International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

An Overview of Load Balancing Algorithms in Cloud Computing for Efficient Resource Utilization

Deepa Kumari[1], Muskaan Nandu[2], Kunj Gala[3], Purvi Harniya[4]

[1]*Deepa Kumari, Dept. of Information Technology, KJ Somaiya College of Engineering, Maharashtra, India*
[2]*Muskaan Nandu , Dept. of Information Technology, KJ Somaiya College of Engineering, Maharashtra, India*
[3]*Kunj Gala, Dept. of Information Technology, KJ Somaiya College of Engineering, Maharashtra, India*
[4]*Purvi Harniya, Dept. of Information Technology, KJ Somaiya College of Engineering, Maharashtra, India*

Abstract - *Cloud computing is a current model for getting to administrations by denotes of the web. This model has a few strains, for example, load-adjusting, security measures, asset orchestrating scaling, Quality of Service (QoS) control, administration availability, and server farm energy use. Among these, quite possibly the most prominent trials are load-balancing. Load balancing issue is a multivariate, multi-requisite issue that corrupts the execution and efficacy of processing assets. Load balancing methods correspond with the answer for load imbalanced circumstances for two unwanted aspects of overloading and under-stacking. Load balancing minimizes the overhead and maximizes throughput by dividing the tasks among the available machines using various suitable load balancing algorithms. In this paper, we have provided an overview of various aspects of load balancing and its algorithms.*

Key Words: **cloud computing, load balancing, significance, SWOT analysis, goals, round robin, stochastic hill climbing, max-min, resource allocation.**

1. INTRODUCTION

Cloud computing is a web-based advancement or a network technology based on the internet that has a part in the swift progress of communication technology, giving a platform for applications and services and a way to configure and adjust. It is a decentralized way of computing with location independence, device independence computational process.

Cloud is usually referred to as 'ubiquitous' which means 'being present everywhere at the same time.' and its contents are configurable and shareable. It has led to the advancement of distributed systems to an extensive computing network using which, firms like Amazon, IBM, Google & Yahoo deliver cloud services to users all over the world. Here, apps and services are offered on-demand to end-users and hence, they need not install it on their local systems.

Load Balancing insinuates the distribution of the inevitable load among various computer collections, computer solutions, relation to the network, disks, servers, CPUs, etc. ensuring that no computing machine is under-loaded, overloaded or idle. It helps in preventing the occurrence of a deadlock and overloading, and assists networks and resources by providing a high throughput and minimum response time and is a major challenge faced in cloud computing.

LB proposes ways to maximize the system output, device performance, usage of resources and also offers accessibility, scalability and availability. The efficiency of a cloud computing model is determined by its utilization of the resources. The best results can be attained by implementing and properly managing the cloud resources. These resources are given to the users through VMs that are Virtual Machines. They make use of Virtualization which utilizes hardware, software or an entity called a hypervisor.

Our main aim is to analyse the types of load balancing and its different algorithms along with a comparative study of their advantages and drawbacks, in this paper. The paper also elaborates on the significance of load balancing and the challenges faced along with suggested methods that can be used in the future.

2. SIGNIFICANCE OF LOAD BALANCING ALGORITHMS

Load Balancing is helpful in cloud environments where immense workloads overwhelm a server easily. As certain performance metrics like availability of service and response time become crucial to some business operations, the need for load balancing also increases. Load balancing is a way to identify available servers and redirect the traffic to them while one server is being overloaded. This ensures no server is sitting idle. Thus, if Load Balancing is not ensured, the new virtual servers won't be able to manage the incoming traffic in an organised manner.

