IJCSE International Journal of Computer Sciences and Engineering Open Access

Survey Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

Best Fit Resource Allocation in Cloud Computing

Harish C. Sharma[1*], Meenakshi Bisht[2]

[1]R/S Uttarakhand Technical University, Dehradun, India
[2]Department of Computer Applications, SGRR University, Dehradun, India

Corresponding Author: hcs19@yahoo.com

DOI: https://doi.org/10.26438/ijcse/v7i3.871875 | Available online at: www.ijcseonline.org

Accepted: 13/Mar/2019, Published: 31/Mar/2019

Abstract - Cloud Computing is performing computing using the internet facility. Computing is performed as on demand of the user. The Cloud Computing Load Balancing algorithms can be applied in Static, Dynamic and Centralized environment. The paper compares and summarizes some of the load balancing strategies in cloud computing environment. The paper discusses some existing cloud load balancing algorithms and compares them according to the usage of resources and makespan at each Node. The paper proposes new improved algorithm for cloud resource allocation in cloud computing and compares its resource allocation with the existing algorithm.

Keywords: Cloud Computing, Load Balancing, Static Load Balancing

I. INTRODUCTION

Cloud Computing [1] provides pool of resources to its end-user that can be reallocated to different purposes within short time frames. Cloud computing power is delivery of computing services from computing power to computing infrastructure, applications etc. to end users as and when it is needed. The cloud computing combines the set of hardware, software, networks, storage, services and interfaces that combine to deliver aspect of computing as a service.[2][3]

There are different reasons for using cloud computing.

- No hardware, software is required.
- Operating System independent
- Dynamic allocation
- Movement of programs
- Scalability
- Pay as you use
- No commitments
- Massive, Web-scale abstracted infrastructure. [4][5]

Cloud Computing uses requirement based hardware as its base. The hardware can be replaced at any time without affecting the cloud. It uses a object – based software container system. For example, a service should be able to be moved from one cloud provider to any other cloud provider with no effect on service. Cloud Computing requires virtualization, abstraction layer for hardware and software. The some services provided by the Cloud Computing are:

- Software-as-a-service (SaaS)
- Platform-as-a-Service (PaaS)
- Infrastructure-as-a-Service (IaaS)

Rest of the paper is organized as follows:
Section II describes Cloud Computing Load Balancing techniques with the some existing load balancing algorithms. Section III compares existing load balancing algorithm according to the execution time. Section IV proposes a Best Fit Algorithm for resource allocation. Section V concludes the above work.

II. LOAD BALANCING

Load balancing [2] is the technique of handling of workload among nodes in a distributed computing environment such that it ensures no node in the system is overloaded or sits idle for any instant of time. Cloud load balancing reduces costs and maximizes resource opportunity. [6] An efficient load balancing algorithm will make sure that every node in the system does more or less same volume of work. The duty of load balancing algorithm is that to map the jobs which are set forth to the cloud domain to the unoccupied resources so that the overall available response time is improved as well as it provides efficient usage of resource utilization. Load Balancing is an important factor in cloud computing environment since we cannot predict the number of requests that are issued at each second in the cloud environment.[7][8] The unpredictability is due to the ever-changing behavior of the cloud. The purpose of load balancing in the cloud computing is in giving the load dynamically among the nodes in order to satisfy the user requirements and to provide maximum resource utilization and maximum throughput by asserting the overall available load to distinct nodes.[9][10]

Active monitoring load AMLB balancing is dynamic in nature. It keeps information about each VM's and the number of request presently assigned to which VM when a request is to assign a new VM reaches. If there are more than one VM, the first recognized is selected and AMLB returns the VM id to the datacenter controller. The datacenter controller sends the request to the VM known by that VM id. The datacenter controller warns the AMLB to new allocation and request is sent to it.[11] [12][13]

AMLB [13] always find least loaded VM for assigning new incoming request but it will not check whether it's previously utilized or not (so some VM over utilized and some is still ideal). [14]

(i) ACTIVE MONITORING ALGORITHM

The load balancing policy attempts to try equal workloads on all the available VMs. The algorithm used is quite similar to the throttled case except that the in this virtual machine selected for the request processing is the one having the least load.

1.ActiveVmLoadBalancer maintains an index table of VM and the number of requests currently allocated to the VM. At the start all VM's have 0 allocations.

2. When a request to allocate a new VM from the Data Center Controller arrives, it parses table and identifies the least loaded VM. If there are more than one, the first one identified is selected.

