balancer then continuously scans this queue and the list of virtual machines. If there is a VM available that can handle request of the node/client, the VM is allocated to that request [15]. If however there is a VM that is free and there is another VM that needs to be freed of the load, then the balancer distributes some of the tasks of that VM to the free one so as to reduce the overhead of the former VM. The jobs are submitted to the VM manager, the load also maintains a list of the jobs, their size and the resources requested. The balancer selects the job that matches the criteria for execution at the present time. Though there algorithm offers better results as shown in further section, it however requires a lot of computational overhead.

The following figure shows how ESCE works.[16]

Figure 1: ESCE Scheduling Process

3.1.1. Algorithm

1. Find the next available VM
2. Check for all current allocation count is less than max length of VM list allocate the VM
3. If available VM is not allocated create a new one
4. Count the active load on each VM
5. Return the id of those VM which is having least load.
6. The VMLoad Balancer will allocate the request to one of the VM.

7. If a VM is overloaded then the VMLoadBalancer will distribute some of its work to the VM having least work so that every VM is equally loaded.
8. The datacenter controller receives the response to the request sent and then allocates the waiting requests from the job pool/queue to the available VM & so on.
9. Continue from step-2.

3.2. Throttled Load Balancer (TLB)

Throttled load balancer is a dynamic load balancing algorithm. In this algorithm, the client first requests the load balancer to find a suitable Virtual machine to perform the required operation. It is shown in fig.2. The process first starts by maintaining a list of all the VMs each row is individually indexed to speed up the lookup process. If a match is found on the basis of size and availability of the machine, then the load balancer accepts the request of the client and allocates that VM to the client. If, however there is no VM available that matches the criteria then the load balancer returns -1 and the request is queued. The following figure shows how it works [17].

Figure 2: Throttled Scheduling Process

3.2.1. Throttled Algorithm

1. Throttled VM Load Balancer maintains an index table of VMs and the state of the VM (BUSY/AVAILABLE). At the start all VM's are available.
2. Datacenter Controller receives a new request.
3. Datacenter Controller queries the Throttled VM Load Balancer for the next allocation.
4. Throttled VM Load Balancer parses the allocation table from top until the first available VM is found or the table is parsed completely.

If found:

- i. The Throttled VM Load Balancer returns the VM id to the Datacenter Controller.
- ii. The Datacenter Controller sends the request to the VM identified by that id.
- iii. Datacenter Controller notifies the Throttled VM Load Balancer of the new allocation.
- iv. Throttled VM Load Balancer updates the allocation table accordingly.

If not found:

- i. The Throttled VM Load Balancer returns -1.
- ii. The Datacenter Controller queues the request.
5. When the VM finishes processing the request, and the Datacenter Controller receives the response cloudlet, it notifies the Throttled VM Load Balancer of the VM de-allocation.
6. The Datacenter Controller checks if there are any waiting requests in the queue. If there are, it continues from step 3.
7. Continue from step 2.

IV. CLOUD ANALYST

In this paper we propose a tool, called Cloud Analyst, Cloud Analyst [9] [17] [18] is a GUI based tool that is developed on CloudSim architecture. CloudSim is a toolkit that allows doing modeling, simulation and other experimentation. The main problem with CloudSim is that all the work need to be done programmatically. It allows the user to do repeated simulations with slight change in parameters very easily and quickly. The cloud analyst allows setting location of users that are generating the application and also the location of the data centers. In this various configuration parameters can be set like number of users, number of request generated per user per hour, number of virtual machines, number of processors, amount of storage, network bandwidth and other necessary parameters. Based on the parameters the tool computes the simulation result and shows them in graphical form. The result includes response time, processing time, cost etc. By performing various simulations operation the cloud provider can determine the best way to allocate resources, based on request which data center to be selected and can optimize cost for providing services.

Figure 3: Cloud Analyst Architecture

Figure 4: Cloud Analyst GUI

4.1. Simulation Parameters

4.1.1. Region

In the Cloud Analyst the world is divided in to 6 'Regions' that coincide with the 6 main continents in the World. The other main entities such as User Bases and Data Centers belong to one of these regions.

