International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 01 | Jan 2021 www.irjet.net p-ISSN: 2395-0072

Priority Based Algorithm for Load Balancing and Scalability in Distributed Environment of Cloud

Palak Jadav[1], Prof and HOD Dr. Gayatri S Pandi(Jain)[2]

[1]Department of computer Engineering, L.J Institute of Engineering & Technology (Gujarat Technology University), Ahmadabad, Gujarat, India
[2]Professor and HOD, L.J Institute of Engineering & Technology (Gujarat Technology University), Ahmadabad, Gujarat, India

---

Abstract - Load unbalancing problem is a multi-variant, multi-constraint problem that degrades performance and efficiency of computing resources. Load balancing techniques cater the solution for load unbalancing situation for two undesirable facets- overloading and under-loading. In contempt of the importance of load balancing techniques to the best of our knowledge, there is no comprehensive, extensive, systematic and hierarchical classification about the existing load balancing techniques. Further, the factors that cause load unbalancing problem are neither studied nor considered in the literature. This presents a detailed encyclopedic review about the load balancing techniques. The advantages and limitations of existing methods are highlighted with crucial challenges being addressed so as to develop efficient load balancing algorithms in future. Parameters used are total number of processes on the node, Resource demands of these processes, Architecture speed of node's processor.

***Key Words*: Cloud Computing, Load Balancing, Virtualization, Response time, Amazon EC2 platform, Priority**

1. INTRODUCTION

Cloud computing a relatively new technology, which has been gaining immense popularity over the last few years where user can rent software, hardware, infrastructure and computational recourse as per user basis. It is an entirely internet-based approach where all the applications and files are hosted on a cloud which consists of thousands of computers interlinked together in a complex manner. These are emerging distributed systems which follows a "pay as you use" model.[2] The number of cloud users has been growing exponentially and apparently scheduling of virtual machines in the cloud becomes an important issue to analyze. Users can submit their jobs into cloud for computational processing or leave their data in cloud for storage. Cloud scheduler must be able to schedule the task properly.[1]Load balancing is use to balance load between multiple resources to get minimum makespan, improve performance, reduce response time and optimal resource utilization.

This paper has been choreograph as follows. Section 1 gives introduction on cloud computing. Section 2 specifies the literature study relevant to this research focus. Section 3 comes out the proposed load balancing algorithm. Section 4 gives out the implementation of this study using tools. Section 5 deals with the conclusion and future directions.

1.1 Load Balancing

There is a limitation to the number of requests a single computer can handle at a given time. When faced with a sudden surge in requests, your application will load slowly, the network will time out, and your server will creak.

You have two options: **scale up or scale out.** When you scale up (vertical scale), you increase the capacity of a single machine by adding more storage (Disk) or processing power (RAM, CPU) to an existing single machine as needed on demand. But scaling up has a limit—you'll get to a point where you cannot add more RAM or CPUs. A better strategy is to scale out (horizontal scale), which involves the distribution of loads across as many servers as necessary to handle the workload. In this case, you can scale infinitely by adding more physical machines to an existing pool of resources.

1.2 Working Of Load Balancing

Firstly, I would like to clear that load in load balancing refers not only to website traffic but also comprises of memory capacity, network and CPU load on the server. The primary function of load balancing technique is to ensure that each system of the network is equipped with the same amount of work. It means neither of the system goes overloaded or underutilized.

The load balancer equally distributes the data depending on how busy the server is. Without load balancer, the client would wait long to process their data that might be frustrating for them.

During this load balancing process, information like job arrival rate and CPU processing rate are exchanged among the processors. Any failures in the application of load balancers can head to some severe consequences such as data loss.

Various companies use different load balancers along with multiple load balancing algorithms. One of the most

commonly used methods or algorithms is the "Round Robin" load balancing.

2. LITERATURE REVIEW

In cloud computing, load balancing is a most important issue. Significant research and development of algorithms in cloud load balancing has grabbed more attention in recent years. Thousands of users have accessed a website at a particular time. It is challenging for applications to manage the load that comes from all these requests at a time. Sometimes, it may result in a breakdown of your entire system.