3. ActiveLoadBalancer returns the VM id to the Data Center Controller.

4. The DataCenterController sends the request to the VM identified by that id.

5. Data Center Controller notifies the Active Load Balancer of the new allocation.

6. Active Vm Load Balancer updates the allocation table increasing the allocations count for that VM.

7. When the VM finishes processing the request, and the DataCenerController receives the response cloudlet, it notifies the ActiveVmLoadBalancer of the VM de-allocation.

8. The ActiveVmLoadBalancer updates allocation table by decreasing the allocation count for the VM by one.

9. Continue from step 2

(ii) MAX-MIN ALGORITHM

The minimum completion time for the all of the submitted task is calculated. Among this minimum computed time for all tasks, the task which takes maximum time is first selected. Max-Min algorithm is similar to the Min-Min. The Max-Min algorithm suffers from starvation where the tasks having the maximum completion time will get executed first while leaving the task which are having smaller completion time.[15][16]

(iii) MIN – MIN ALGORITHM

It starts by executing Min-Min algorithmic steps. It aims to minimize the completion time of the jobs. LBIMM starts executing Min-Min algorithm. Next it picks the smallest size task from the most heavily loaded resource and calculates the completion time of the task on the available resource. If the makespan produced by LBIMM is less than the Min-Min than task is submitted to the resource that produce it. And time of the resources gets updated.[17][18]

(iv) RASA ALGORITHM

RASA is Resource Aware Scheduling Algorithm. It uses Min-Min and Max-Min algorithm alternatively to produce a better performance.

III COMPARATIVE PERFROMANCE OF EXISTING ALGORITHM

The comparative performance of the few existing algorithm is calculated on the basis of the following information presented in form of the table.

Table 1: Task Table

Task	Instruction in Millions
1	150
2	230
3	310
4	475
5	570

The above table (Table: 1) defines five task with their instructions size in millions.

Table 2: Node Processing Capacity Table

Node	Million Instructions per Second (MIPS)
1	60
2	120
3	160

The above table (Table: 2) have three Nodes with their processing capacity in Million Instructions per Second (MIPS).

Table: 3 Execution of task at each Node

	Node-1	Node-2	Node-3

Task-1	2.50	1.25	0.93
Task-2	3.83	1.91	1.43
Task-3	5.16	2.58	1.93
Task-4	7.91	3.95	2.96
Task-5	9.5	4.75	3.56

The above table (Table: 3) shows execution time of each task in different nodes.

The following parameters are being used in the study of existing algorithms and proposed algorithm.

Makespan (m) is the maximum execution time taken by a node.

Makespan(m) = max(C_i) (i)
C_i = Completion time of node i
Resource utilization U_a

$$U_a = \frac{\sum_{i=1}^{N} c_i}{Nm} * 100 \qquad \text{(ii)}$$

Where N is the total number of nodes.

