Load Balancing Techniques: Need, Objectives and Major Challenges in Cloud Computing- A Systematic Review

Nitin Kumar Mishra
School of InformationTechnology (SOIT),
RGPV, BHOPAL

Nishchol Mishra
SOIT, RGPV, BHOPAL

ABSTRACT

One of the emerging areas in the field of information technology (IT) is Cloud Computing. It is internet based technology which emphasizes its utility and follows pay-as-you-go model. Load balancing is a critical issue in cloud computing. It is a technique which uses multiple nodes and distribute dynamic workload among them so that no single node is overloaded. The main goal of load balancing includes optimal utilization of resources which increases the performance of the system and minimization of resource consumption which minimizes carbon emission rate. This paper is mainly focused on the concept of load balancing techniques in cloud computing. This review helps in analyzing the issues of existing load balancing algorithms and gives a comparison among these algorithm on the basis of different qualitative metrics like throughput, reliability, power saving feature, performance, scalability, associated overhead etc.

Keywords

Load Balancing, Green Computing, Carbon Emission, Dynamic Load Balancing, Load Balancing Algorithms, Virtualization.

1. INTRODUCTION

Cloud computing is the emerging internet based technology which emphasizes commercial computing. Cloud is a platform providing dynamic pool resources and virtualization. It also facilitates the scalable IT resources such as services, applications and infrastructure. These resources work with internet on pay-per-use basis which helps in adjustment of capacity in a fast way. The cloud computing moves both computing and data from portable PC's and desktop to large data centers. cloud computing can adjust on demand changes which in-turn eliminates the cost of capital needed in software and hardware. Thus, cloud computing provide a frame work for suitable access to computing resources and that too in on demand fashion. By virtue of cloud computing, resources can be granted and removed very quickly as well as with less service provider interaction. Cloud computing also increases availability of resources. Basically the key focus of cloud computing is resource allocation and scheduling (RAS) which is achieved by using its policies and algorithms[1]. These have a direct effect on cloud cost and performance.

Load balancing is one of the major issues in cloud computing[2] i.e., When one or more components of any service failed, load balancing helps in continuation of the services by implementing provisioning and de-provisioning of instances of applications without fail. Thus, Load Balancing is a mechanism for distributing the dynamic local workload evenly across all the nodes in the whole cloud. It will also avoid the situation where some nodes are heavily loaded while others are idle or doing little work. Load balancing increases the overall performance of the system along with its resource utilization property. This working principle of load balancing helps to achieve high user satisfaction. By increasing overall performance of the system efficient distribution of every resources can be done.

In cloud computing consumption of resources and conservation of energy are not considered as a important topic of discussion but there inclusion along with proper load balancing helps in reducing costs and making enterprises greener. Another addressable feature is scalability in cloud computing which is enabled by load balancing. All these features increases resource utilization in such a manner that will reduce energy consumption as well as carbon foot prints, resulting in achieving green computing.

2. LOAD BALANCING

Load balancing can be define as a method for distributing workload on the multiple computers or a computer cluster through network links to achieve optimal resource utilization which maximizes throughput and minimizes overall response time. It minimizes the total waiting time of the resources as well as avoids too much overload on the resources. In this technique traffic is divided among servers, so that data can be sent and received without maximum delay.

One of the crucial issue in cloud computing is to divide the workload dynamically. Workload of a machine means the total processing time it requires to execute all the tasks assigned to the machine. Load balancing is the process of improving the performance of the system by shifting of workload among the processors. The benefits of distributing the workload includes higher resource utilization ratio which further leads to enhancing the overall performance thereby achieving maximum client satisfaction. Through load balancing every virtual machine in cloud system does the same amount of work that maximizes the throughput and minimizing the response time. Hence load balancing is one of the important factor to heighten the working performance of the cloud service provider.

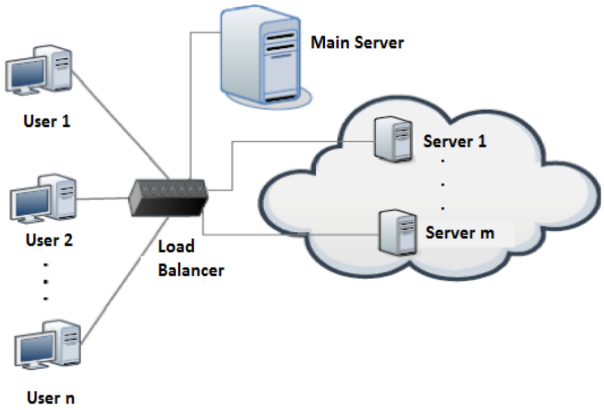

Fig 1: Load Balancing In Cloud Computing

2.1 Why Load Balancing Needed In Cloud Environment

Load balancing in clouds provide a mechanism for distributing the excess dynamic local workload evenly across all the nodes. Load balancing is used for achieving a high user satisfaction and resource utilization ratio, making sure that no single node is overwhelmed, which will improve the overall performance of the system. If load balancing used in a proper way then it can achieve optimal resource utilization which will minimize the resource consumption. Another important advantages of using load balancing are implementing fail-over, enabling scalability, avoiding bottlenecks, over-provisioning, reducing response time and achieving Green Computing in clouds. The factors responsible for it are:

