Load Balancing in Cloud using Hybrid Approach

Manmohan Sharma, V.K. Jain

Abstract: ***Cloud computing is up and coming era of registering what's more, a creating processing worldview in the present-day industry, either might be government associations or people in general associations. In basic terms we will outline Cloud computing is assortment of assorted servers that take into consideration want of numerous customers in lightweight of their demands. Clouds have very capable knowledge centers to handle expansive variety of client's demands. Cloud is defined as a shared reservoir of dynamic resources and virtualization. Load Balancing is needed to cater the needs of customer in an efficient manner. In LB the workload is distributed between numerous virtual machines (VM's) on a Server over the system, to accomplish ideal resource utilization, diminish data processing time, diminish in normal response time, and stay away from over-burden. The target of this paper is to propose effective and productive furthermore, upgraded composite scheduling algorithm that can be used to keep up the load and gives proficient resource distribution procedures. This paper outlines the advantages of combining Equally Spread Current Execution (ESCE) and Priority algorithms. The entire simulation is performed on Cloudsim 3.0 toolbox which is JAVA based simulation.***

Index Terms: Cloud Computing, Load balancing (LB), Virtual Machine (VM), CloudSim, and Priority

I. INTRODUCTION

The cloud computing is the innovation utilized over the Web. Cloud computing depends on the distributed computing giving versatile, hardware, software over the Internet. Cloud offers resources, for example, storage, network, and so on. It is the accumulation of servers appropriated over the all-regions of the world which give distinctive administrations on request. In the cloud registering, actual processing is finished by the remote servers. A cloud can be a public, private, or hybrid sort. These clouds give distinctive services to the end-clients on the request. The most basic administrations are Software as a Service (Saas), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1]. The private clouds are claimed by an association they have limited foundation and administrations they are secured from the outside world. While the general population clouds offer extensive scale foundation and administrations to the customer on the payment basis. Access to the public cloud is boundless it can be gotten to from any piece of the world; public clouds are highly inclined to security control. The third sort of cloud is Hybrid cloud they are framed by the blend of public and private clouds. Hybrid cloud gives adaptability in exchange offs between the cost-reserve funds and versatility of the public cloud and control of information as-sets at a private cloud. Numerous merchants offer cloud services, for example, Amazon, AT and T, Eucalyptus System, Google, Microsoft and some more [1][2] Cloud. The primary goal of cloud computing are to share hardware resources, software resources, data resources through web to decreased general cost of the framework, to give better execution as far as normal response time and normal processing time, keep up the framework consistency and to give future adjustment in the system [3].Basically there are diverse difficulties that should be dealt with like rapid demand changes, throughput, fault tolerance, but the main issue is LB.LB is the procedure of conveying the load among different nodes of a conveyed framework with a specific end goal to limit the correspondence delay and to limit the resource usage and furthermore avoiding a circumstance when a some servers are overloaded, some are under loaded while remaining are sitting idle.

II. LOAD BALANCING

LB is the method of dissemination of workload among nodes in an appropriate way with the end goal that it guarantees no node in the framework is over-burden or sits idle. Cloud LB diminishes costs and augments resource accessibility. An effective LB algorithm will ensure that each node in the framework accomplishes pretty much same volume of work. LB is an essential part since we can't anticipate the quantity of requests that are issued at each second in the cloud environment. The reason for LB in the cloud computing is in apportioning the load progressively among the nodes with a specific end goal to fulfill the client prerequisites and to give throughput by declaring the general accessible load to particular nodes [4][5][6]. Different LB algorithms are studied and then got the insight to combine few algorithms features to generate good algorithm for VM LB. In this paper we clubbed the features of two algorithms that are priority and ESCE [5][7].LB works on the principal of moving the load from lightly heavily loaded VM's of lightly loaded or no loaded VM's to make the system more stable and efficient.