Further, a few benefits of cloud computing are listed using the SWOT analysis (Strength, Weakness, Opportunities and Threat Analysis.) [1]

Table -1: SWOT Analysis of Cloud Computing

Strengths	Weakness
1. Services are economically affordable 2. Simple to understand and use 3. Provides on-demand access 4. It is independent of device and location 5. Can be accessed from all over the world 6. Provides more storage 7. Easy to setup and maintain (automatic updates)	1. No internet means no cloud services 2. Balancing of load is required 3. Cloud security is a big task
Opportunities	**Threats**
1. Agility 2. Scalability 3. Elasticity 4. Monitoring	1. Possible Data Leak 2. Demerits of encryption methods 3. Unreliable firewall may become a problem 4. Improper backup may result in loss of data

The SWOT analysis thus helps in effectively using the resources in cloud computing. Since the flow of incoming requests is unpredictable and heterogeneous and the frequency of requests varies from time to time load balancing is required. During the Peak hours it is necessary to properly manage the load to provide seamless service to the users. The proper resource management is required to reduce the downtime which can be done by using appropriate resource scheduling algorithms/processes. Accessibility of resources is only possible with the help of virtual machines, thus virtualization plays a significant role in cloud computing. Thus, Load balancing is important to provide the best services to the user with a minimum downtime.

3. GOALS OF LOAD BALANCING

The goals of load balancing in cloud computing are defined as follows:

1. Stability of the system remains on track.
2. Promoting a fault tolerance system (performance of the system under partial failure and it's stamina)
3. To enhance performance
4. To use Shared resources fully
5. For adjusting modifications, increasing system's adaptability
6. To maintain system firmness.
7. To protect against system failures.

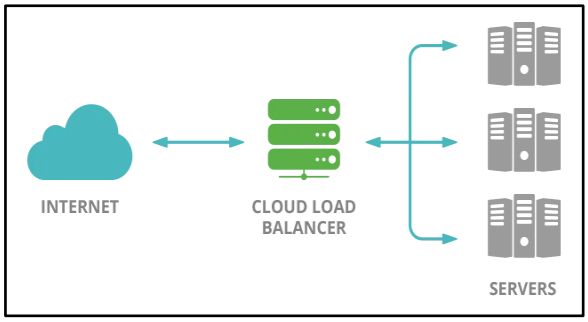

Fig -1: Load Balancing in Cloud Computing

4. TYPES OF LOAD BALANCING

There are majorly two categories of load balancing based on the virtual machine's current state- static and dynamic. These are as follows:

4.1 Static Load Balancing

In static load balancing [3], the information and data about the system like the processing power, storage requirements and client necessities is known beforehand. The condition of the system is already known such as the job resource requirements, processing power of the system, time of computation and the size of the memory and storage device. It follows a collection of predefined rules which don't need to know the current state of the network.This strategy is not extensible although it is quick and efficient and hence it operates well when there is low load variance in the nodes. It's operating period is less as compared to that of dynamic load balancing. Uncertain resource distribution is caused due to the failure of finding the connected servers. A major disadvantage of static load balancing is that the system's actual state is given very less importance to decision making and hence distributed systems with constantly changing states are unacceptable. It does not consider continuous monitoring of the nodes and hence cannot consider load changes during run-time. The strategies for static load-balancing are [4]:

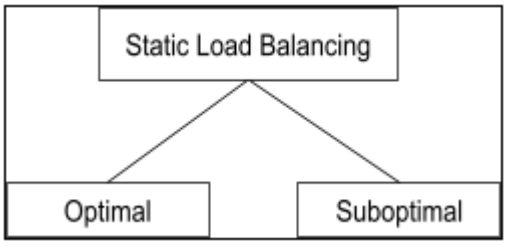

Fig -2: Static Load Balancing

a. Optimal: Resource information is gathered by the data communication network using structured

techniques which is sent to the load balancer where maximum allocation is performed in a limited time period.

b. Sub Optimal: If a decision cannot be correctly given by a load balancer, a suboptimal solution will be decided instead. Min-Min load balancing, Max-Min load balancing, Round Robin, Shortest Job First, Throttled load balancing, Two-phase Opportunistic load balancing, and Central LB are just some static load balancing algorithms.