4.1.2. Users

A User Base models a group of users that is considered as a single unit in the simulation and its main responsibility is to generate traffic for the simulation. A single User Base may represent thousands of users but is configured as a single unit and the traffic generated in simultaneous bursts representative of the size of the user base.

4.1.3. Datacenter Controller

The Data Center Controller is probably the most important entity in the CloudAnalyst. A single Data Center Controller is mapped to a single cloudsims. Datacenter object and manages the data center management activities such as VM creation and destruction and does the routing of user requests received from User Bases via the Internet to the VMs. It can also be viewed as the façade used by CloudAnalyst to access the heart of CloudSim toolkit functionality.

4.1.4. Internet Characteristics

In this component various internet characteristics are modeled simulation, which includes the amount of latency and bandwidth need to be assigned between regions, the amount of traffic, and current performance level information for the data centers.

4.1.5. VM Load Balancer

The responsibility of this component is to allocate the load on various data centers according to the request generated by users. One of the four given policies can be selected.

4.1.6. Cloud App Service Broker

The responsibility of this component is to model the service brokers that handle traffic routing between user bases and data centers. The service broker can use one of the routing policies from the given three policies which are closest data center, optimize response time and reconfigure dynamically with load. The closest data center routes the traffic to the closest data center in terms of network latency from the source user base. The reconfigure dynamically with load routing policy works in the sense that whenever the performance of particular data center degrades below a given threshold value then the load of that data center is equally distributed among other data centers.

V. SIMULATION AND RESULT ANALYSIS

In order to analyze various load balancing policies configuration of the various component of the cloud analyst tool need to be done. We have set the parameters for the user base configuration, application deployment configuration, and data center configuration as shown in figures4, 5 and 6. As shown in figure 4 the location of user bases has been defined in six different regions of the world. We have taken three data centers to handle the request of three users. DC1 located in

region 0, DC2 in region 1 and DC3 in region 2. 50 VM are allocated in each Data Center. Thesimulation duration is set as 60hrs. Here we have taken 6 user bases.

Configure Simulation

Simulation Duration: 60 min

User bases:	Name	Region	Requests per User per Hr	Data Size per Request (Bytes)	Peak Hours Start (GMT)	Peak Hours End (GMT)	Avg Peak Users	Avg Off Peak Users
UB1	0	60	100	3	9	1000	100	
UB2	1	60	100	3	9	1000	100	
UB3	2	60	100	3	9	1000	100	
UB4	3	60	100	3	9	1000	100	
UB5	4	60	100	3	9	1000	100	

Service Broker Policy: Optimise Response Time

Data Center	# VMs	Image Size	Memory	OS
DC1	50	10000	512	1000
DC2	50	10000	512	1000
DC3	50	10000	512	1000

Figure 5: Configure Screen in Simulator

Configure Simulation

Data Center Configuration

Name	Region	Arch	OS	VM	Cost per VM \$/Hr	Memory Cost \$/s	Storage Cost \$/s	Data Transfer Cost \$/Gb	Physical HW Units
DC1	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	3
DC2	1	x86	Linux	Xen	0.1	0.05	0.1	0.1	3
DC3	2	x86	Linux	Xen	0.1	0.05	0.1	0.1	3

Figure 6: Data Center Configuration Screen

Configure Simulation

Advanced

User grouping factor in User Bases: (Equivalent to number of simultaneous users from a single user base) 10

Request grouping factor in Data Centers: (Equivalent to number of simultaneous requests a single application server instance can support.) 10

Executable instruction length per request: (bytes) 100

Load balancing policy across VM's in a single Data Center: Equally Spread Current Execution Load

Figure 7: Advanced Configuration Screen

Figure 8: Output Screen of Simulation ESCE with 6UBs

Figure 9: Output Screen of Simulation of TLB with 6UBs

5.1. Response Time

The overall response time and user Base hourly response time for ESCE and TLB are as shown in figure 10, 11, 12 and 13