III. EXISTING ALGORITHM

To distribute workload among different VM's efficiently in terms of throughput and latency we use:

A. Equally Spread Current Execution Algorithm (ESCE)

The LB balancer makes an attempt to safeguard equivalent load to any or all the VM's connected with the data centers (DC).

Revised Manuscript Received on December 22, 2018.

Manmohan Sharma, CSE, SET, Mody University of Science & Technology, Lakshmangarh, Sikar, Rajasthan, India

V. K. Jain, CSE, SET, Mody University of Science & Technology, Lakshmangarh, Sikar, Rajasthan, India

It appropriates the heap to VM by checking the heap at current time and trade of the heap to that VM which is less stacked and result in less time taken, and gives high throughput. This Load balancer keeps up a record table of VM's and furthermore number of solicitations starting at now distributed to the VM [8]. Exactly when all the VM's are stacked and if the request originates from the DC to assign the new VM, it checks the run-down table for least stacked VM. The DC gives the request to the VM perceived. Right when VM completes the errand, it is educated to DC which is furthermore told by the load balancer.

B. Priority Algorithm

In priority algorithms some priority is decided on which the tasks are executed. In our approach we have used double priority [9]. First priority is on cloudlets length that is the cloudlet with the smallest length will be given highest priority. Second, is on VM's that are the VM's will be sorted on the basis of their MIPS. The cloudlet with the smallest length will be allocated to the VM with lowest MIPS. In this way the overall problem of starvation can also be avoided and response time can also be minimized as cloudlet with large length will be assigned VM with high priority. The prescribed calculation will endeavor to support the execution by giving the assets on ask for, which may achieve increase in number of assignment executions and subsequently decreasing the dismissal in the amount of employments submitted. One of the estimations is picked by stack balancer is illustrated in Fig.1.

IV. PROPOSED WORK

The proposed algorithm is a combination of two algorithms i.e. Double priority [9, 13] and ESCE [12]. Firstly, a decision tree is made indicating status of VM. Then priority algorithm is used when a series of client request are received. The re-quests are then broken into cloudlets and their length is calculated. Then the cloudlets are sorted on the basis of their length. Again, VM's are also sorted on the basis of their MIPS and are allocated to cloudlets in first come first serve (FCFS) order such that cloudlet with smallest length receives VM with least MIPS. Using this technique, the problem of starvation is avoided and total time taken to execute the cloudlet will also be minimized. Now we use the ESCE algorithm to distribute the load equally among all the servers so that no server is under loaded, overloaded or sitting idle. The whole scenario is demonstrated in figure 2.

Step 1: Initially set the status of all VM's to available and the decision tree has no entry.
Step 2: Again, the data center controller will create VM's on the basis of their MIPS and will store them in the same way.
Step 3: The data center controller receives a new task.
Step 4: It will break the task into cloudlets and will calculate their length.
Step 5: Now the data center will sort the cloudlets on the basis of their length and will assign priority.
Step 6: Now allocate the sorted cloudlets to the list of VM's both in FCFS order.
Step 7: The load balancer will check if there is any VM which is under loaded, overloaded or sitting idle.
Step 8: If found, then load will be transferred from overloaded and to under loaded or sitting idle.
Step 9: Update the decision table.
Step 10: Finish the execution.

