JCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.-6, Issue-11, Nov 2018 E-ISSN: 2347-2693

Load balancing in cloud using prioritization based on Quality of Services (QoS) demand

Manmohan Sharma[1*], Vinod Kumar Jain[2]

[1] Department of Computer Science, Mody University, Lakshmangarh
[2]Department,of Computer Science, Mody University, Lakshmangarh

Available online at: www.ijcseonline.org

Accepted: 25/Nov/2018, Published: 30/Nov/2018

***Abstract*—** Cloud computing is the emerging technology that delivers the services on demand using the underline internet based software's. It is used to store a large amount of data and provide the accessibility to that data with minimum response time. Minimum bandwidth is required to assess the stored data from anywhere. In cloud there is a shared repository of resources that the cloud service providers try to utilize efficiently. Cloud vendors lease these services to the cloud customers as per their needs or demands. Customers also pay to the vendors as per their usage of cloud services. It is beneficial for both the service provider and customer as service provider earn money leasing their resources and customers also earn using the resources which otherwise they need to purchase. Cloud computing generally uses the concepts of virtual machines (VM) and their scheduling to feed maximum customers. This paper briefs about how to improve the quality of the services provided to the customer. Here we assign priorities to the customers on the basis of certain factors like number of VMs required, request type based on pricing and association with company etc. On the basis of these factors we assign priorities to the customers and the customers are feed on the basis of this priority. Here we also included the concepts of feeding the users request from the same geographical areas with minimum distance. This will improve the overall performance and throughput of the cloud with maximum customer satisfaction.

***Keywords*—** Cloud Computing, Load balancing, Priority , Laod balancer

I. INTRODUCTION

Cloud computing is an exceptionally current theme and the term has picked up a great deal of consideration as of late. It can be characterized as on request pay according to utilize display in which shared assets, data, programming and different gadgets are given by the customers' prerequisite when required. Human reliance on cloud is clear from the way that today's most well-known social organizing, email, archive sharing and web based gaming destinations are facilitated on cloud. Google, Microsoft, IBM, Amazon, Yahoo and Apple among others are exceptionally dynamic in this field [1].

Load balancing is the significant zone of worry in cloud computing. It is a component that disperses the workload equitably over every one of the hubs in the entire cloud to maintain a strategic distance from a circumstance where a few hubs are vigorously stacked while others are sit still or doing little work. Load balancing is the way toward reassigning the aggregate burdens to the individual hubs of the aggregate framework to make the best reaction time and furthermore great usage of the assets. Cloud computing is an administration giving office over the web in which the heap adjusting is the one of the testing undertaking. Different strategies are to be utilized to improve a framework by distributing the loads to the hubs in an adjusting way however because of system blockage, transmission capacity use and so forth, there were issues are happened [1][2]. These issues were tackled by a portion of the current algorithms. A load balancing algorithm which is alterable in nature does not require any past data conduct of the framework, that is, it relies on upon the present conduct of the framework. There are different objectives that identified with the load balancing, for example, to enhance the execution significantly, to keep up the framework steadiness and so on. Contingent upon the present condition of the framework, load balancing calculations can be arranged into two sorts they are static and dynamic algorithms.

II. LOAD BALANCING

It disseminates the aggregate load to the individual machines of the framework in such a way that each and every hub successfully uses the assets and limits the reaction time.

It tackles situations where a portion of the hubs are over stacked or under stacked. A dynamic calculation does not consider the past state & relies upon the present conduct of the framework [3].The key qualities utilized are: execution of framework, correlation of load, security of various framework, collaboration between the hubs, estimation of load, choosing of hubs, nature of work to be exchanged and numerous different ones .Load balancing can be categories into basic two types:

A. Static load balancing:

Here the cloud supplier uses homogeneous assets. Here progressed provisioning is done in which the suppliers set up the suitable assets before beginning of administrations as per the agreements done. Non versatile assets are used due to the static condition. The cloud requires prior information of hubs limit, preparing power, memory, execution and insights of client necessities which are not variable [4].

B. Dynamic load balancing:

Heterogeneous assets are supplied by cloud here. Dynamic provisioning is done with adaptable assets in which the specialist co-op apportions more assets as they are expected to the client and expelled them when they are not utilized. Cloud does not require any previous learning [5] and the prerequisites of the clients are versatile. Dynamic condition is hard to be reproduced yet is very customizable with distributed computing condition.