The load balancing scheduling algorithm has been developed in a Fog Computing environment. Fog layer is basically an intermediate layer between client layer and cloud layer and has been introduced to improve the efficiency of cloud computing environment by proper utilization of bandwidth, as data transmitted or exchanged between cloud and fog for processing get reduced. In fog computing environment, the enormous amount of data of wireless objects such as sensors and Internet of things in distributed environment has been placed the at the edge of the cloud, so that it allows faster accessing, give maximum throughput and met other computing requirement of real time applications. It's become feasible as it has not to be hosted and worked from a centralized cloud thereby fog computing is also called as an edge computing. In our proposed work, a real time efficient scheduling (RTES) load balancing algorithm has been proposed and implemented in the CloudSim tool in the fog computing environment.

In Central Load Balancer (CLB) technique, they tried to avoid the situation of over loading and under loading of virtual machines. The Central Load Balancer (CLB) manages load distribution among various virtual machines and assigns load corresponding to their priority and states. In this way this technique efficiently shares the load of user requests among various virtual machines.

Genetic algorithm mimics the procedure of natural selection which is used as a problem solving technique. At first level of genetic algorithm, individuality is randomly selected from the population to see the closeness between them to solve the problem. An individuality who is fit than others of the population, are allowed to produce the next generation gives the best solution of the problem. Genetic algorithm (GA) is developed to find the most optimized solution for a given job. This algorithm is relevant to search for the solution of high degree of complexity that often involves attributes that are large, non-linear and discrete in nature.

resource allocation algorithm to improve the performance of the applications running in virtual machine in terms of response time and distribute the load across the servers. We conducted a experiment on Xen Cloud Platform. We have used load generating tool stress to generate the load on virtual machines. We used httperf to measure the response time of the applications running in each virtual machine. We have implemented algorithms in shell script using Xen API. Based on the experiments conducted, we have observed that the proposed algorithm, by using the features of scaling and migration has considerably improved the performance of the applications running in virtual machine in terms of response time.

In this research work, we utilized Amazon Web Services cloud condition to build up an effective auto-scaling model. We have analyzed and evaluated the auto scaling strategy in Amazon EC2 with the proposed dynamic ALD algorithm. The simulation results demonstrated evidence that the algorithm has a considerable measure of advantages in limiting the VM's reaction time in the cloud data focuses. Also, we have achieved 70% of throughput in VM load analysis the cloud data focuses by utilizing the proposed ALD algorithm. This particular research work could be stretched out for the auto scaling procedure in a real-time condition for the traffic shaping and traffic flood issues in the cloud computing situations. Also, the proposed ALD algorithm could be applicable to versatile cloud computing to calculate the computational expense through many simulation techniques.

3. PROBLEM STATEMENT

In every system there is always a possibility that some nodes are heavily loaded while some are having fewer loads and some of them are idle among processor in a system. The performance of any web server has been affected by the web traffic usually, and the web server makes a slow response because it gets overloaded. Due to the increased traffic over the Internet, a web server faces challenges to serve the large number of users with high-speed availability.

4. PROPOSED SYSTEM

The proposed PBVMLBA is a load balancing algorithm in which all the allocation and decision of scheduling are completed by a special node called as Load Balancer (LB). This node is responsible for storing knowledge base of entire cloud network and can apply dynamic approach for load balancing. The Data Center Controller (DCC) receives all the requests from the users from all around the world, which is one of the major components of Cloud. Data Center Controller forwards the request to the Load Balancer to assign the request to the available virtual machines. It handles a table which contains the job id of the user request (priority or no priority), completion time of the virtual machine and the state of the virtual machine. Initially, checks the jobs priority, if any priority, allocate the VM and update the status or allocate the VM based on the condition of the completion time of that task is less than to makespan of RPA_LBIMM. To handle further request, this algorithm will search the table and repeat the above procedure until all the tasks get completed.

4.1 PROPOSED ALGORITHM (PBVMLBA)

Step1: Request from user to DCC.

Step2: DCC forwarded the request to LB if DCC=Null go to stpe8

Step 3: Compute the completion time for all tasks for a VM.

Step4: Check if the request is priority or not and check the CT < MP.

Step5: If status = idle

Step 6: Allocate the VM and update VM status. Otherwise Wait for signal until the job gets completed.

Step 7: Repeat step 4 and 9 till some user request exist.