4.2 Dynamic Load Balancing

This approach [3] is more accurate and efficient and it considers the current state of the system to make further decisions. Dynamic load balancing leverages the fact that it allows the tasks to transfer to an underloaded machine from an overloaded machine and therefore it is adaptable which leads to an improved performance. Some other benefits of dynamic load balancing include increased scalability, resilience to faults, and reduced expenses to increase efficiency which can also handle unreliable processor loads. It monitors the loading of nodes while processing, regularly to calculate the node workload and redistribute the workload among the nodes. Although dynamic load balancing approaches are adaptive in nature and are good for fault tolerance, they have less stability and have high utilization of resources.

Dynamic load balancing algorithms [4] are further divided into:

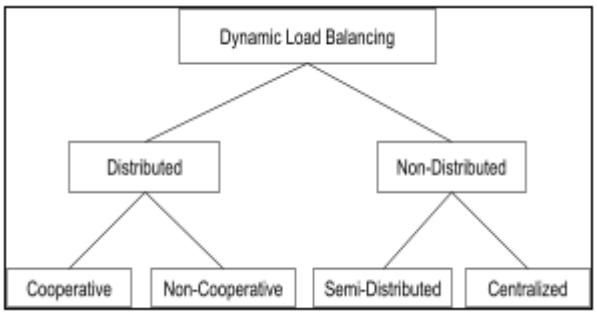

Fig -3: Categories of Dynamic Load Balancing

a. Distributed Dynamic Load Balancing: These algorithms apply dynamic load balancing and assign a functioning schedule to all the nodes in the network. Distributed dynamic load balancing algorithms are further divided into: cooperative and non-cooperative.

b. Non-Distributed Dynamic Load Balancing: The nodes in this approach work independently which provides for a common purpose. Non-distributed algorithms are further classified into:semi-distribution and centralized.

Some of the dynamic load balancing algorithms are as follows:

1. Game Theory Load Balancing Algorithm
2. Stochastic Hill Climbing Algorithm
3. Genetic Load Balancing Algorithm
4. Ant Colony Optimization Based Load Balancing Algorithm
5. Honey Bee Behavior Inspired Load Balancing Algorithm
6. Fireflies Based Load Balancing Algorithm
7. Particle Swarm Optimization Based Load Balancing Algorithm

5. CHALLENGES OF LOAD BALANCING

The most pressing challenges of load balancing [4] are as listed below:

1. Distributed Geographical Nodes
2. Single Point of Failure
3. VM Migration
4. Heterogeneous Nodes
5. Handling Data
6. Load Balancer Scalability
7. Algorithm Complexity
8. Automated Service Provisioning [6]

6. ALGORITHMS OF LOAD BALANCING

6.1 Round Robin Algorithm

Introduction:

1. Round Robin is an algorithm for Static load balancing that is it depends on the past knowledge of resources and software of the system and the choice of distributing workload does not entirely depend on the system's present status.
2. It uses Round robin fashion to allocate jobs and this scheduling is an effective and efficient time scheduling policy.
3. The nodes for load balancing are randomly selected by this algorithm.
4. The important duty, here, is the process of handling load balancing in cloud computing which is carried out by data centers.
5. When the controllers of the data centers get requests from the user, then this request is passed to the round robin algorithm.
6. The partitioning of time in parts inside the RR algorithm is called time quantum or slices of time. Hence, this algorithm is uniquely made for dividing time.[2]

Method:

1. Initially, all processors are kept in a queue that is circular.
2. For every processor within the queue, the server is allocated by the scheduler in the defined slot of time.

3. The new processes would be added at the end of the queue.
4. Scheduler randomly selects the first process from the queue.
5. The selected process will run for a given time slice and when the time slice terminates it is added at the end of the queue after it has been passed on from the server.
6. If the process completes its execution completely before the time slice, then the process is released by the server.
7. Then, the server is assigned to the next process in the ready state in the queue. In this way, the user request is processed in a circular manner using a round robin algorithm.