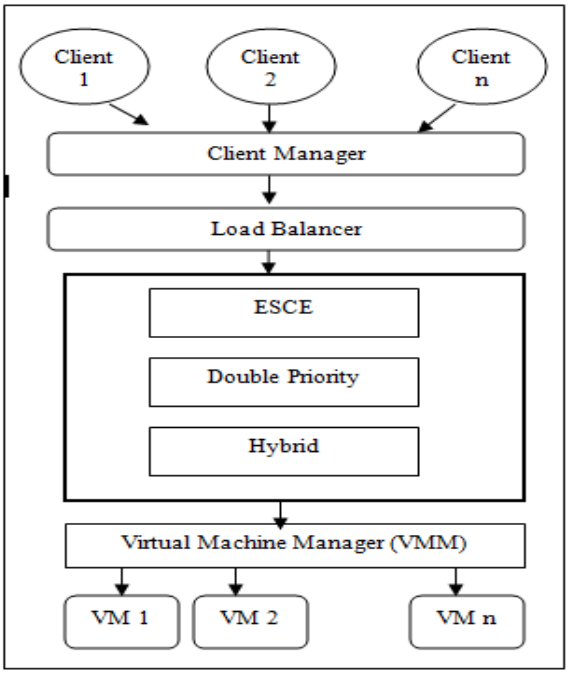

Figure 1: Load balancing algorithms

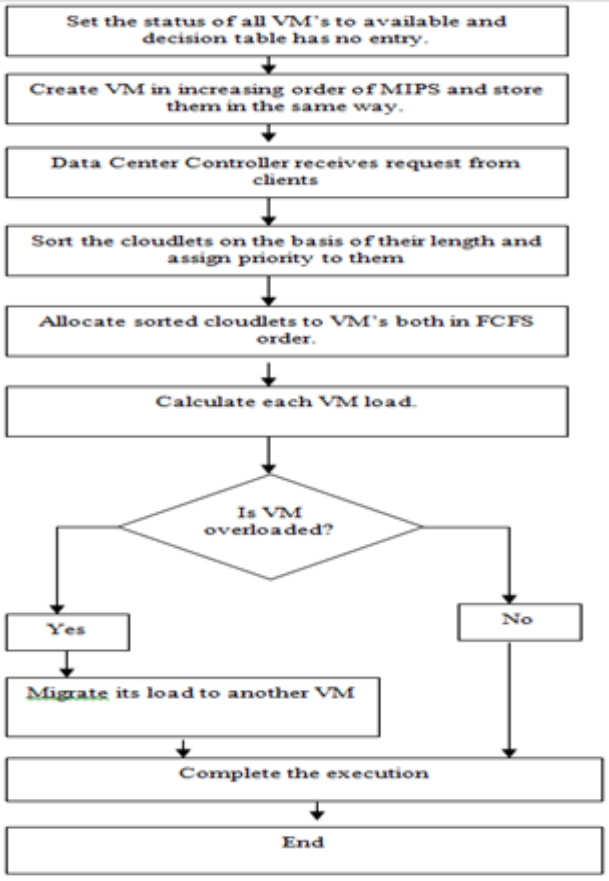

Figure 2: Flowchart of algorithm.

V. SIMULATION ENVIRONMENT

Simulation remains for making a domain which looks and carries on like unique one. The proposed procedure or algorithm can be examined in a simulation environment.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

By utilizing the effectiveness, execution and so forth can be examined. It is more profitable to clients since they can watch their algorithm before executing it on real condition. Additionally, it decreases general cost as changes can be made before acknowledging it really.

A. *CloudSim*

CloudSim objective is to give a reproduction situation that engages showing, recreation, and experimentation of creating Cloud application, empowering its customers to eye on distinct structure setup aftereffect that they have to inspect, without getting stressed over the lowered purposes of intrigue related to Cloud-based establishments and organizations. Remembering the true objective to look at changed load adjusting calculations outline of the distinctive sections ought to be set [10]. We can design our own dataset for different customer and different servers. By using CloudSim, researchers and industry-construct specialists can amass in light of specific structure design issues that they have to investigate, without getting stressed over the low level purposes of premium related to Cloud-based systems and organizations

Distinctive functionalities of CloudSim are:

1. Support simulation of huge scale cloud computing data centers.
2. Support simulation and demonstrating of virtual server host.
3. Support simulation and demonstrating of computational resources.
4. Support simulation and demonstrating of datacenters.
5. Support for user defined scheduling policies.

B. *Eclipse*

Eclipse is a Integrated Development Environment (IDE) for growing principally with Java [2]. It contains workspace where programmer can do his programming. Mostly it is composed in Java and can run on Windows, Linux, and many other platforms having the support of JVM. In Eclipse Platform user can design an application using the concepts of modules. Applications developed using Eclipse Platform (counting the Eclipse IDE itself) can also be re-used by other developers [11].It also provide a good sup-port for additional plugins.