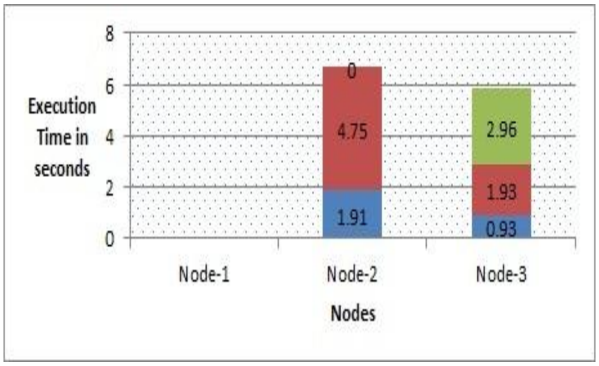

Figure 1: Min-Min Scheduling Performance

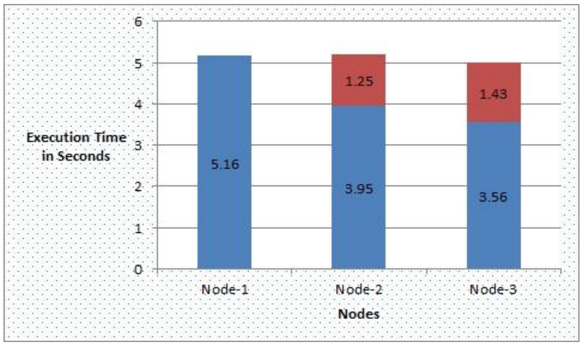

Figure 2: Max-Min Scheduling Performance

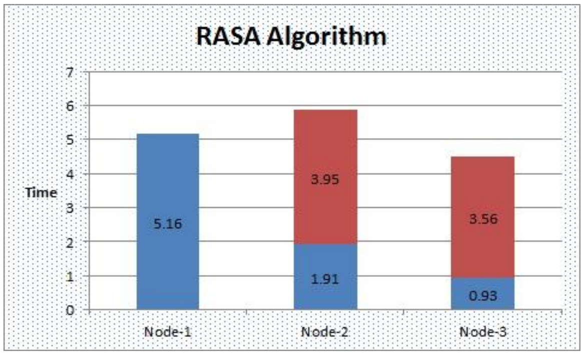

Figure 3: RASA Algorithm

IV. PROPOSED WORK AND ALGORITHM

In our proposed algorithm, we take account of both actual instant processing power of VM and size of assigned jobs. We include two factors in a method of estimating the end of service time in VMs. The criterion of VM selection for the next job is VM can soonest finish it.

On next allocation request, the load balancing algorithm must estimate the time that all queuing jobs and the next job (of current request) are completely done in every VM. The VM that corresponds with the earliest will be chosen to distribute the job.

Measuring the strength of load balancing depends on several factors of which load and capacity. Load is the queue index CPU and CPU utilization. The capacity has average response time of a user request. The objective of our algorithm is to improve the response time and processing time in four scheduling cases:

PROPOSED ALGORITHM

Step 1: Create a Datacenter-Broker and maintaining a status index table of the VM. Jobs currently allocated to the VM

Step 2: When there is a request to allocate a VM, Datacenter-Broker will analyze status index table, estimate the time of completion of the job on each VM. If no free VM is available then VM with earlier completion time and highest MIPS power is selected. If two vm have same completion time one with higher MIPS is selected.

Step 3: The algorithm return Id of the preferred VM to Datacenter-Broker.

Step 4: Datacenter-Broker posts job to VM that are identified by Id.

Step 5: Datacenter-Broker notify the algorithm about the new allocation.

Step 6: The algorithm will update the status index table of VM and of job.

Step 7: When the VM finishes processing requirements and Datacenter-Broker is responded about job, it will update that job is completed in the status index table and decrease 1 job in the index table.

Step 8: Continue to step 2

The execution of the task using the above proposed algorithm is represented in the following figure:

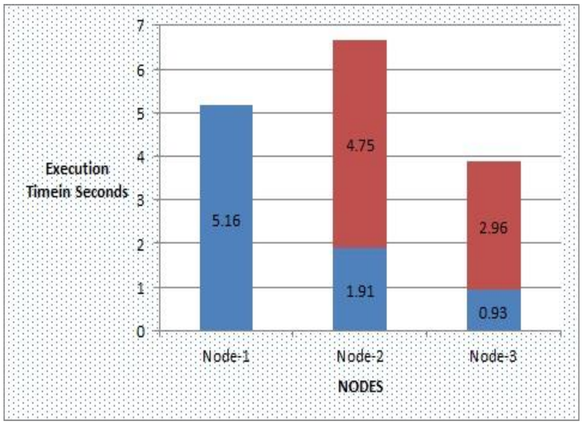

Figure 4: Performance using proposed Algorithm.

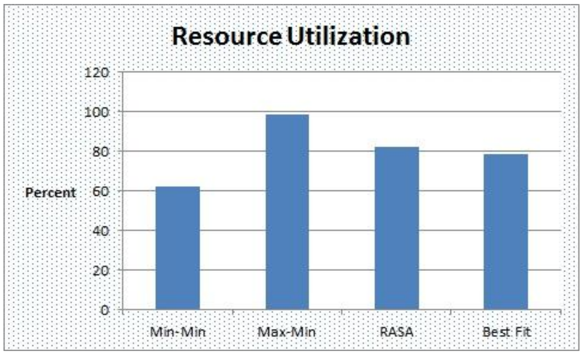

Figure 5: Resource Utilization under Different Algorithms.

The proposed algorithm distributes tasks to all available nodes to improve the performance of the algorithm. As compared to the Min-Min where the all task are allocated to the node-3 and node -2, node -1 remains unutilized.

V. CONCLUSION

The load balancing in cloud computing has great importance. Load balancing helps to improve the system performance. The paper summarizes different cloud computing algorithm and their advantages and disadvantages. The Existing load balancing techniques that have been discussed mainly focus on reducing associated response time, increasing throughput, reducing makespan, and improving performance by some other parameters as CPU, Memory and disk. The paper summarizes with VM migration techniques used for load balancing in cloud computing.

REFERENCES

[1] Peter Mell and Timothy Grance (September 2011). *The NIST Definition of Cloud Computing (Technical report). National Institute of Standards and Technology*: U.S. Department of Commerce. doi:10.6028/NIST.SP.800-145. Special publication 800-145.