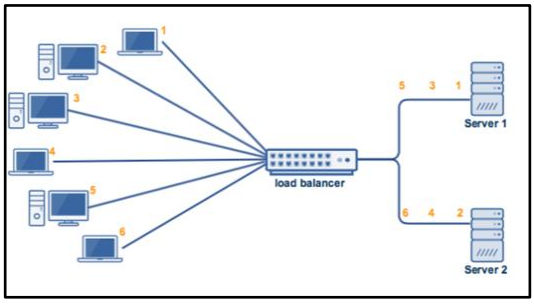

Fig -4: Round Robin Algorithm

In the Fig-5,
BT: (Burst time) The execution period required by a process.
TQ: (Time Quantum) The amount of time allowed for a processor to run.

In Round Robin Algorithm, since the servers are selected on a random basis, there are chances that few servers may be loaded more than their capacity. This will in turn lead to the decline in the performance of load balancing. To overcome this issue, a weighted round robin load balancing algorithm is used which is an extended version of the round robin method. [2] In this method, the administrator can assign a weight to each server based on criteria like traffic handling capacity. So, the servers will gain more requests from clients since they are allocated with more weights.

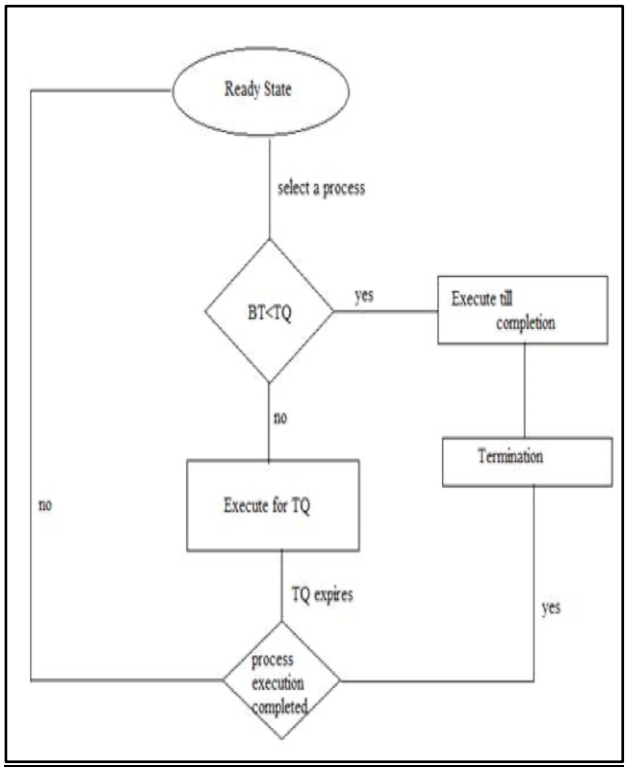

Fig -5: Flowchart of Round Robin Algorithm

6.2 Stochastic Hill Climbing

Stochastic Hill climbing is a variant of hill climbing algorithm but it is an incomplete method, i.e, it does not guarantee an accurate and optimal answer for every input, it rather finds fulfilling assignments for solvable problems with high probability. It is used for solving optimization problems.

In this algorithm, a loop steadily moves in the upward direction of rising graph value and stops only when it reaches a "peak" where there is no neighbour with a greater value. This variant of the hill-climbing algorithm, chooses at random from the uphill moves.

The probability of selection may differ with the curvature of the uphill move. All the components of the set are evaluated based on certain specific criteria to stay in close proximity to a valid task, to enhance the evaluation score of the state. The next task is the one which is the finest component of the collection.

This fundamental operation is repeatedly performed until we reach either a solution or a halt criteria. [7]

Two main components of this algorithm are:

1. A candidate generator: that is responsible for taking one solution candidate and mapping it to a set of successors which are possible.
2. an evaluation rubric/criteria that decides whether each solution is valid or invalid complete assignments, in a way that improved evaluations lead to better solutions.