Overall Response Time Summary

	Average (ms)	Minimum (ms)	Maximum (ms)
Overall Response Time:	158.87	40.18	378.12
Data Center Processing Time:	0.53	0.01	1.19

[Export Results](#)

Response Time By Region

Userbase	Avg (ms)	Min (ms)	Max (ms)
UB1	50.235	40.181	62.434
UB2	50.407	41.695	62.441
UB3	50.052	40.355	61.105
UB4	300.564	231.108	378.116
UB5	298.719	237.06	361.617
UB6	200.434	160.44	245.44

Figure 10: Response Time for ESCE

Figure 11: Hourly Response Time for ESCE

Figure 12: Response Time for TLB

Figure 13: Hourly Response Time for TLB

5.2. Data Center Request Service Time

The data center request service time for ESCE and TLB are shown in figure 14 and 15 respectively.

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.596	0.048	1.18
DC2	0.796	0.054	1.19
DC3	0.388	0.015	0.854

Figure 14: Data Center Request Service Time for ESCE

Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	0.596	0.051	1.18
DC2	0.793	0.054	1.19
DC3	0.388	0.015	0.854

Figure 15: Data Center Request Service Time for TLB

5.3. Processing Cost

The total processing cost for ESCE and TLB are shown in tables 16 and 17 respectively.

Cost

Total Virtual Machine Cost : \$10.54

Total Data Transfer Cost : \$0.38

Grand Total : \$10.92

Data Center	VM Cost	Data Transfer Cost	Total
DC3	0.502	0.192	0.693
DC2	5.018	0.065	5.084
DC1	5.018	0.128	5.146

Figure 16: Total Processing Cost for ESCE

Cost

Total Virtual Machine Cost : \$10.54

Total Data Transfer Cost : \$0.38

Grand Total : \$10.92

Data Center	VM Cost	Data Transfer Cost	Total
DC3	0.502	0.192	0.693
DC2	5.018	0.065	5.084
DC1	5.018	0.128	5.146

Figure 17: Total Processing Cost for TLB

Figure 18: The Analysis Result of Simulation of 3Data Centers with 25, 50 and 75 VMs using ESCE

Figure 18: The Analysis Result of Simulation of 3Data Centers with 25, 50 and 75 VMs using TLB

VI. CONCLUSION

We have simulated two different dynamic load balancing algorithms for executing the user request in cloud environment. Each algorithm is observed and their scheduling criteria like average response time, data center service time and total cost of different data centers are found. Here we have used 6UserBases and 3 DataCenters and compared the performance of algorithms using 25, 50 and 75VMs and result is as shown in graph. Our future work is to develop an adaptive algorithm suitable for heterogeneous environment such that it handles Big Data and improve overall response time with reduced cost.

REFERENCES

- [1] G. Pallis, "Cloud Computing: The New Frontier of Internet computing", IEEE Journal of Internet Computing, Vol. 14, No. 5, September/October 2010, pages 70-73.
- [2] Qi Zhang, Lu Cheng, RaoufBoutaba, "cloud computing: state of-the-art and research challenges", 20th April 2010, Springer, pp. 7-18.
- [3] Sonia Lamba and Dharmendra Kumar, "A Comparative study on

