Juxtaposition of Load Balancing Algorithms in Cloud Computing using Cloud Analyst Simulator

Indresh Gangwar
Computer Science & Engineering
Krishna Institute of Engineering & Technology
Ghaziabad, India

Poonam Rana
Computer Science & Engineering
Krishna Institute of Engineering & Technology
Ghaziabad, India

ABSTRACT

Cloud Computing is a term which is generally used in reference to Internet. The whole Internet can be viewed as a cloud. The cloud is quickly changing a worldwide network of computers into the largest single, "virtual" computer in the world. This includes sharing of resources which increases the load on single machine. Thus the overall performance degrades and this problem is named as load balancing. Load Balancing is one of the challenging issues now a day. To overcome from this problem be reassign the load by implementing various algorithms introduced for this problem. In this paper we study and compare various algorithms, techniques used to solve the problem of Load Balancing and Distributed Load Balancing.

Keywords

Load Balancing; Round Robin; Throttled Load Balancing Algorithm; Equally Spread Current Execution Algorithm. Cloud analyst.

1. INTRODUCTION

Cloud computing is most popular technique that is defined as “A dummy for on-demand network access to a common pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort or service provider Interaction” Cloud Computing is specialized scalable entity that delivers different level of services to the third party. These services are Platform as a service(PaaS), Software as a service(SaaS) & Infrastructure as a Service(IaaS) [1, 11]. Cloud Computing involves various distributed technologies to share resources, software information via internet to reduce the cost, increase computation and performance and satisfy the needs of customer. While providing various benefits it has some side effects or it has various issues like security, load management etc. Load Balancing was identified as a major concern to allow Cloud computing to scale up to increasing demands [2].Load balancing performances is shown by using cloud analyst simulator for different algorithms.

1.1 Goal of load balancing

It main aim is to give best performance by balancing or reassigning the load among various resources. It is new technique that facilitates resources by providing even distribution of load, minimum response time and maximum throughput. One of the best example related to load balancing are websites [5]. As while using website users could experience delays, possible long system responses and timeouts due to traffic and overloading. Load balancing resolve this problem. It provides better distribution of the communication traffic so that website availability is conclusively settled.

1.2 Metrics for Load Balancing

- **Throughput**: It is used to compute the no. of tasks for a give time. The performance of any system is improved if throughput is maximised.
- **Fault Tolerance**: It is defined as to recover from failure. The load balancing should be a best fault tolerant method.
- **Migration time**: During execution of a job, for reducing load it is need to migrate the jobs or resources from one node to other nodes and the time required is migration time which should be minimised to improve the performance of the system.
- **Response Time**: It is the time required to response a task for a specific load balancing algorithm in a system. It should minimum for better performance of a system.
- **Scalability**: It is the capability of an algorithm to perform Load balancing in a system for any determined number of nodes. This parameter should be improved.

1.3 Policies of Load Balancing

- **Information policy:** It defined that what information is required and how this information is collected. This is also defined that when this information is collected.
- **Triggering policy:** In this we explain that time period when the load balancing operation is starting to manage the load.
- **Selection policy:** This policy is used to find out the task which transfers from overloaded node to free node.
- **Resource type policy:** It explains all types of resources which are available during the load balancing.
- **Location policy:** This uses all the results of the resource type policy. It is used to find a partner for a server or receiver.

2. LITERATURE REVIEW

Load Balancing is a technique in which workload is distributed over a number of nodes to decrease the traffic and increase the efficiency and throughput of the cloud. There are various parameters which will improve the load balancing if we properly utilizing the resources. Some of these parameters are reliability, waiting time, adaptability, load rejection, stability, response time, turnaround time, throughput, processor thrashing, execution time, resource utilization, fault tolerant, process migration, etc[3].There are many algorithms that are used to distribute the load. Different algorithms have different criteria and environment to improve the performance. Following are the algorithms used for distributing the load among several nodes.