Step 8: if user request complete then stop the allocation process

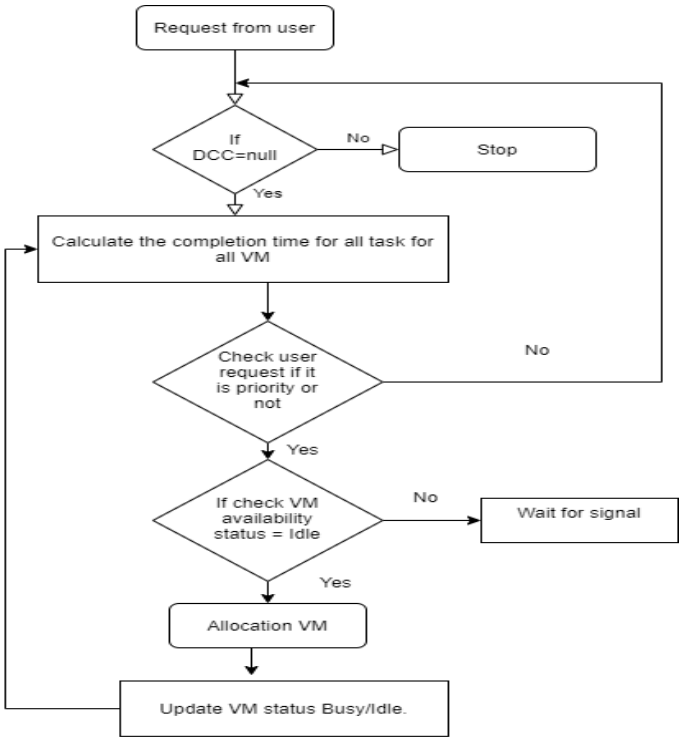

Fig -1: Block diagram of proposed method

4.2 Priority Assignment

CalculatePriority(ResourceList Rs_List)

While(Rs_List!=NULL)

For each resource

/*OC = No. of operations per cycle per processor PN = No. of processors per node

NS = No. of nodes in a cloud*/ /* LL_List contains the amount of parallelism Exhibited by each resource */

LL_List[i] = OC*PN*NS

End While

Find the Max, Min and Mid values in PR_List

/* LJ_List contains the amount of parallelism exhibited by each job */

For each job in LJ_List

If LJ_List[i] >= Maximum

LP_List[i] = High //LP_List contains the level of parallelism value

Else If LJ_List[i] >= Middle

LP_List[i] = Medium

Else LP_List[i] = Low

EndIf

End CalculatePriority

Amount of Parallelism = OC* PN*NS

Where,

OC = No. of operations per cycle per processor

PN = No .of processors per node

NS = No. of nodes in a cloud.

Let m represent number of free resources available in the cloud and n represent the number of jobs present in the queue. The worst case time complexity of the algorithm is O(n logn), when m <= n and O(m logm) when m > n.

Assign Priority Function

AssignPriority(CloudList CL_List)

While(CL_List !=NULL)

For each job

/* CompC_List contains the Computational Complexity of

jobs */

If (CompC_List[i] =High AND LP_List[i] = High)

Priority[i] = 1

Else If (CompC_List[i] = High AND LP_List[i] =

Medium)

Priority[i] = 2

Else If (CompC_List[i] = High AND LP_List[i] =Low)

Priority[i] = 3

Else If (CompC_List[i] = Medium AND LP_List[i] = High)

Priority[i] = 4

Else If (CompC_List[i] = Medium AND LP_List[i] = Medium)

Priority[i] = 5

Else If (CompC_List[i] = Medium AND LP_List[i] = Low)

Priority[i] = 6

Else If (CompC_List[i] = Low AND LP_List[i] = High)

Priority[i] = 7

Else If (CompC_List[i] = Low AND LP_List[i] = Medium)

Priority[i] = 8

ElseIf (CompC_List[i] = Low AND LP_List[i] = Low)

Priority[i] = 9

EndIf

End AssignPriority

5. RESULT AND DISCUSSION

The proposed algorithm PBVMLBA is implemented using cloudsim real time cloud Assume we have a setup of three available resources (VM) to which various users can submit their tasks. Suppose five tasks have been submitted by users. Table1 represents the id, size and the user group of each task. Table 2 represents the id, processing speed and service type of each resource, data present in Table3 is the completion time for all task for all Vs.