Steps of algorithm :

1. Maintain an index table of VM servers containing the status of the VM BUSY/AVAILABLE. At the beginning, all VMs are available.
2. A new cloudlet arrives.
3. Generate a query for the next allocation.
4. Generate and assign a VM id randomly.
5. Analyze the allocation table to get the status of the VM.
   If the VM is unallocated:
   Step 5a: Return the VM id.
   Step 5b: Submit the request to the VM associated with that id.
   Step 5c: Update the table of allocation accordingly.
   If the VM is allocated:
   Step 5d: Use a random function to create a random VM.
   Step 5e: Select the VM to be allocated to the job with a probability such that this VM can perform the job efficiently.
   Step 5f: Keep account of performance of the VM. If it does not perform as per the expectation, reduce its probability of assignment in the next iteration
   Step 5g: Update the table of allocation accordingly.
6. When the response cloudlet is received after the request is processed by the VM, generate a notification for VM de-allocation.
7. Continue from Step 2 for the next allocation.

6.3 Max-Min Algorithm

Introduction:

Max-Min Algorithm is a time based algorithm which takes all the available tasks into the system, calculates the minimum time of completion for all the tasks and chooses the task which has the maximum estimated completion time.

This algorithm outperforms the min-min algorithm in cases where short tasks are more in number compared to the long tasks. If there is a very long task in the system, the Max-Min algorithm executes the other short tasks simultaneously while the long task executes. [8]

The longer tasks have higher priority here, but simultaneous execution is allowed. The make span focuses on how many small tasks will get executed simultaneously with the large one. When tasks having maximum completion time are executed first, leaving behind smaller tasks, starvation will occur in this algorithm. The existing algorithm is explained in Fig-6.[9]

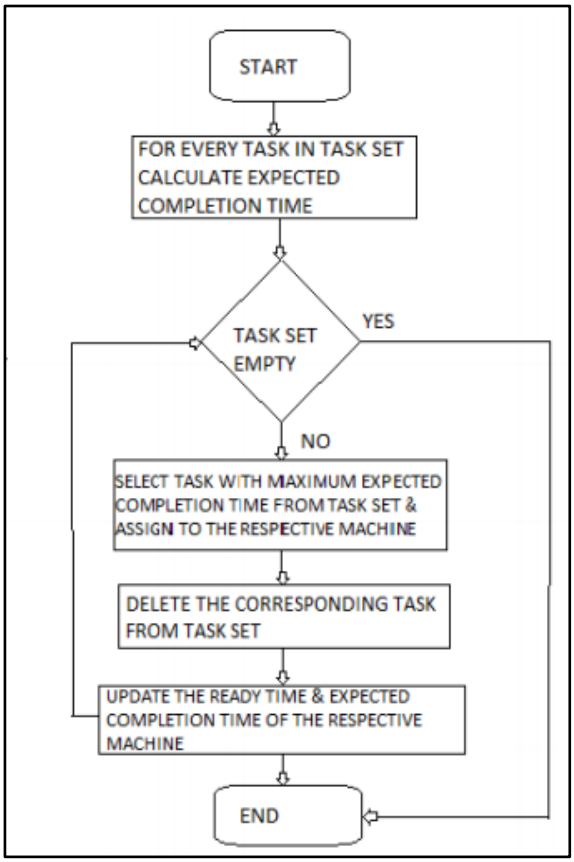

Fig -6: Existing Algorithm of Max-Min algorithm.

Enhanced Algorithm:

Enhanced algorithm is introduced [10] to overcome the disadvantages of the existing Max-Min algorithm. The algorithm is revised to reduce the makespan. It also increases resource utilization. Keeping in mind user priority, the important jobs will execute first. The enhanced approach will first perform the max-min algorithm and if the job is more essential than others then the job's priority is set as high. The higher priority jobs will be processed first and thus the user's demands will be satisfied.