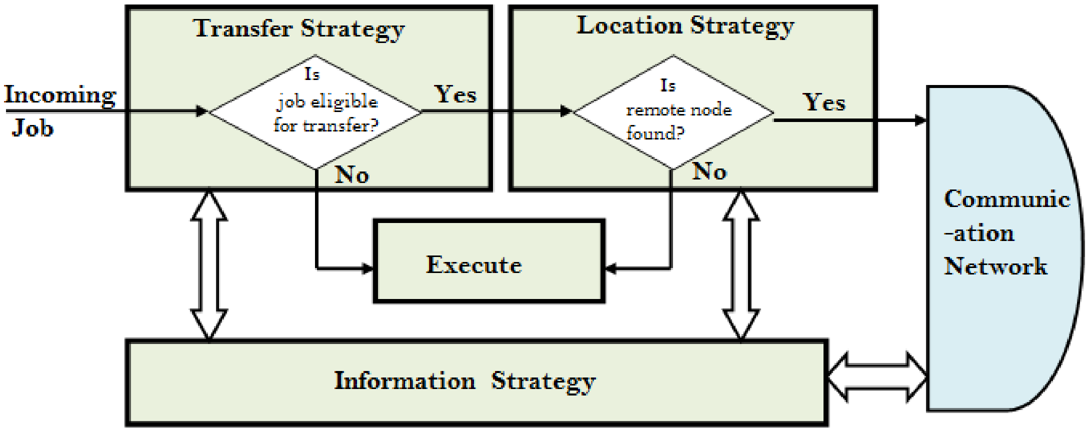


Fig no. 1 Implementation of polices of Load Balancing

On the basis of who initiated the process, load balancing algorithms can be of classified as [3, 4]:

- **Sender Initialized**: Initialized by the sender.
- **Receiver Initiated**: In this load balancing algorithm is initiated by the receiver.
- **Symmetric**: It is hybrid of both sender initiated and receiver initiated.

On the basis of the current state of the system, load balancing algorithms can be classified as [3, 4]:

- **Static:** It doesn't depend on the current state of the system Prior knowledge of the system is needed. It basically evenly distributes the load between servers [12].
- **Dynamic:** Decisions on load balancing are based on current state of the system. Thus it does not require any prior knowledge. Thus it is improved version of static approach. In this we search the lightest load node or server to balance the load [12].

2.1 Distributed Load Balancing

Distributed Load Balancing is used to distribute the work load over multiple networks. For this many distributed solution have been identified. Comparison of different algorithms is shown in table no. 1.

There are different algorithms used for it. Some of them are as follows:

- **Honey Bee Algorithm:** Achieves global load balancing through local serve actions. Its whole concept is based on the idea honey bee, how they search their food and then inform others for the same by waggle dance. The strength or power of waggle dance gives idea about the amount of food present. In the same way the load balancing is done. As virtual machine is overloaded the user request is forwarded to next less loaded virtual machine [5].
- **Round Robin:** In this algorithm the processes are divided between all processors in a round robin order and each processor having equal work load but different processes have different job completion time. Thus some nodes are idle for much of time and some are heavily loaded. Thus it is not a good algorithm [6].
- **Min-Min Algorithm:** First, Minimum completion time for all tasks are found and among these tasks minimum value is chosen and corresponding task is scheduled on corresponding machine. Assigned task is removed from the list of tasks that are assigned to machine. This algorithm has a drawback i.e. Starvation [7, 8].
- **Max-Min Algorithm:** It is similar to min-min algorithm except after finding out minimum execution times and selecting task having the maximum value or having maximum time among all the tasks on any resources[5,8]
- **Biased Random Sampling:** Achieves load balancing across all system nodes using random sampling of the System domain to accomplish self-organization thus stabilize the load among nodes. A virtual graph is constructed to represent the load on serve and connectivity among them [8, 9].
- **Equally Spread Current Execution Algorithm:** It handles processes with Priorities. It distribute the load randomly by checking the size and transfer the load to that virtual machine which is lightly loaded or handle that task easy and take less time , and give maximize throughput. It is also Known as Spread Spectrum technique as the entire load is distributes among nodes by load balancer [7, 8].
- **A Task based Scheduling Algorithm on Load Balancing:** It has two level task scheduling processing. In first level it maps tasks to virtual machines and then virtual machines to host resources. It provides maximum throughput or maximize resource utilization with minimum task response time [5, 10].
- **Throttled Load Balancing Algorithm:** In this algorithm the client first compete to find a suitable Virtual Machine with the help of load balancer to perform the required operation. Virtual machines are designed to handle specific task. The load balancer on the request of user assigns virtual machine on the basis of the requirement of task.
- **Ant Colony Optimization**: This technique take the idea from the ant behaviour, how they collect information and leave the liquid(pheromone) in the path to inform others the about the path of food. This algorithm maintains a pheromone table on the basis of resource utilization and node selection method. The ants searches for overloaded node and then traverse it and then go back to fill the

under loaded node so as to make balance or evenly distribute the load[13]