VI. RESULTS

The whole scenario is implemented in CloudSim 3.0 After implementing the proposed hybrid algorithm the following results are achieved.

From figure 3 it is clearly seen that the MIPS of all VM are produced randomly but in such a way that VM 0 will have the least MIPS while the last VM will have the maximum MIPS. Problem of starvation can be avoided using above approach. In figure 6, it is clearly demonstrated that no VM is under loaded, overloaded or sitting idle. All VM's are equally utilized to approximately 73%. Hence the proposed hybrid algorithm handles the problem of LB. In figure 5, it is clearly demonstrated that no VM is under loaded, overloaded or sitting idle. All VM's are equally utilized to approximately 73%.

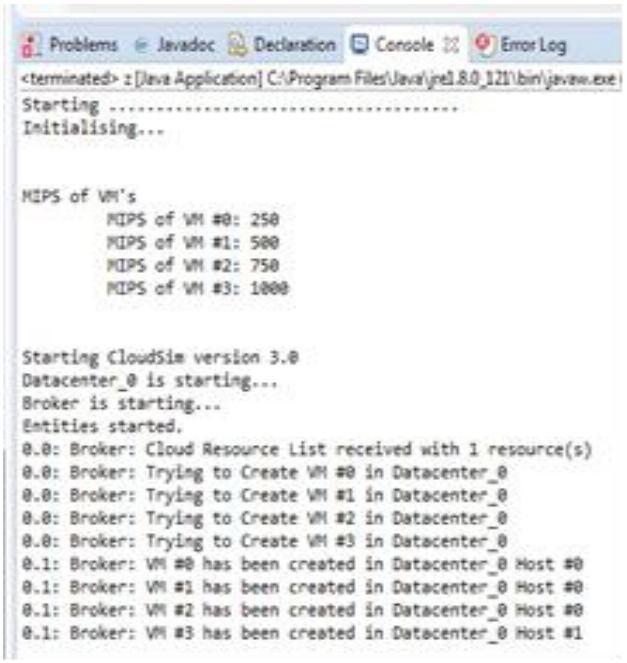

Figure 3: VM's in increasing order of MIPS.

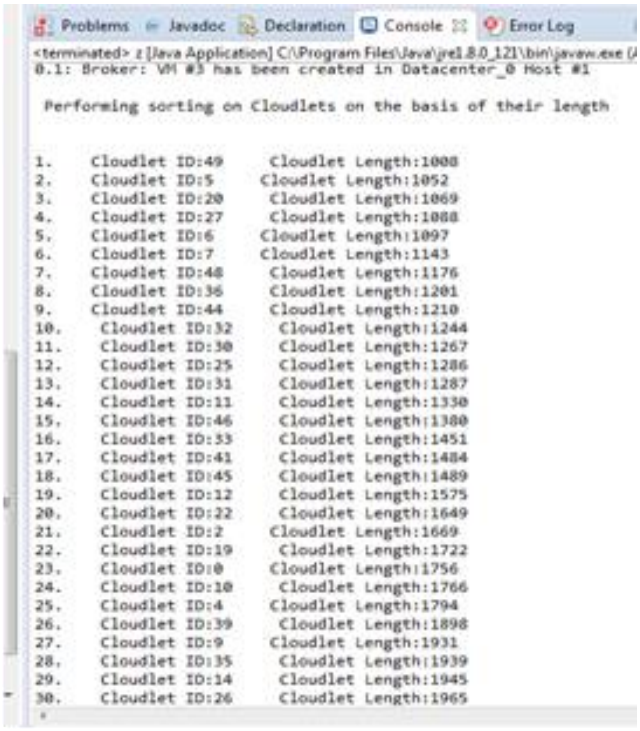

Figure 4: Performing sorting on cloudlets

Hence the proposed hybrid algorithm handles the problem of load LB. In figure 6, the output of the proposed algorithm is demonstrated and it clearly seen that the total time taken to complete the task is reduced to a large amount. Now results of hybrid algorithms are compared with other two algorithms with same set of inputs shown in table 1.Those results are plotted in figure 7 and 8.It is concluded that overall response time of hybrid algorithms is improved about 30 to 40 percent as compared with other algorithms like ESCE and priority.