III. METHODOLOGY

It uses the priority which is assigned to both user tasks and datacenters depending on parameters stated. 5.1 Prioritization on user: After submitting the tasks, it will be processed by LB which calculates the priority. The user task will provide parameters like cost based on QoS request (CBQR) provided 60% priority which satisfies SLA conditions which aims to increase the overall efficiency [3][8]. More the amount of CBQR, high the priority it will be assigned. Second parameter will be the count of VMs required which is assigned 40% priority. Less priority will be assigned to the task which will require more number of VM. Now the load balancer will calculate the total priority of the task using table 1 and 2.

5.2 Prioritization on datacenters: Again the prioritization to datacenters will be assigned by the load balancers on the basis of one parameter. More number of VMs a datacenter possess, higher priority it will be assigned. The table of the datacenters priority (table 3) will be available to load balancer [4]. On the basis of this the load balancer will prepare table of datacenters on the basis of their geographical location (GL) and the count of VMs they have and also the number of available VMs in a datacenter.

5.3 Load balancer: It is the middleware which is responsible for assigning the priority to user tasks and datacenters. It is the heart of the algorithm & contains table on CBQR which describes the range of priority.

CBQR		
Quality	Cost Range	Priority
Excellent	CBQR>1000000	60%
Good	50000<CBQR<=1000000	30%
Average	10000<CBQR<=50000	10%

Table 1: Priority assignment based on CBQR

It also allocates priority to a parameter that is VMs count required. More the number of VM required less will be the priority as depicted in table 2. Also the load balancer will manage a table of priority of datacenters. The priority will be assigned to datacenters on the basis of VMC. More the VMC a datacenter had more is the priority. This is because higher VMC will lead in increase in overall efficiency of that datacenter and thus more priority.

Virtual Machine count(VMC)	
Total VMs required	Priority
1-3	40%
4-6	30%
7-10	20%
>10	10%

Table 2: Priority assignment based on VMs

Datacenter	
Total VMs	Priority
1-3	10%
4-6	20%
7-10	30%
>10	40%

Table 3: Priority assignment to Datacenter.

5.4 Priority calculation: Here we have considered 4 users U1, U2, U3 and U4 whose parameters are stated in table 3. The load balancer will calculate the priority of each user task by using table 1 and 2.

Priority is calculated as follows. Let us consider U1. VMC required here is 6 and by table 2 we can see that the priority for VMC=6 is 30%. Again CBQR is 1000001 and by looking in table 1 we can see that when CBQR>1000000 priority is 60%. Hence total priority to U1 is 90%. Similarly, priorities of all users can be calculated as demonstrated in table 3.

Priority table of user tasks				
UID	VMC	CBQR	Priority	GL
U1	6	1000001	90%	Asia
U2	8	60000	60%	Africa
U3	4	40000	30%	America
U4	7	20000	40%	Asia

Table 4: Priority table of users

Data Center in Asia Region			
DC ID	VMC	VMC available	Priority
DC 1	10	10	40%
DC2	8	8	30%
Dc3	7	7	30%

Table 5: Data center in Asia Region

Data Center in Africa Region			
DC ID	VMC	VMC available	Priority
DC4	9	9	30%
DC5	7	7	30%

Table 6: Data center in Africa Region

Data centre in America Region			
DC ID	VMC	VMC available	Priority
DC 6	4	4	20%
DC7	5	5	20%
Dc8	2	2	10%

Table 7: Data center in America Region

5.5 Datacenter allocation process: The LB now has table of user tasks & datacenter table along with their priority and GL, VMC, priority. The LB will look for the maximum priority task [6][7]. Next it will check mapping. It checks the GL of the task and searches for the cluster of datacenter of that location or nearby location in order to minimize response time. Load balancer will check datacentres according to range of priorities of table 2. Once it founds a datacenter with VMs which satisfies the task need, it will check number of VMs available at that time. If count of VMs present are equal to VMs needed, task will be allocated to that VM otherwise another datacenter is searched. In our example the LB will first process task of U1. It will check its GL which is Asia. Then it will search for the cluster of datacenter that have GL= Asia or nearby Asia. Now the LB will check datacentres according to their range of priorities & will apply best fit to the task [9][10].

5.6 Waiting list

1. Task left idle will be placed in the waiting queue according to their GL and decreasing priorities [5].
2. This waiting list is handled by the load balancer.
3. When execution gets over, it deallocates the VMs and the status is updated in the DC table of that particular GL.
4. Request with higher priority in the waiting list is checked & executed else next task in the waiting list is considered.

Proposed Algorithm

Phase 1:
In first phase all VMs in the datacenter are available.