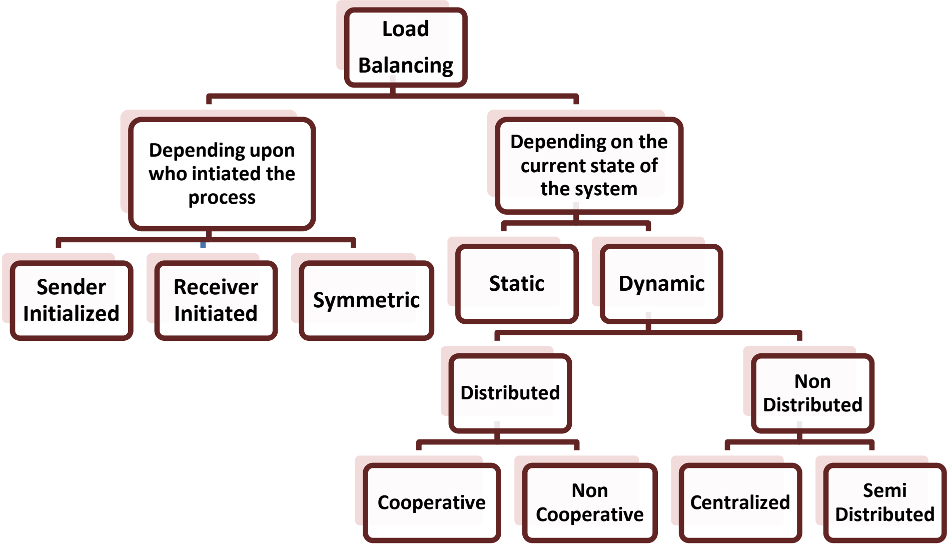


Fig no. 2 Hierarchy of Load Balancing Algorithms

3. SIMULATION TOOL: CLOUD ANALYST

Cloud computing relieved the IT companies from the headache of setting basic hardware and software infrastructure. Thus it also reduces the setting cost of any project or even the setting any IT business. Also they can focus on new technology invention or quality of the service they provide. Some recently emerging application services which are based on cloud technology include content delivery, social networking, real-time instrumented data processing and web hosting. All of these applications have different feature, configuration, and composition and deployment requirements. Thus to check or test the performance of policies like scheduling and allocation in real environment of cloud computing is very difficult and facing many challenges. Because cloud shows dynamic demands, system size, supply patterns and also varying resources. Another challenge is due heterogeneous user requirement of varying quality of service and also dynamic workloads and varying performances. It is really very difficult to use real time application on varying or dynamic requirement and it's also having many benchmarking. Thus it is impossible to perform benchmarking experiments in repeatable, scalable and dependable environments using real-world Cloud environments. To do so we use various simulation tools. One of them is Cloud Analyst .It is a modified version of CloudSim simulator. Cloud analyst is a GUI based simulator which is very easy to use and having ability to define simulations with a higher degree of flexibility and configurability [15]. It provides graphical output and you can easily change the parameters focus on parameters without focusing on programming. You can perform experiments in repeated manner for same parameters or by changing parameters and also save your simulation in a different file [15].

4. EXPERIMENTAL ANALYSIS

We analysis three algorithms Round Robin, Throttled Load Balancing Algorithm and Equally Spread Current Execution Algorithm. We assume some parameters for analysis on cloud analyst.

For Round Robin I Time

- **Overall response time**: Avg(291.61ms) Min(38.26ms) Max(482.02ms).
- **Data Processing time**: Avg(0.18 ms) Min(0.00ms) Max(0.75ms)

For Throttled Load Balancing I time

- **Overall response time**: Avg(291.51ms) Min(40.26ms) Max(482.02ms).
- **Data Processing time**: Avg(0.18 ms) Min(0.00ms) Max(0.75ms).