[2] Deepti Sharma, Vijay B. Aggarwal, "*Dynamic Load Balancing Algorithms for Heterogeneous Web Server Clusters*", International Journal of Scientific Research in Computer Science and Engineering, Vol.5, Issue.4, pp.56-59, 2017

[3] Raza Abbas Haidri, C.P. Katti, P.C. Saxena, "*A Load Balancing Strategy for Cloud Computing Environment*", 2014 IEEE International Conference on Signal Propagation and Computer Technology (ICSPCT), Ajmer, India, pp. 636-641.

[4] A. Kumar and M. Kalra, "*Load balancing in cloud data center using modified active monitoring load balancer*," 2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA) (Spring), Dehradun, 2016, pp.1-5

[5] Ren, X., R. Lin and H. Zou, "*A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast*", International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE sept. 2011, pp: 220-224

[6] A. Beloglazov, Jemal Abawajy, Rajkumar Buyya, "*Energy Aware Resource Allocation heuristics for efficient management of data centers for Cloud Computing*", Future Generation Computer Systems 28 (2012) 755–768

[7] Mohammadreza Mesbahi, Amir Masoud Rahmani, "*Load Balancing in Cloud Computing: State of art Survey*, MECS, pp. 64-78, March 2016

[8] R. N. Carlheiros, Rajiv Ranjan et al., "*CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*", Wileyonlinelibrary.com, August 2010, DOI: 10.1002/spe.995

[9] M. Katyal, A. Mishra, "*A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment",* International Journal of Distributed and Cloud Computing, Vol. 1, Issue 2, December 2013.

[10] S.B. Shaw, A.K. Singh, "*A Survey on Scheduling and Load Balancing Technique in Cloud Computing Environment*", 2014 International Conference on Computer and Communication Technology (ICCCT), Allahabad, 2014, pp. 87-95.

[11] Geethu Gopinath P P, Shriram K Vasudevan "*An in depth analysis of load balancing technique in cloud computing environment*", 2nd International Symposium on Big Data and Cloud Computing Challenges, VIT University, Chennai, India, ISBCC-2015, pp. 427-432, 2015.

[12] S.Domanal, G. Reddy, "*Optimal Load Balancing in Cloud Computing by Efficient Utilization of Virtual Machines*", IJATES, Vol 3, issue 2, 2014, pp.122-129

[13] Sanyogita Manhas, Jawahar Thakur, "*Comparison of Load Balancing Algorithm in Cloud Computing*", IJCST Vol. 3, issue 4, Dec. 2012

[14] S. Wang, K. Yan, W. Liao, and S. Wang, "*Towards a Load Balancing in a Three-level Cloud Computing Network*", Proceedings of the 3rd IEEE International Conference on

Computer Science and Information Technology (ICCSIT), Chengdu, China, September 2010, pages 108-113.

[15] Jasmin James, Dr. Bhupendra Verma, "*Efficient VM Load Balancing Algorithm for a Cloud Computing Environment*", International Journal on Computer Science and Engineering (IJCSE) , Vol. 4 No. 09 Sep 2012 , pp. 1658-1663.

[16] Boutaba, R., Zhang, Q., & Zhani, M. F. (n.d.). "*Virtual Machine Migration in Cloud Computing Environment*s." In Communication Infrastructures for Cloud Computing (pp. 383–408). IGI Global. https://doi.org/10.4018/978-1-4666-4522-6.ch017

[17] Harish C. Sharma, Himanshu Bahuguna, "*A Survey of Load Balancing Algorithm in Cloud Computing*", International Journal of Computer Engineering and Applications, issn: 2321-3469, Vol. XI, Issue XII, December 2017, pp. 150-160

[18] Harish C. Sharma, Himanshu Bahuguna, "*Comparative Study of Load Balancing in Cloud Computing*", International Journal of Scientific & Engineering Research, Vol. 7, Issue 12, Dec. 2016, pp. 12-17, ISSN 2229-5518

Authors Profile

Harish Chandra is Research Scholar, at Uttarakhand Technical University, Dehradun, India, He received his master's degree in Computer Science and Engineering from Uttrarakhand Technical University, Dehradun, India, in 2016. He is currently working at SGRR University, Dehradun, India as Assistant Professor in Department of Computer Science and Information Technology.

Meenakshi Bisht, working as an Assistant professor under Department of Computer Application & Information Technology at the SGRR University, Dehradun. She is having over 14 years of teaching experience.