Step 1: Let there be N set of users {U1, U2, Un}.
Step 2: The load balancer will maintain 3 priority assignment tables that is table 1, 2 and 3.
Step 3: Load balancer receives task.
Step 4: Load balancer will calculate the priority of the user tasks using table 1 and 2.
Step 5: LB will maintain a table of datacenters by making cluster on the basis of their GL.
Step 6: The LB will be possessing information about the VMCs in the datacenter and the geographical location of that datacenter.
Step 7: Load balancer will pick up the task with high priority and will check its GL.

Step 8: Load balancer will look for cluster of datacenter with that particular GL.
Step 9: After finding cluster of datacenters with that particular GL load will be assigned to datacenter with the minimum number of VMs by applying best fit.
Step 10: Update the status of virtual machines available in the datacenters table (table 7, 6, 5).
Step 11: Begin the execution.
Step 12: Deallocates the VM once the execution is completed and update the status in datacenter table.

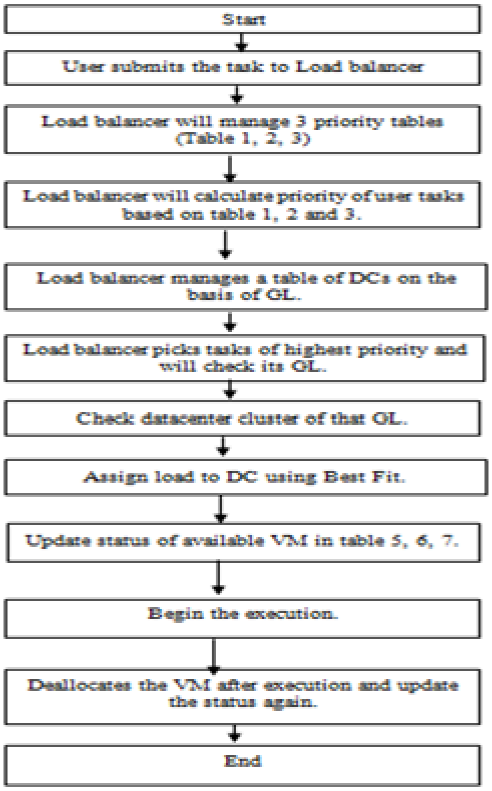
Start
User submits the task to Load balancer
Load balancer will manage 3 priority tables (Table 1, 2, 3)
Load balancer will calculate priority of user tasks based on table 1, 2 and 3.
Load balancer manages a table of DCs on the basis of GL.
Load balancer picks tasks of highest priority and will check its GL.
Check datacenter cluster of that GL.
Assign load to DC using Best Fit.
Update status of available VM in table 5, 6, 7.
Begin the execution.
Deallocates the VM after execution and update the status again.
End

Figure 1 Flowchart of phase 1

Phase 2:
VMs in the datacenter are not available.

Step 1: Let there be N set of users {U1, U2… Un}.
Step 2: The load balancer will maintain 3 priority assignment tables i.e. table 1, 2 and 3.
Step 3: Load balancer receives task.
Step 4: Load balancer will calculate the priority of the user tasks using table 1 and 2.
Step 5: LB maintains table of datacenters by making cluster depending on GL and will update VMC.
Step 6: The load balancer will be possessing information about the Virtual Machine Counts in the datacenter, available virtual machines and the geographical location of that datacenter.
Step 7: Load balancer will pick up the task with high priority and will check its GL.
Step 8: Load balancer will look for cluster of datacenter with that particular GL.
Step 9: After finding cluster of datacenters with that particular GL, load balancer will look for datacenter with the nearest Virtual Machine Count required. and will also check the available Virtual Machine Counts with datacenter.
Step 10: If it satisfies the need of that task, it is allocated to the datacenter. Else repeat step 9 to 10.
Step 11: Update the status of virtual machines available in the datacenters table (table 7, 6, 5).
Step 12: Begin the execution.
Step 13: Deallocates the VM once the execution is completed and again update the status in datacenter table.
Step 14: If there is any task which is not allocated then it will be put in the waiting queue according to the decreasing order of priority.
Step 15: Once virtual machines are available tasks are picked up from waiting queue according to priority.