For Equally Spread Current Execution Algorithm I time

- **Overall response time**: Avg(291.41ms) Min(40.26ms) Max(482.02ms).
- **Data Processing time**: Avg(0.81 ms) Min(0.00ms) Max(0.75ms).

Table 2 contains the assumed parameters and table 3 gives information about user bases. Fig. no. 3 is the output of the round robin algorithm performance.

Table No. 1 Comparison of various Load Balancing Algorithms

ALGORITHM NAME	PARAMETERS USED	TYPE OF ALGORITHM	MERITS	DEMERITS
Honey Bee Algorithms[14]	Throughput, Job completion time, Overhead	Dynamic	Achive Global Load Balancing, Maximize resource utilization, low overhead	Low Priority load
Round Robin[6]	Completion time	Static	Every process get equal weight age so no process will go under starvation.	Most of the time processor remains idle.
Min- Min Algorithm[8]	Execution time	static	Job with smallest execution time is executed.	Starvation
Min-Max Agorithm[8]	Execution time	Static	Improved version of Min-Min algorithm	
Task Scheduling algorithm[5]	Response time, Throughput	Dynamic	Minimize the response time and maximize resource utilization.	
Based Random Sampling[5]	Threshold value or maximum walk length	Dynamic	Stablilize load among nodes	Performance degrades as no. of serve
Equally Spread Current Execution[13]	Response time,	Dynamic	Maximize throughput	Less priorites process need to wait too long.
Throttled Load Balancer[5]	Communication cost,Network Delay,Load Movement Factor	Dynamic	High Load Movement Factor	High communication
Ant Colony Optimization[13]	Fault tolerance, ResourceUtilization,Scalability	Dynamic	High fault tolerance and resource utilization,good scalability	Complex network and need to manage phenome table

5. CONCLUSION

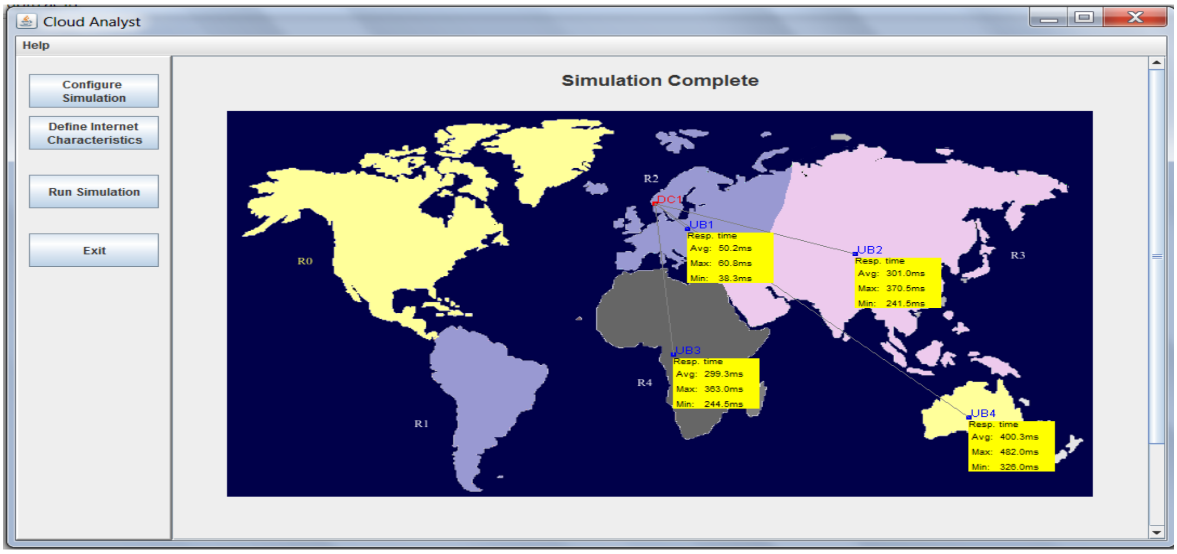
Day-by-day use of cloud computing is increasing which increase the load on the servers providing services to third party. Due to this the overall performance degrades and there is poor resource utilization. This problem is referred as load balancing which is a major issue now days. To solve this problem various algorithms are proposed. In this paper be study the various algorithms, their advantages and disadvantages and the parameters which improve the load balancing problem. Cloud analyst is tool or simulator on which we analyze the performance of various parameters which improves the load balancing problem thus improving overall performance and maximize the resource utilization. Future scope in this field is to find an algorithm and the parameter which best reduces the load balancing problem.