i. **Limited Energy Consumption**
Load **balancing** reduces the amount of energy consumption by avoiding over heating of nodes or virtual machines due to excessive workload.

ii. **Reducing Carbon Emission**
Energy consumption and carbon emission are interconnected to each other since both are directly proportional. As load balancing help in reducing energy consumption it will automatically reduce carbon emission helping in achieving Green Computing [3].

2.2 Objectives of Load Balancing

The main objectives of load balancing are:

1. To increase the performance significantly.
2. To have a backup plan in case the system fails even partially.
3. To keep the system stable.
4. To provide future enhancement in the system.

2.3 Types Of Load Balancing On The Basis Of Cloud Environment

Load balancing classification can be illustrated by following diagram:

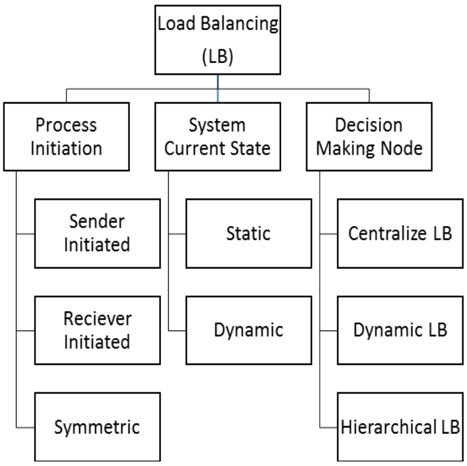

Fig 2:Load Balancing Classification

As shown in figure 3.2, load balancing can be divided into three based on process initiation:

1. **Sender Initiated**
In this algorithm firstly client sends request and then a receiver is assign to him to receive his workload i.e. the sender initiates the process.

2. **Receiver Initiated**
In this algorithm firstly receiver sends a acknowledged request to a sender who is prepared to share the workload i.e. the receiver initiates the process.

3. **Symmetric**
It is a combination of both sender and receiver initiated type of load balancing algorithm.

On the bases of the current state of the system, load balancing algorithm can be categories into two:

1. Static Load Balancing

In static algorithms [3], prior knowledge about the system is already known which includes processing power, memory, performance and data about user's requirements. These algorithms do not need the information regarding current state of the system. This type of algorithms have serious drawbacks in case of sudden failure of system resource, tasks and also task can't be shifted during its execution for load balancing. Round robin is a example of static load balancing algorithm which divides the traffic equally among servers. A lot of problems were appearing in round robin algorithm to overcome these problems a new algorithm is proposed called Weighted Round Robin. The main concept behind this algorithm is that each server has been assigned a weight and then, server having the highest weight receives more connections. In equal weighted condition, servers will receive balanced traffic. This approach is generally defined during the design or implementation period of the system.

2. Dynamic Load Balancing

Dynamic algorithms[4] are decision concerning load balancing based upon the current state of the system i.e. any prior knowledge about the system is not required. This will overcome the draw backs of static approach. The dynamic algorithms are complex, but they can provide better performance and fault tolerance. Some policies are used in dynamic load balancing algorithm. These can be define as following:

1. Transfer Policy: Selection of a job in dynamic load balancing algorithm for transferring from a local node to a remote node is known as transfer policy.
2. Selection Policy It specifies the processors involved in the load exchange.
3. Location Policy: Selection of a destination node for a transfer task in load balancing algorithm is referred as location policy or location strategy.
4. Information Policy: Collection of information about the node in the system in load balancing algorithm is referred to as information policy or information strategy.

i. Load estimation Policy: Total amount of workload on a processor or machine is estimated by this policy.

ii. Process Transfer Policy: : It is used for deciding which task is to be executed locally or remotely.

iii. Priority Assignment Policy: In it priority are assigned to processes for executing them locally and remotely.

iv. Migration Limitting Policy:
It sets a limit on the maximum number of times a task can migrate from one machine to another machine.

Figure 3 shows the interaction among these policies in dynamic load balancing algorithm.