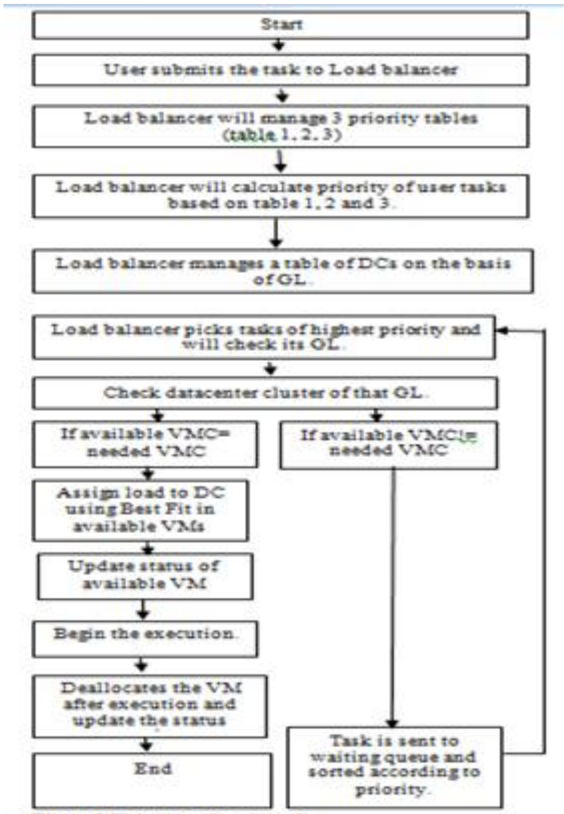
Start
User submits the task to Load balancer
Load balancer will manage 3 priority tables (table 1, 2, 3)
Load balancer will calculate priority of user tasks based on table 1, 2 and 3.
Load balancer manages a table of DCs on the basis of GL.
Load balancer picks tasks of highest priority and will check its GL.
Check datacenter cluster of that GL.
If available VMC= needed VMC
If available VMC!= needed VMC
Assign load to DC using Best Fit in available VMs
Update status of available VM
Begin the execution.
Deallocates the VM after execution and update the status
End
Task is sent to waiting queue and sorted according to priority.

Figure 2 Flowchart for phase 2

IV. Results and Discussion

We have taken cloud analyst tool to represent our results. For simulation three user base (UB) were taken with different priorities. First we put the data center (DC) in the same geographical region of the user base and compute average response time. Next we put data center in different geographical location and compute again the average response time. We find out that the response time for the UB served from DC in same geographical location is far less as compared to UB served from DC in different locations. As per the results we try to minimize the response time [14]. Average response time is reduced if the cloud users will be served from same geographical region and of high priority.UB1 get cloud access first then UB2 and UB3as shown in table 8.Figure 9 variation in response time based on values of table 8.

S.No	User Base(UB) with priority	Average response time(ms)	
		UB served from Data Centre in same Geographical location	UB served from Data Centre in different Geographical location
1.	UB1 (p1)	50.41	250.98
2.	UB2 (p2)	53.04	299.55
3.	UB3 (p3)	58.29	310.95

Table 8: Results comparison

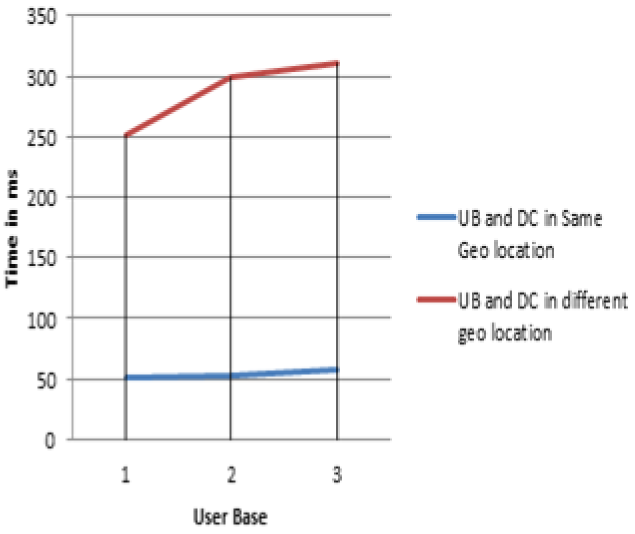

Figure 3: Graph plotted from table 8

V. Conclusion and Future Scope

In today's situation it is obligatory for cloud service providers to have great relationship with their valued client. To maintain this relationship they need to serve their client needs in most efficient manners.

This paper briefs about the priority based load balancing approach that have a numerous advantages over traditional approach. The whole paper is divided into three basic sections. Section one depicts about the basic of load balancing and different load balancing approach. Section two focuses on proposed methodology and flowchart. Methodology depicts how priority is assigned to the users and datacenter and how these valued customers are served effectively and efficiently. Section three sheds lights on the proposed algorithm steps. When a customer sends a request for a cloud access then that customer can be provided services from that geographical area on the basis of his priority. Serving the customer on the basis of geographical locations in turn reduces the response time and hence increases the efficiency of the cloud.