Steps of Enhanced Algorithm:

1. Start
2. for (submitted tasks in meta-task Ti):
   for(resource Rj):
   Calculate Cij = Eij + rj
3. while meta-task != empty:
   All tasks are sorted in decreasing order of burst time.
4. Give priority to tasks, remaining tasks are considered to have normal priority. Consider three classes of priority (Highest, lowest, normal)
5. First highest priority tasks are loaded and arranged in descending order of Burst time.
6. Arrange Vm list in decreasing order on the basis of Resource Cost
   Resource cost = (RAM of VM * Cost/memory) +(Size of Virtual machine*Cost/storage)
7. Repeat above steps till high priority tasks are executed completely.
8. Repeat step 4 first for normal priority class followed by lower priority group until all tasks are executed completely.
9. End

6.4 Resource Allocation

For efficient resource utilization, this algorithm [11] has two phases :
In phase 1 there is DCBT load balancing algorithm (Divide and Conquer and modified Throttled algorithm) which will distribute the tasks and allocate the server which VM has capacity.
And in phase 2 it is based on VM data stored and based on the past data SVM gives the prediction of scaling up or down. For scaling up, a final decision check is made based on the current workload and then if the workload is coming from users then it takes the decision of scaling.

The Proposed Algorithm:

a. Phase 1:

1. For all total tasks
2. For all available RH from servers
   Divide tasks by RH
   Assign tasks to available RH
3. End for
4. End for
5. Return RH for executing the task
6. Find the available RH from servers
7. If (RH (I) or VM <= RH (i+1)) and so on
   Assign RH i
8. Else
   Assign RH i+1
1. End for
2. If (RH or VM is used in previous iteration)
   Search next RH or VM
3. Else
   Assign current RH or VM
4. Return RH for executing the task

b. Phase 2:

1. For i=1 to n
   i utilize server story(i)
2. End for
3. label[i] y[i]
4. For j=1 ton
   Add features x [i]
5. End for
6. Prediction utilization = SVM_Predict(X[i])
7. Return Predict Utilization
8. Set threshold based on <Prediction Utilization, Current Workload>
9. If Utilization < Min threshold
   Call Scale-down
10. If Utilization > Max threshold
    Call Scale up