Table No. 2 Factors Assumed For Performance Analysis

Factor assumed	I time	II time
User group factor in user base	1	10
Request grouping factor in data center	1	10
Execute instruction per request	1	10
Simulation time	3 min	3 min

Table No. 3 User bases for simulation

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	2	20	100	3	9	1000	100
UB2	3	40	100	3	9	1000	100
UB3	4	80	100	3	9	1000	100
UB4	5	60	100	3	9	1000	100



Simulation Complete

UB1
Resp. time
Avg: 50.2ms
Max: 60.8ms
Min: 38.3ms

UB2
Resp. time
Avg: 301.0ms
Max: 370.5ms
Min: 241.5ms

UB3
Resp. time
Avg: 299.3ms
Max: 363.0ms
Min: 244.5ms

UB4
Resp. time
Avg: 400.3ms
Max: 482.0ms
Min: 326.0ms

Fig No. 3 Output of round robin algorithm having 4 users and datacenter

6. REFERENCES

[1] Bhaskar Prasad Rimal, Eunmi Choi and Ian Lumb, "A Taxonomy and Survey of Cloud Computing Systems" in 2009 Fifth International Joint Conference on INC, IMS and IDC.

[2] Ian Foster, Yong Zhao, Ioan Raicu and Shiyong Lu, "Cloud Computing and Grid Computing 360-Degree Compared" in Grid Computing Environments Workshop, 2008, IEEE.

[3] Abhijit A. Rajguru and S.S. Apte, "Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters" in International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-1, Issue-3, August 2012.

[4] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing" Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2009-28.

[5] Rajwinder Kaur1 and Pawan Luthra , "Load Balancing in Cloud Computing" in ACEEE, Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC.

[6] Subasish Mohapatra, Subhadarshini Mohanty and K. Smruti Rekha "Analysis of Different Variants in Round RobinAlgorithms for Load Balancing in Cloud Computing" in International Journal Of Computer Applications(0975-8887) Volume 69 – No. 22, May 2013.

[7] Amandeep Kaur Sidhu, Supriya Kinger ,"Analysis of Load Balancing Techniques in Cloud Computing" in International Journal of Computers & Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061.

[8] Shanti Swaroop Moharana , Rajadeepan D. Ramesh & Digamber Powar, "Analysis of Load Balancers in Cloud Computing" in International Journal of Computer Science and Engineering (IJCSE), ISSN 2278-9960 Vol. 2, Issue 2, May 2013, 101-108 ©IASE.

[9] Sheeja S. Manakattu ad S.D. Madhu kumar,"An improved biased random sampling algorithm for load balancing in cloud based systems" in ACM ICACCI '12 Proceedings of the International Conference on Advances in Computing, Communications and Informatics.

[10] Ruhi Gupta, "Review on Existing Load Balancing Techniques of Cloud Computing" in International Journal of Advanced Research in Computer Science and Software Engineering.

[11] M. Armbrust, et al., A view of cloud computing. Commun. ACM. vol. 53 (2010), pp. 50-58.

[12] Zenon Chaczko, Venkatesh Mahadevan , Shahrzad Aslanzadeh and Christopher Mcdermid, “Availability and Load Balancing in Cloud Computing” in 2011 International Conference on Computer and Software Modeling IPCSIT vol.14 (2011) © (2011) IACSIT Press, Singapore.

[13] M Dorigo, G.D. Caro, and L.M. Gambardella, "Ant algorithms for discrete optimization," Artif. Life, vol.5, no.2, pp.137-172, 1999.

[14] Dhineshhh Babu L.D and P. Venkata Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments” in Applied Soft Computing Volume 13, Issue 5, May 2013, Pages 2292-2303

[15] Bhathiya Wickremasinghe,“CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments” in 433-659 DISTRIBUTED COMPUTING PROJECT, CSSE DEPT., UNIVERSITY OF MELBOURNE.