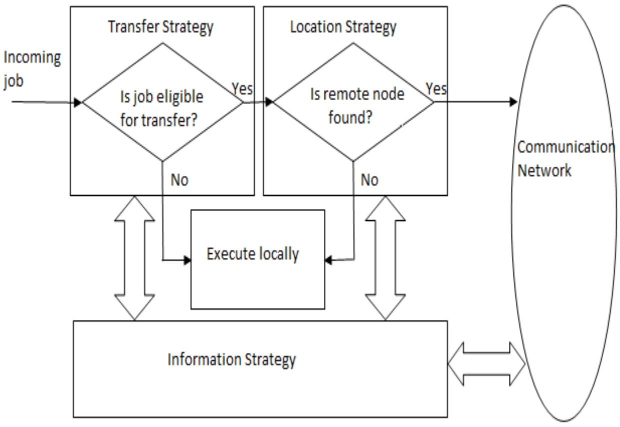

Fig 3: Interaction among Components of A Dynamic Load Balancing Algorithm

Depending on the decision making, the load balancing algorithm can be divided into three categories:

1. Centralized Load Balancing

In centralized load balancing technique, a single node performs all the allocation and scheduling. This node has knowledge base of entire cloud network which can be applied for static or dynamic approach for load balancing. This technique reduces the time required to analyze different cloud resources but creates a great load on the centralized node. This type of network is not provide fault tolerance and recovery also difficult in case of failure.

2. Distributed Load Balancing

In this technique, resource allocation or task scheduling decision are not taken by a single node. Multiple domains are responsible to make accurate load balancing decision. A local knowledgebase is maintained by every node in the network for ensuring efficient distribution of tasks in static environment and re-distribution in dynamic environment. Failure in intensity of a node is not neglected in distributed scenario which makes the system as fault tolerant.

3. Hierarchical Load Balancing

In this algorithm various levels of the cloud are involved. slave mode operation technique is used in hierarchical load balancing. These can be implemented using tree data structure where every node in the tree is balanced under the supervision of its parent node. Master uses light weight agent process for getting statistics of slave nodes. Parent node take the decision based on gathered information.

3. EXISTING LOAD BALANCING TECHNIQUES IN CLOUD COMPUTING

The existing load balancing algorithms can be categories into static and dynamic algorithms. Three phase hierarchical scheduling is proposed in paper [5].

Static Load Balancing Algorithms

3.1 Round Robin Load Balancing

In this algorithm [6] the round robin mechanism follows a time slice while processing the data. Each process is going to execute in the time slice and then switch to other process and follow on ring manner. In round robin until the all processes completed their task, a balance technique is followed in order to balance the process in a group. The process is going to occur in round robin until the all processes complete their task such that a balance technique is implemented in order to balance the process in a group. This algorithm is widely used in web servers where http requests are of similar in nature and distributed equally.

3.2 Shortest Job Scheduling Algorithm

In this algorithm [7] shortest executable job is selected first. The approach follows to perform the complete execution of short jobs to utilize the resources in completion of heavy jobs. Shortest job had an advantage that the waiting time for the processes is less which makes it a powerful approach.

3.3 Min-Min Load Balancing

An approach for the load balancing where all the information related to the job is available in prior. Min-Min algorithm [8] begins with a set of all pending jobs. First of all, a time taken to complete a task is calculated. The job with minimum completion time is selected. Then, the node which has the minimum completion time for all jobs is selected. Finally, the selected node and the selected job are mapped. The ready time of the node is updated. This process is repeated until all unassigned jobs are assigned. The advantage of this algorithm is that job with the smallest execution time is executed. The drawback of this algorithm is that some jobs may experience starvation.

3.4 Max-Min Load Balancing

It works on the Opposite strategy as compare with min-min approach where the maximum value is consider to execute first. Max-Min [9] is almost same as the min-min algorithm except that after finding out minimum completion time of jobs, the maximum value is selected. The machine that has the minimum completion time for all the jobs is selected. Finally the selected node and the selected job are mapped. Then the ready time of the node is updated by adding the execution time of the assigned task. A maximum time taken process is shifted one by one.

3.5 Two-phase (OLB + LBMM) load balancing algorithm

It is proposed by S.-c. Wang et al. [10] that merges Opportunistic Load Balancing (OLB) and Load Balance Min-Min (LBMM) scheduling algorithms to achieve better executing efficiency of the system. Working principle of OLB algorithm is to put each and every single node in working condition so that the goal of cloud computing can be achieved. On the other hand LBMM scheduling algorithm is used for minimizing the execution time of the tasks on node which reduce of overall completion time. Combining these two algorithms help achieving proper utilization of all resources and enhances the work efficiency in the network of multiple processor.