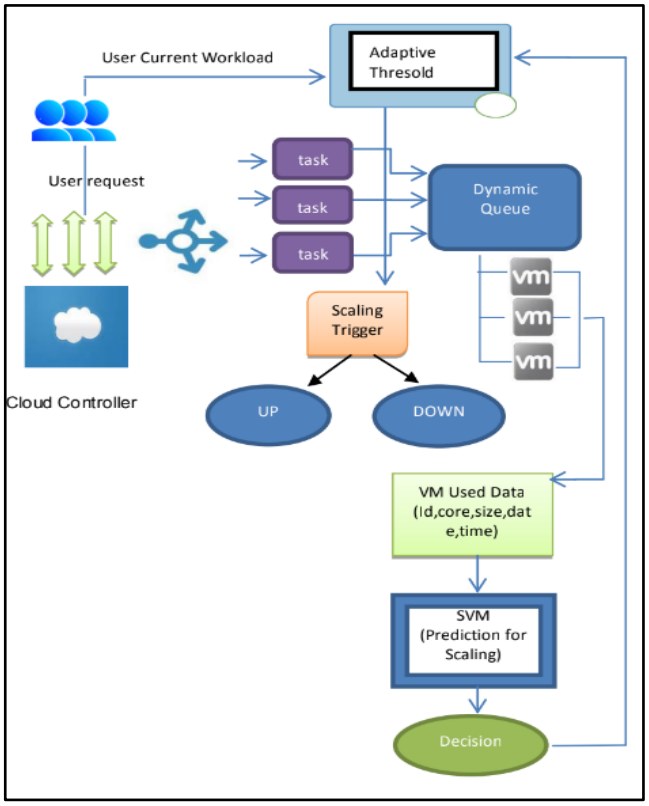

Fig -7: The proposed model of the algorithm

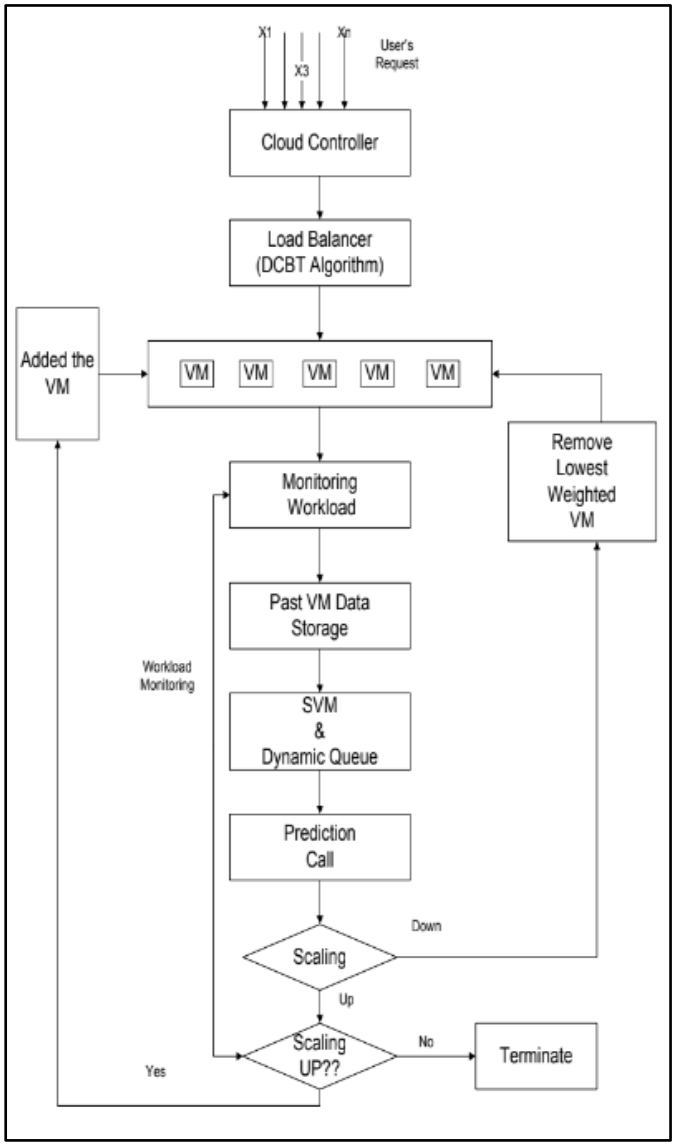

Fig -8: Flowchart of resource allocation algorithm

7. Comparison of Algorithms

The comparison of 5 algorithms explained above are listed in Table 2.

Table -2: Comparison of Algorithms

Algorithm	Advantages	Disadvantages
Round Robin	1. It is simple to understand and implement. 2. The Round Robin Algorithm is widely used. 3. The requests are distributed equally among the available servers.	1. When the time slice is large, the efficiency of the algorithms is the same as that of first come first serve. So, there is no advantage of round robin when the time quantum is large. 3. Deciding time quantum size generates addition load on the scheduler
Stochastic Hill Climbing	1. Issue of bottlenecks is tackled 2. Good distribution of system workload	1. No perfect solution for solving problems of optimization
Max-Min	1. Is a quick & easy algorithm. 2. Improves the overall make-span. 3. Outperforms Min-Min LB algorithm 4. Smaller tasks are higher in number relative to long tasks.	1. May lead to starvation
Resource Allocation	1. Ensures efficient resource utilization. 2. Is a dynamic algorithm which uses scaling up/ down based on the state of the system & distributes the user's load on the server.	1.There is an additional overhead to take the final decision of scaling up/down.

8. RESULTS

The above algorithms were compared based on their performance metrics as described in [12][13].