Figure 5: All VM's are equally utilized

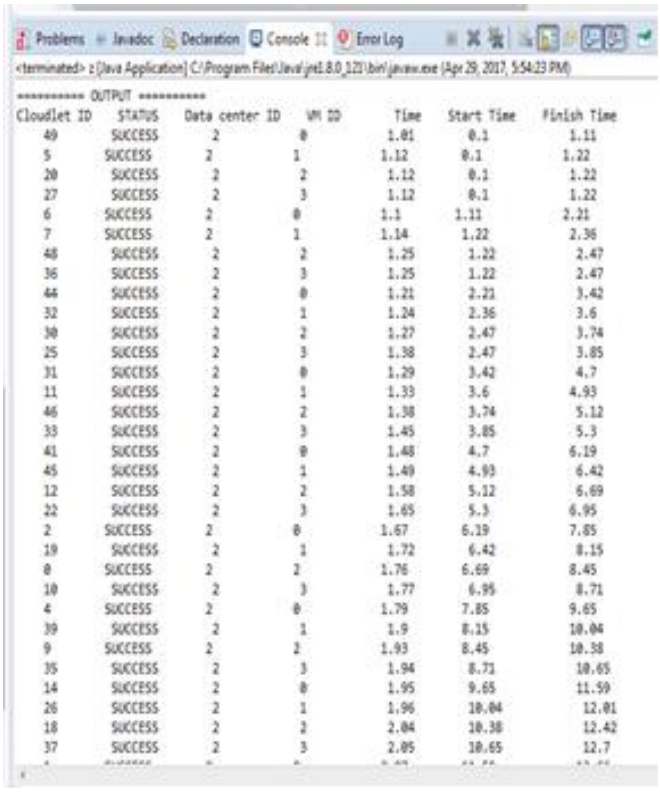

Figure 6: Output of Execution.

Table 1: Results comparison in terms of response time.

Cloudlet ID in sorted order of their length	Cloudlet size (kilobytes)	Overall Response time (seconds)		
		ESCE Algorithm	Priority Algorithm	Hybrid Algorithm
49	1008	1.64	1.56	1.10
5	1052	1.76	1.60	1.22
20	1069	1.79	1.63	1.22
27	1088	1.82	1.71	1.22
6	1097	1.88	1.78	1.22
7	1143	3.66	3.22	2.36