3.6 Central load balancing policy for virtual machines (CLBVM)

A. Bhadani et al. [11] proposed a policy that balances the load evenly in a distributed virtual machine/cloud computing environment. This technique known as Central Load Balancing Policy for Virtual Machines (CLBVM). This policy improve the overall performance of the system but does not consider the systems that are fault-tolerant.

Dynamic Load Balancing Algorithms

3.7 Power Aware Load Balancing (PALB)

In this [12] process firstly utilization percentage of each computing node is estimated for the working module, Which decides the number of operating computing nodes while other nodes are completely shut down or not in working condition. This algorithm has three section in working module: balance section, upscale section and downscale section. Balance section is responsible for determining initialization process where virtual machine is going to start. The second section power-up the additional computing nodes and the third downscale section shut-downs the idle compute node in the process participant. The algorithm perform its best on consumption as compare with the other existing algorithm in same category.

3.8 Fuzzy Active Monitoring Load Balancing (FAMLB)

Srinivas Seth et al. [13] proposed a load balancing algorithm based on fuzzy logic. This algorithm uses two parameters processor speed and load on virtual machine. In [14], the authors have introduced a new fuzzy logic based dynamic load balancing algorithm with additional parameters like memory usage, bandwidth usage, disk space usage, virtual machine status and named it as fuzzy Active Monitoring Load Balancer (FAMLB)

3.9 Throttled Load Balancing

In the paper [15] author described algorithm in which the client first requests the load balancer to find a suitable virtual machine to perform the required operation for the incoming process. In Cloud computing, there may be multiple instances of virtual machine. These virtual machines can be grouped based on the type of requests they can handle. So as per the incoming requests it work accordingly. Whenever a client sends a request, the load balancer will first look for that group and if it is ready to accept and handle the request it is going to assign request to it.

3.10 Honeybee Foraging Behavior

This is a load balancing algorithm [16] which is analogous to the behavior of how honey bees finds and reaps their food. There is a category of bees called forager bees. They search for food and after getting it they come back for announcement. They announce it by doing a dance called waggle dance. This dance is the description of available metadata food. After getting the information the scout bees follows the searcher bees towards the food location for storage purpose. Then returning to beehive they again do a waggle dance which gives the information of available food to be occupied and then more food can be consumed by the honey bee. In load balancing with the increasing and decreasing web server’s demand, the services are also assign dynamically to map the users changing demands. Within virtual servers the server are clustered, each virtual server having its own virtual service queue. Like the quality that bee shows by their waggle dance each server also calculate a profit or reward from the request queues. This reward can be measured by the amount of time that the CPU spends on the processing of a request. In case of honey bees the dance floor is analogous to an advert board here. This mechanism in virtual server and load balancing is also useful while occupy the server for a process.

3.11 Active Clustering

This load balancing algorithm works on the principle of grouping similar one’s and working on them group wise. The performance of the system is enhanced with high resources thereby increasing the parameter outcome using the algorithm. This algorithm is degraded with an increase in system diversity[17]. A node initiates the process and select another node called the matchmaker node from its neighbors, satisfying the criteria that it should be a different type than the former one. The following set of processes are executed one by one up to process end.

i. The match maker algorithm perform mechanism to form a connection between matchmaker node and neighbor of it which is of the same type as the initial node.

ii. The matchmaker node then detaches the connection b/w itself and the initial node.

3.12 Biased Random Sampling

In this paper author M. Randles et al. [18] proposed a dynamic approach which is based on random sampling of the system domain to achieve self-organization, thus balancing the load in all system available node. Here a virtual graph is used, with the connectivity of each node which shows the load on the server. Regarding job execution and completion in the network:

i. Whenever a node does or executes a job, it deletes an incoming edge, it indicate the occupation of the resources.

ii. Once a job is finished it also free the resources.

The addition and deletion of processes is done by the mechanism of random sampling algorithm. The walk starts at any one node and at every step a neighbor is chosen randomly. The last node is selected for allocation of load. Alternatively, another method can be used for selection of a node for load allocation, that being selecting a node based on certain criteria like computing efficiency, etc. Yet another method can be selecting that node for load allocation which is under loaded i.e. having highest in degree. If b is the walk length, then, as b increases, the efficiency of load allocation increases. We define a threshold value of b, which is generally equal to log n experimentally. A node upon receiving a job, will execute it only if its current walk length is equal to or greater than the threshold value. Else, the walk length of the job under consideration is incremented and another neighbor node is selected randomly. When, a job is executed by a node then in the graph, an incoming edge of that node is deleted. After completion of the job, an edge is created from the node initiating the load allocation process to the node which was executing the job. Finally what we get is a directed graph. The load balancing scheme used here is fully decentralized, thus making it apt for large network systems like that in a cloud.