Table -3: Performance Metrics of Algorithms

LB Algo - rithms		Performance	Throughput	Overhead	Fault Tolerance	Response Time	Resource Utilization	Scalability	Power Saving
Round Robin	G	✔	✔	✔	×	✔	✔	✔	×

Stochastic Hill Climbing	G	✔	✔	×	×	✔	✔	×	×
Max-Min	G	✔	✔	✔	×	✔	✔	×	×
Resource Allocation	G	✔	✔	×	✔	✔	✔	✔	×

Here, G = General, NP = Natural Phenomenon

9. Conclusions and Future Work

Cloud computing ensures the delivery of customer support at all times. A major challenge in Cloud computing is load balancing, since overloading of a device may lead to dreadful results. Hence, there is a constant need for an effective LB algorithm with the help of which resources can be efficiently utilized. The major aim of load balancing is to serve the user's requirements by allocating the work-load across several network nodes and amplifying the usage of resources, thereby increasing the performance of the cloud system, minimizing the response time, and reducing the number of job rejection which results in a reduction of the energy consumed and the carbon emission rate. This paper describes the significance of cloud computing, goals of Load balancing in cloud computing, types of load balancing and load balancing algorithms. We described five load balancing algorithms and presented their comparison based on their performance metrics [14]. There will be demand for new fully autonomous dynamic Load Balancing algorithms in the future that will allow increased resource utilization, improved degree of mismatch, effective task migrations, lowered make-span and shorter time span.

REFERENCES

[1] S. Rao Gundu1, C. Arur Panem and A.Thimmapuram: "Real-Time Cloud-Based Load Balance Algorithms and an Analysis", Published online: 28 May 2020

[2] N.Tadapaneni, "A Survey of Various Load Balancing Algorithms in Cloud Computing" Published online: April 2020.

[3] N. Verma, V. Sharma, M. Kashyap and A. Jha, "Heuristic Load Balancing Algorithms in Vulnerable Cloud Computing Environment," 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Greater Noida, India, 2018.

[4] Muhammad Asim Shahid, Noman Islam, Muhammad Mansoor Alam, Mazliham Mohd Su'ud and Shahrulniza Musa: "A Comprehensive Study of Load Balancing Approaches in the Cloud Computing Environment and a Novel Fault Tolerance Approach", July 27, 2020.

[5] P. Kumar and R. Kumar, "Issues and challenges of load balancing techniques in cloud computing: A survey", ACM Comput. Surveys, vol. 51, no. 6, pp. 1-35, Feb. 2019.

[6] R. Z. Khan and M. O. Ahmad, "Load balancing challenges in cloud computing: A survey", Proc. Int. Conf. Signal Netw. Comput. Syst., vol. 396, pp. 25-32, 2016.

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

[8] Kanani, Bhavisha, and Bhumi Maniyar. "Review on max-min task scheduling algorithm for cloud computing." Journal of Emerging Technologies and Innovative Research 2.3 (2015): 781-784.

[9] Chawda, P.. "An Improved Min-Min Task Scheduling Algorithm for Load Balancing in Cloud Computing." (2016).

[10] V. Soni and Dr. N.C. Barwar: "Performance Analysis of Enhanced Max-Min and Min-Min Task Scheduling Algorithms in Cloud Computing Environment'' in International Conference on Emerging Trends in Science, Engineering and Management (ICETSEM-2018), Pune, Maharashtra, India on 14th October 2018

[11] V. Joshi and U. Thakkar, "A Novel Approach for Real-Time Scaling in Load Balancing for Effective Resource Utilization," 2018 International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, 2018.

[12] S. Kumar Mishra, B. Sahoo and P. Paramita Parida :"Load balancing in cloud computing: A big picture", Journal of King Saud University – Computer and Information Sciences 32, (2020)

[13] U. Patel and H. Gupta, "A Review of Load Balancing Technique in Cloud Computing" Published online: April 24, 2019.

[14] Bhargavi K*, Sathish Babu B and Jeremy Pitt: "Performance Modeling of Load Balancing Techniques in Cloud: Some of the Recent Competitive Swarm Artificial Intelligence-based", November 20, 2019.