REFERENCES

[1] Khiyaita, A., Zbakh, M., Bakkali, H. El., and Kettani, D.El, 2012, "Load balancing cloud computing: state of art," In National Days of Network Security and Systems (JNS2), IEEE, pages106–109

[2] Seungmin Kang, Bharadwaj Veeravalli, Khin Mi Mi Aung," Scheduling Multiple Divisible Loads in a Multi-cloud System", In 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing,USA, 2014, Page(s):371 – 378.

[3] Q. Cao, B. Wei and W. M. Gong, "An optimizd algorithm for task scheduling based on activity based costing in cloud computing," In International Conference on eSciences Modeling and Simulation Design. AK Peters Ltd ,2009, pp. 1-3.

[4] Chaczko Z, Mahadevan V, Aslanzadeh S, Mcdermid C. Availabil-ity and load balancing in cloud computing. In International Confer-ence on Computer and Software Modeling, Singapore 2011 Sep 16 (Vol. 14)

[5] Monika Choudhary, Sateesh Kumar Peddoju "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment" IJERA ISSN: 2248-9622 Vol. 2, Issue 3, May-Jun 2012, pp.2564-2568.

[6] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," In International Conference on Emerging Trends in Electrical and Computer Technology (ICETECT), 2011, pp. 933-936.

[7] L. Kuan, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, A. Jehangiri, "QoS-aware VM placement in multi-domain service level agreements scenarios", *Proc. Int. Conf. on Cloud Computing*, pp. 661-668, 2013

[8] Mohammadreza Mesbahi,Amir Masoud Rahmani,Anthony Theodore Chronopoulos "Cloud Light Weight : a New solution for Load Balancing in Cloud Computing",In IEEE International Conference on Data Science & Engineering (ICDSE),Kerla,pp44-50,2014.

[9] Martin Randles, David Lamb, A. Taleb-Bendiab, A Comparative Study into Distributed Load Balancing Algorithms for Cloud

Computing”,In 10 Proceedings of the 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops,USA,pp 551-556, 2010.

[10] Hung-Chang Hsiao, Hsueh-Yi Chung, Yu-Chang Chao, “Load Rebalancing for Distributed File Systems in Clouds”, IEEE transactions on parallel and distributed systems, vol. 24, no. 5, pp 951-962, may 2013.

[11] Amandeep, Vandana Yadav, Faz Mohammad, “Different Strategies for Load Balancing in Cloud Computing Environment: a critical Study” , International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 3, Issue 1,pp 52-56 April 2014.

[12] Zhen Xiao, Weijia Song, Qi Chen, “Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment” , IEEE transactions on parallel and distributed systems, vol. 24, no. 6, pp 1107-1117, june 2013.

[13] Preeti Kushwah “A Survey on Load Balancing Techniques Using ACO Algorithm”, International Journal of Computer Science and Information Technologies (IJCSIT), Vol. 5 (5),pp 6310-6314 , 2014.

[14] M. Thenmozhi, N. Tamilarasi “Oppurtunistic Routing Through Delay Analytical Methods in Ad-Hoc Wireless Networks”International journal of Computer Science and Engineering, Vol.6 , Issue.11 , pp.246-253, Nov-2018.

Authors Profile

Mr. Manmohan Sharma pursed Bachelor of Engineering from Rajasthan University in 2005 and Master of Engineering from BITS Pilani in year 2009. He is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computational Sciences, CET-Mody University, Lakshmangarh since 2012. He is a member CSI since 2011. He has published more than 10 research papers in reputed international journals including Thomson Reuters (ESCI & Web of Science) and conferences including IEEE and it’s also available online. His main research work focuses on Cloud computing,Data mining,Soaftware testing. He has 8 years of teaching experience and 3 years of Research Experience.

Dr.V.K.Jain pursed M. Sc. (Electronics), MBA, M. Tech (CS) and PhD in Computer Science and Engineering from Devi Ahilya University, Indore. He is currently working as Dean of CET-Mody University, Lakshmangarh since 2016. He is a member of IEEE ,CSI and ISTD since 2013 .He has published more than 190 research papers and articles in reputed international journals including Thomson Reuters (SCI & Web of Science) and conferences including IEEE and it’s also available online. His main research work focuses on ERP, RFID and EDI Technologies, IT enabled SCM, e- Governance, Software Engineering, Total Quality Management and Research Methodology. He has 22 years of teaching experience and 10 years of Research Experience.