3.13 Generalized Priority Algorithm

In this algorithm [19] the tasks are prioritized according to the size of the tasks such that the task with highest size gets the highest priority in the system and execute first at its best. Also the virtual servers are prioritized according to their million instruction per second (MIPS) value in the virtual server distribution system, such that the Server with the highest MIPS value gets highest priority. Hence the load balancing is done accordingly and it get the maximum utilization of the resources according to the data size in progress.

3.14 Join-Idle-Queue

Y. Lua et al. [20] proposed an algorithm for web services and systems called as Join-Idle-Queue load balancing algorithm. It facilitates large scale load balancing with distributed dispatchers. In each dispatch firstly load balancing algorithm idles the processors for the availability and then do allotment of the task to processors in such a way that reduces the queue length at each server. This algorithm remove the load balancing work from critical path of request processing which helps in effective reduction of the system load.

3.15 Genetic Algorithm Based Load Balancing

It is proposed by Kousik Dasgupta and Brototi Mandal[21]. This algorithm thrives to balance the load of the cloud infrastructure while trying to minimize make span of a job. Genetic based approach follow some rules and randomization according to the network load effectively.

3.16 Ant Colony Optimization

It is algorithm proposed by Anamika jain and Ravinder Singh [22]. Ant colony optimization is meta heuristic approach for load balancing system. The heuristic algorithm has guaranteed for optimal solution with any number of jobs and machines that are used in it. The approach ACO is based on nature of real ants which form the network in order to process the job. It is proposed by Dorigo at [1991].The ants are moving for searching food from source to nest in a path. The ants communicates with each other using a liquid evaporating content named as pheromone. During the path, other ants follow the same path with the help of pheromone. If the intensity of pheromone is high, ants follow that path otherwise no optimal solution. The social agents are ants, birds, honey bees for their problem solving capacity which change dynamically through environments. In the proposed strategy ants are moved in the graph where all the nodes are connected and randomly moved until an optimal solution has found.

3.17 Stochastic Hill Climbing Technique

Kousik Dasgupta and Brototi Mandal proposed [23] a novel load balancing strategy by using Stochastic Hill Climbing algorithm. The hill climbing chooses randomly form the uphill moves with effective probability. Author of a local optimization approach Stochastic Hill climbing utilize the resources and the algorithm is used for allocation of incoming jobs to the servers or virtual machines (VMs).

3.18 Decentralized Content Aware Load Balancing

H.Mehta et al.[24] proposed a different content aware load balancing policy known as Workload and Client Aware Policy (WCAP) which explain the unique and special property of the requests as well as computing nodes by a unique and special property (USP). USP helps the scheduler in deciding the best and fit resources to execute the process. This technique has less overhead and implemented in decentralized manner. This technique improves the searching performance by using the content information. It also improve the utilization of resources by reducing the idle time of the computing nodes.

3.19 Server-Load Balancing For Internet Distributed Services

A. M. Nakai et al. [25] presented a distributed server based technique for web servers. It facilitates the reduction in service response time by using a protocol that bounds the redirection of requests to the closest remote servers without overloading them. A middleware is used in this technique to implement this protocol. To endure overload, web server uses Heuristic. Heuristic scheme provide a surety to get balance of load based on the job size and also guarantee to not get repetition of same size job in single node.

3.20 Load Balancing Based On A Lock-Free Multiprocessing Solution

X. Liu et al. [26] proposed a technique that avoids the use of shared memory whereas other multiprocessing load balancing solutions used shared memory concept and locks to maintain a user session. That’s why this technique is called a lock-free multiprocessing solution for load balancing. This solution helps in improving the overall performance of load balancer in a multi-core environment by running multiple load-balancing processes in one load balancer.

3.21 Load balancing Scheduling Strategy For Virtual Machine Resources

J. Hu et al. [27] presented a scheduling strategy that uses previous logs and the current status of the server. By using a genetic algorithm approach this strategy helps in reduction of dynamic migration in the system. It helps in resolving the issues of load-imbalance and high cost of migration thus achieving better resource utilization. A drawback for the system is that, sometime previous logs can not give the current scenario at its level best.

3.22 Load Balancing strategy for Virtual Storage (LBVS)

H. Liu et al.[28] proposed a load balancing virtual storage strategy (LBVS) that provides a large scale net data storage model and storage as a service model based on Cloud storage. Storage virtualization is achieved using an architecture that is three-layered and load balancing is achieved using two load balancing modules. It improves the efficiency of concurrent access by using replica balancing which reduces the response time and enhances the capacity for backup. LBVS improves the flexibility, robustness and use rate of storage resource of the system.