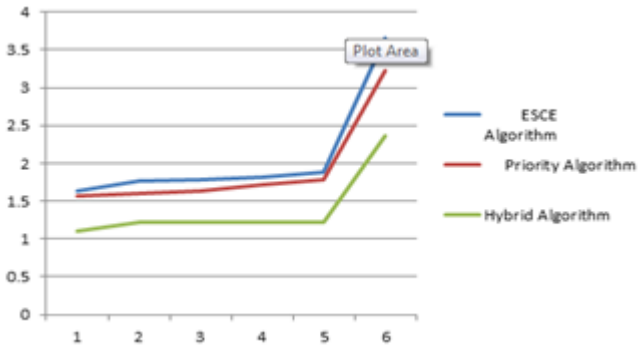

Figure 7: Results comparison line graph

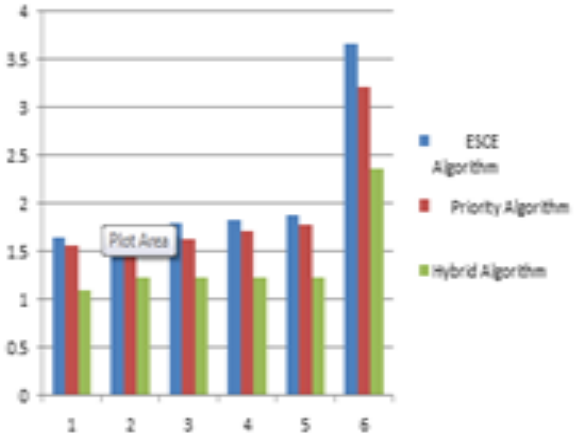

Figure 8: Results comparison bar graph

VII. CONCLUSION

In this paper a hybrid approach is proposed using double priority and equally spread current execution algorithms and then implemented in CloudSim 3.0 environment. The algorithm proposed takes advantages of both the above-mentioned algorithms. Here overall response time and finish time are considered as evaluation parameters. From the results it can be clearly observed that overall response time and processing time is less for hybrid approach as com-pared to others.

In future some more parameters can also be taken for evaluation and priority can be assigned on some other factors of cloudlets such as total cost to execute a cloudlet. In our approach we have considered MIPS of VM's as an important factor but in future some other factors like bandwidth, memory can also be taken into account.

REFERENCES

1. Ranjan Dinesh, Canino Anthony, Izaguirre A Jesus and Douglas Thain "Converting a High-Performance Application to an Elastic Cloud Application" 3rd IEEE International Conference on Cloud Computing Technology and Science, Nov 11.
2. A. Y. Zomaya,& Y. H. Teh. (2014). Observations on using genetic algorithms for dynamic load-balancing. IEEE Transaction on Parallel and Distributed Systems, vol. 12, no. 9, pp. 899-911.
3. Buyya, Rajkumar.,Broberg, James., Goscinski, Andrzej. "Cloud Computing Principles and Paradigms" (1sted.). Hoboken, New Jersey, USA: Wiley, 2011.
4. R. N. Calheiros and R. Buyya, "Meeting deadlines of scientific workflows in public clouds with tasks replication," IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 7, pp. 1787–1796, 2014.
5. Eddy Caron , Luis Rodero-Merino ―Auto-Scaling , Load Balancing And Monitoring In Commercial And OpenSource Clouds ― Research Report, January2012.
6. Mishra ,Ratan , Jaiswal, Anant,P―Ant Colony Optimization: A Solution Of Load Balancing In Cloud‖,April 2012, International Journal Of Web & Semantic Technology;Apr2012, Vol. 3 Issue 2, P33.
7. R. Basker, V. RhymendUthariaraj, and D. Chitra Devi, "An enhanced scheduling in weighted round robin for the cloud infrastructure services," International Journal of Recent Advance in Engineering & Technology, vol. 2, no. 3, pp. 81–86, 2014.
8. BhathiyaWickremasinghe ,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 And Applications ,2010.
9. GenaudStephane and GossaJulien "Cost-wait Tradeoffs in Client-side Resource Provisioning with Elastic Clouds", IEEE 4th International Conference on Cloud Computing, 2011.
10. R.N. Calheiros, R. Ranjan, A. Beloglazov, C. Rose, R. Buyya, "Cloudsim:A toolkit for modeling and simulation of cloud computing environ- ments and evaluation of resource provisioning algorithms",inSoftware:PracticeandExperience(SPE),Vol:41,No:1,ISSN:00380644,Wiley Press,USA,pp:23-50,2011.
11. Buyya R, Ranjan R, Calheiros R N. "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), Leipzig, Germany. IEEE Press: New York, U.S.A., 21–24 June 2009; 1–11.
12. Dr.S. Suguna and R. Barani," Simulation of Dynamic Load Balancing Algorithms", Bonfring International Journal of Software Engineering and Soft Computing, Vol. 5, No.1, July 2015.
13. Ishwari Singh Rajput, Deepa Gupta, "A Priority Based Round Robin CPU Scheduling Algorithm for Real Time Systems", IJIET, vol. 1, no. 3, October 2012, ISSN 2319-1058.