Task ID	Task Size (MB)	User Group
T1	100	Ordinary
T2	150	Ordinary
T3	200	Ordinary
T4	250	Priority
T5	500	Ordinary

Table1: Task Parameter

Resource (VM) ID	Resource (VM) Speed(Mbps)	Type
VM 1	20	Priority
VM 2	16	Ordinary
VM 3	10	Ordinary

Table2: Resource Speed

Task	Resource		
	Priority VM 1	**VM 2**	**VM 3**
T1	5	6.25	10
T2	7.5	9.375	15
T3	10	12.5	20
T4	12.5	15.625	25
T5	25	31.25	50

Table3: Computed Completion time (CT)

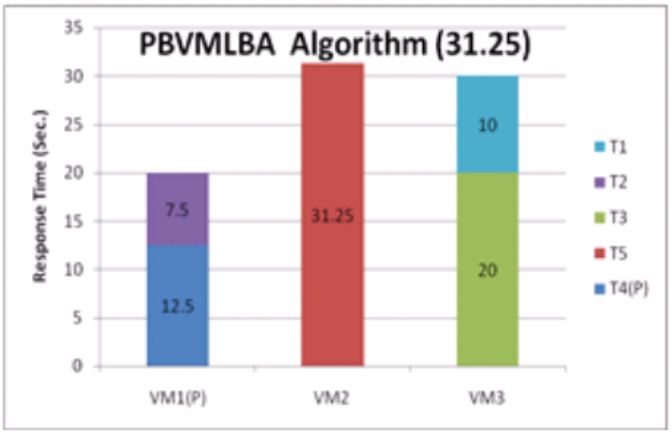

Chart -1: Proposed PBVMLBA

6. CONCLUSIONS

The main focus in this research project is to use the distributed dynamic priority based algorithm used for balancing the load on instances effectively and to improve the system consistency, minimum response time and increase the

throughput. Allocating the resources on virtual machines based on priority achieves the better response time and processing time. Load balancing ensures all instances in a node in the networks to do the equal amount of work at any instant of time. Priority based resource provision to improve the utilization of resources and reducing response time of cloud services.

REFERENCES

[1] Manisha Verma, Neelam Bhardwaj, Arun Kumar Yadav,"Real Time Efficient Scheduling Algorithm for Load Balancing in Fog Computing Environment", International Journal of Information Technology and Computer Science(IJITCS), Vol.8, No.4, pp.1-10, 2016. DOI: 10.5815/ijitcs.2016.04.01

[2] Manisha Verma, Neelam Bhardwaj Arun Kumar Yadav," An architecture for load balancing techniques for Fog computing environment", International Journal of Computer Science and Communication, Vol. 8 Number 2 Jan - Jun 2015 pp. 43-49

[3] Atul Vikas Luthra and Dharmendra Kumar Yadav,"MultiObjective Tasks Scheduling Algorithm for Cloud Computing Throughput Optimization", International Conference on Intelligent, Communication & Convergence, Procedia Computer Science 48(2015) 107- 113.

[4] Brototi Mondala, Kousik Dasguptaa, Paramartha Duttab"Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach", Elsevier, Procedia Technology 4(2012) pp. 783 – 789.

[5] Singh K, Alam M, Sharma S. "A Survey of Static Scheduling Algorithm for Distributed Computing System." International Journal of Computer Applications. 2015; 129(2): 25-30.

[6] Alam M, Varshney AK. "A New Approach of Dynamic Load Balancing Scheduling Algorithm for Homogeneous Multiprocessor System." International Journal of Applied Evolutionary Computation (IJAEC). 2016; 7(2): 61-75.

[7] M. Hines and K. Gopalan, "Post-copy based live virtual machine migration using adaptive pre-paging and dynamic self-ballooning", in Proceedings of the ACM/Usenix International Conference on Virtual Execution Environments (VEE'09), pp.51-60, March 2009.

[8] M. Randles, D. Lamb, and A. Bendiab, "A comparative Study into distributed load balancing algorithms for cloud computing", IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp.551-556, April 2010.

[9] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), pp. 370-375, February 2011.

[10] F. M a, F. Liu and Z. Liu, "Distributed load balancing allocation of virtual machine in cloud data center", IEEE 3rd International conference on Software Engineering and Service Science (ICSESS), pp.20-23, June 2012.

[11] Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 69–73. IEEE Press (2012)

[12] LoadBalancing,https://en.wikipedia.org/wiki/Load_balanc ing_%28computing%29

[13] LoadBalancingOverview,https://cloud.google.com/load-balancing/docs/load-balancing-overview