3.23 Load Balancing based on a Task Scheduling Algorithm

Y. Fang et al. [29] proposed a two-level task scheduling Mechanism. It is used for meeting the dynamic requirements of users and obtaining a high utilization of resources. The mechanism is based on load balancing. In this mechanism firstly tasks are first mapped to virtual machines and then virtual machine to host resources. This mechanism improves the task response time as well as resource utilization.

3.24 Load Balancing mechanism based on ant colony and complex network theory (ACCLB)

Z. Zhang et al. [30] presented a load balancing mechanism which is based on ant colony and complex network theory in an open cloud computing federation. In this mechanism small-world and scale-free characteristics of a complex network are used for obtaining better load balancing. The technique is excellent in fault tolerance as well as have good scalability as it can overcome heterogeneity by adapting the dynamic environment. Thus by using this mechanism the system performance can be improved.

3.25 Event-driven

V. Nae et al. [31] proposed an algorithm for real-time Massively Multiplayer Online Games (MMOG) called event driven load balancing. This algorithm uses capacity based mechanism for input capacity processing. On reception of input the algorithm analyzes the component in context of the resources. This algorithm also analyzes the components on the bases of global states of the game session and after completion of this session, algorithm generates the game session load balancing action. Event-driven algorithm can scale up or down a game session on multiple resources depending on variable user load. The only drawback of it is the occasional QoS breaches.

3.26 CARTON

R. Stanojevic et al. [32] proposed a mechanism for cloud control named as CARTON that merges the use of load balancing (LB) and distributed rate limiting (DRL). Load balancing is useful in equally distributing the jobs among servers so that we can minimize the associated costs and DRL is used to ensure fair resource allocation among servers. DRL also adapts server capacities for the dynamic workloads so that performance equality is achieved at each server with very low computation and communication overhead. This algorithm is also simple and easy to implement.

3.27 Compare and Balance

This algorithm [33] uses the concept of compare and balance to reach an equilibrium condition and manages unbalanced system's load. A comparison apply based on the sorting approach on the basis of probability (no. of virtual machine running on the current host and whole cloud system).The current node selects randomly a node and compares the load with itself. The algorithm is simple and follow traditional approach for load balancing in cloud computing.

3.28 Vector Dot

A. Singh et al. [34] proposed Vector Dot dynamic algorithm for load balancing. This algorithm is capable to handle the hierarchical complexity of the data-center. It can also handle multidimensionality of resource load across servers, network switches and storage in an agile data center. This agile data center has integrated server as well as storage virtualization technologies. Vector Dot uses dot product to distinguish nodes based on the item requirements and removes overload on servers, switches and storage nodes.

4. QUALITATIVE METRICS FOR LOAD BALANCING

Qualitative metrics consist some parameters which are useful in order to find best algorithm among them. The different qualitative metrics or parameters that are considered important for load balancing in cloud computing are discussed as follows:

4.1 Throughput

This is used for estimating total number of task, whose execution has been completed successfully. A high throughput is required for better performance of the system.

4.2 Associated Overhead

The amount of overhead that is produced by the execution of the load balancing algorithm. Minimum overhead is expected for successful implementation of the algorithm.

4.3 Fault tolerant

It is the ability of the algorithm to perform correctly and uniformly even in conditions of failure at any arbitrary node in the system.

4.4 Migration time

The time taken in migration or transfer of a task from one machine to other machine in the system. This time should be minimum for improving the performance of the system.

4.5 Response time

It is the minimum time that a distributed system takes to respond for executing a specific load balancing algorithm.

4.6 Resource Utilization

It is the degree to which the resources of the system are utilized. A good load balancing algorithm provides maximum resource utilization.

4.7 Scalability

It determines the ability of the system to accomplish load balancing algorithm with a restricted number of processors or machines.

4.8 Performance

It represents the effectiveness of the system after performing load balancing. If all the above parameters are satisfied optimally then it will highly improve the performance of the system.

5. LOAD BALANCING CHALLENGES IN CLOUD COMPUTING

Although cloud computing has been widely adopted but research in cloud computing is still in its early stages. Some scientific challenges still remain unsolved by the scientific community. Some main challenges in load balancing are following:

5.1 Automated Service Provisioning

A key feature of cloud computing is elasticity i.e., resources can be allocated or released automatically. Then how can we use or release the resources of the cloud, by keeping the same performance as traditional systems and using optimal resources?

5.2 Virtual machine Migration

Virtualization make it possible to see an entire machine as a file or set of files. Virtualization also provides a facility to move a virtual machine among heavily loaded physical machines so that balance can be achieved among them. The main objective of virtual machine migration is to distribute the load in a datacenter or set of datacenters. Then how can we dynamically distribute the load when moving the virtual machine to avoid bottlenecks in Cloud computing systems?