- Load Balancing Algorithms with Different Service Broker Policies in Cloud Computing" International Journal of Computer Science and Information Technologies,(IJCSIT) Vol. 5 (4) , 2014, pp. 5671-5677
- [4] R.Shimonski. Windows 2000 &Windows server 2003 clustering and Load balancing. Emeryville. McGraw-Hill Professional publishing. CA, USA(2003), Pp 2,2003.
- [5] R.X.T. and X.F.Z. "A load balancing strategy based on the Combination of static and dynamic, in database technology and applications (DBTA),2010 2nd international workshops, pp. 1-4, (2010)
- [6] Wenzheng Li,HongyanShi "Dynamic Load Balancing Algorithm Based on FCFS" IEEE, 2009. pp.1528-1531.
- [7] M.Randles,D.Lamb and A.Talen-Bendiab," A comparative study into Distributed load balancing algorithms for cloud computing," 2010, IEEE 24th international conference onadvanced information Networking and application workshops,2010,pp.551-556.
- [8] Ram Prasad Pandhy (107CS046), P Goutam Prasad rao (107CS039). "Load balancing in cloud computing system" Department of computer science and engineering National Institute of Technology Rourkela, Rourkela-769008, Orissa, India May-2011.
- [9] Dr. Hemant, S.Mahalle, Prof. ParagR.Kaveri and Dr.Vinay chavan, "Load Balancing On Cloud Data Centers". International Journal of Advanced Research in Computer Science and Software Engineering, Jan-2013.
- [10] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, Shun-Sheng Wang, "Towards a Load Balancing in a Three-level Cloud Computing Network", 2010 IEEE, pp. 108-113.
- [11] MayankaKatyalandAtul Mishra, "A comparative study of Load Balancing Algorithms in Cloud Computing Environment", International Journal of Distributed and cloud Computing, Volume 1, Issue 2, December 2013.
- [12] NayandeepSran and NavdeepKaur,"Comparative Analysis of Existing Load Balancing Techniques in Cloud Computing", International Journal of Engineering and Science Invention, Volume 2, Issue 1, Pp.60-63, January 2013.
- [13] J. Uma, V. Ramasamy and A. Kaleeswaran, "Load Balancing Algorithms in Cloud Computing Environment-A Methodical Comparison", International Journal of Advanced Research in Computer Engineering and Technology(IJARCET), Volume 3,Issue 3, February 2014.
- [14] N.S.Raghava and Deepti Singh," Comparative Study on Load Balancing Techniques in Cloud Computing", Open Journal of Mobile Computing and Cloud Computing, Volume 1, Number 1, August 2014.
- [15] Tanvee Ahmed, Yogendra Singh "Analytic Study of Load Balancing Techniques using Tool Cloud Analyst". International Journal Of Engineering Research and Applications. Pp.1027-1030, 2012.
- [16] Subasish Mohapatra, K. Smruti Rekha and Subhadarshini Mohanty, "A comparison of our popular Heuristics for Load Balancing of Virtual Machines in Cloud computing", International Journal of Computer Applications (0975 – 8887) Volume 68– No.6, April 2013.
- [17] A. Singh, P. Goyal, S. Batra :Anoptimized round robin scheduling Algorithm for CPU scheduling, International journal of computer and Electrical engineering (IJCEE), vol. 2, No. 7, Pp 2383- 2385, December, 2010.
- [18] Bhathiya, Wickremasinghe" Cloud Analyst: A Cloud Sim-based Visual Modeler for Analyzing Cloud Computing Environments and Applications", 2010, IEEE.

Dr. S. Suguna, working as Assistant Professor in Department of computer Science in Sri Meenakshi Govt. Arts College for Women (A),Madurai , Tamil Nadu,India.

Dr. S. Suguna received the M.C.A degree from Fatima College, Madurai Kamaraj University, Madurai in 1997, and M.Phil. degree in Computer Science from Mother Teresa Women's University, Kodaikanal in 2003, and Ph. D degree in Computer Science from Mother Teresa Women's University, Kodaikanal in 2011. She has a teaching experience of 18 years. Her area of interest includes Data Mining, Web Mining, Text Mining, Cloud Computing, and Image mining. She has presented and published several papers in National and International Conferences, Seminars and Journals.

R. Barani, working as Assistant Professor in Department of computer Science in Sri Sarada Niketan College Of Science for Women, Karur-5, Tamil Nadu, India.

R. Barani received the M.C.A degree from Sri Sarada Niketan College, karur, Bharathidasan University, Trichy in 2001, and M.Phil.Degree in Computer Science from Alagappa University, Karaikudi in 2005, She has a teaching experience of 11 years. She is now doing Ph.D. in Bharathiar University, Coimbatore. Her area of interest includes Cloud Computing, Big Data and Computer Networks. She has presented paper in National and International Conference.