5.3 Energy Management

Energy saving is a main challenge for load balancing algorithms. Energy saving is extremely needed in cloud environment to achieve green computing. There is always a need of energy efficient algorithm which minimizes resource consumption but keeps acceptable performance.

5.4 Stored Data Management

In the last few years exponential growth has been seen in stored data across networks. This data may be belong to any company or any individual. The management of data storage for companies or individuals, become a major challenge for cloud computing. Then how can we distribute the data to the cloud for optimum storage of data while maintaining fast access?

5.5 Emergence Of small Data Centers for Cloud Computing

Small datacenters have some benefits over large data centers like they are more beneficial, cheaper and less energy consumer. Small providers can deliver cloud computing services leading to geo-diversity computing. Load balancing will become a problem on a global scale to ensure an adequate response time with an optimal distribution of resources.

6. COMPARISON AMONG DIFFERENT TYPES OF LOAD BALANCING TECHNIQUES

In the previous section different load balancing techniques proposed by various researchers have been discussed. Table1 gives a comparative analysis of different existing load balancing techniques with respect to different performance parameters and table 2 present a comparative study of three load balancing algorithm on the basis of policies adopted by them.

Table 1. Comparison among different existing load balancing techniques based on various metrics

Metrics/ Techniques	Performance	Through-put	Overhead	Fault Tolerance	Migration Time	Response Time	Resource Utilization	Scalability	Power Saving
Round Robin[6]	Yes	Yes	Yes	No	No	Yes	Yes	Yes	No
Dynamic Round Robin[6]	No	Yes	Yes	Yes	Yes	No	Yes	No	No
Shortest Job Scheduling Algorithm[7]	No	No	No	No	No	No	Yes	No	No
Min-Min[8]	Yes	Yes	Yes	No	No	Yes	Yes	No	No
Max-Min[9]	Yes	Yes	Yes	No	No	Yes	Yes	No	No
OLB+LBMM[10]	Yes	No	No	No	No	No	Yes	No	No
CLBVM[11]	Yes	Yes	No	No	No	Yes	Yes	No	No
PALB[12]	No	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
FAMLB[13,14]	No	Yes	Yes	No	Yes	Yes	Yes	Yes	No
Throttled[15]	Yes	No	No	Yes	Yes	Yes	Yes	Yes	No
HoneyBee Foraging[16]	No	No	No	No	No	No	Yes	No	No
Active Clustering[18]	No	No	Yes	No	Yes	No	Yes	No	No
Biased Random Sampling[18]	Yes	No	Yes	No	No	No	Yes	No	No
Generalized Priority algorithm[19]	No	Yes	No	No	Yes	No	Yes	No	No
Join Idle Queue[20]	Yes	No	Yes	No	No	Yes	No	No	No
Genetic Algorithm[21]	Yes	No	No	No	No	No	Yes	No	No
Ant Colony Optimization[22]	Yes	No	No	No	Yes	No	Yes	No	No
Stochastic Hill Climbing Technique[23]	Yes	Yes	No	No	No	Yes	Yes	No	No
Decentralize Content Aware[24]	Yes	No	Yes	No	No	Yes	Yes	Yes	No
Server-based LB for Internet distributed services[25]	Yes	No	No	No	No	Yes	No	Yes	No
Lock-free multi-processing sol. for LB[26]	Yes	Yes	No	No	No	No	No	No	No
Scheduling strategy on LB of VM resources[27]	No	No	Yes	No	No	No	Yes	No	No
LBVS[28]	Yes	No	No	Yes	No	Yes	No	Yes	No
Task Scheduling based on LB[29]	Yes	No	No	No	No	Yes	Yes	No	No
ACCLB[30]	Yes	Yes	No	No	No	No	No	Yes	No
Event Driven[31]	No	No	No	No	No	No	Yes	Yes	No

Table 2. Comparative study of three dynamic load balancing algorithm on the basis of policies adoption

Algorithm	Initiated On	Initiated By	Job Transfer	Transfer policy	Selection Policy	Location Policy	Information Policy
Sender-Initiated	Job Arrival	Sender	Preemptive	Threshold Based	Consider Only New Jobs	Random threshold or shortest	Demand Driven
Receiver-Initiated	Job Departure	Receiver	Non Preemptive	Threshold Based	Consider All Jobs	Random	Demand Driven
Symmetrically Initiated	Both	Both	Both	Threshold Based	Both	Depends On Design	Demand Driven

7. CONCLUSION

In this review, various load balancing algorithms in cloud computing environment are analyzed and various issues are also discussed which must be taken into account during designing of new load balancing algorithms. In literature existing static and dynamic load balancing algorithms are discussed and comparative analysis is performed on the basis of different metrics parameters like performance, scalability, throughput, resource utilization, fault tolerance, response time etc. Further work can be done by exploring new efficient load balancing algorithm which can maintain better balance among parameters and also helps to achieve green computing.

8. AKNOWLEDGEMENT

This research is supported by School of Information Technology (SOIT), RGPV University (State Technological Universities of Madhya Pradesh) Bhopal, India. Sincere thanks to all the faculty members who provided insight and expertise that greatly assisted the research. Thanks to Garima Tanwar [M.Tech] and Vikas Yadav[M.Tech] for remarks that greatly enhanced the manuscript.

9. REFERENCES

[1] Tinghuai Ma, Ya Chu, Licheng Zhao & Otgonbayar Ankhbayar, “Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm”IETE Technical review Volume 31, Issue 1, pp.4-16,January 2014.

[2] B. P. Rima, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, pp.44-5l, August 2009.

[3] Nidhi Jain Kansal, Inderveer Chana, "Cloud Load Balancing Techniques: A Step Towards Green Computing", IJCSI, Vol. 9, Issue 1, January 2012.

[4] Hitesh Bheda, Hiren Bhatt,” An Overview of Load balancing Techniques in Cloud Computing Environments”,International journal of Engineering and Computer Science Volume 4, pp.9874- 9881,JANUARY 2015.

[5] Sukhvir Kaur, Supriya Kinger “Review on Load Balancing Techniques in Cloud Computing Environment”, International Journal of Science and Research (IJSR) Volume 3,Issue 6, June 2014.

[6] Nusrat Pasha, Dr. Amit Agarwal and Dr.Ravi Rastogi,“Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment” International Journal of Advanced Research in Computer Science and Software Engineering Volume 4, Issue 5, May 2014.

[7] Poonam Devi,Mr. Trilok “GabImplementation of cloud computing by using short job scheduling” International Journal of Advanced Research in Computer Science and Software Engineering.

[8] T. Kokilavani and Dr. D.I. George Amalarethinam, “Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing”International Journal of Computer Applications Volume 20– No.2, pp.0975-8887, April 2011.

[9] Shahrzad Aslanzadeh, Venkatesh Mahadevan, Christopher Mcdermid,“ Availability and Load Balancing in Cloud Computing” International Conference on Computer and Software Modeling IPCSIT vol.14 IACSIT Press, Singapore 2011.

[10] S. Wang, K. Van, 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 (ICC SIT), Chengdu, China, pp.108-113, September 2010.

[11] A. Bhadani and S. Chaudhary, “ performance evaluation of web servers using central load balancing policy over virtual machine on cloud”, proceedings of third Annual ACM.

[12] J. M. Galloway, K. L. Smith, and S. S. Vrbsky, “Power aware load balancing for cloud computing,” in Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp.19–21, 2011.

[13] S. Sethi, A. Sahu, and S. K. Jena, “Efficient load balancing in cloud computing using fuzzy logic,” IOSR Journal of Engineering, vol. 2, no. 7, pp.65–71, 2012.

[14] Z. Nine, M. SQ, M. Azad, A. Kalam, S. Abdullah and R. M. Rahman, “fuzzy logic based dynamic load balancing in virtualized data centers” In fuzzy system (FUZZ), IEEE International conference on, pp. 1-7, 2013.

[15] Ms.Nitika, Ms.Shaveta, Mr. Gaurav Raj; “Comparative Analysis of Load Balancing Algorithms in Cloud Computing”, International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May 2012.

[16] M. Randles, D. Lamb, and A. Taleb-Bendiab, “Experiments with Honeybee Foraging Inspired Load Balancing” Proceedings IEEE International Conference

on Developments in eSystems Engineering (DESE), pp.240 – 247,Abu Dhabi,Dec 2009.

[17] http://www .loadbalancing.org/.

[18] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings IEEE International Conference on Advanced Information Networking and Applications Workshops, Perth, Australia, pp.551-556, April 2010.

[19] Dr. Amit Agarwal, Saloni Jain "Efficient optimal algorithm of task scheduling in cloud computing environment" International Journal of computer Trends and Technology (IJCTT).

G. Kliotb, Y. Lua, Q. Xiea, A. Gellerb, J. R. Larusb, and A. Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", An international Journal on computer Performance and evaluation, In Press, Accepted Manuscript, Available online 3 August 201l.

[20] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta,Jyotsna Kumar Mandal,Santanu Dam , "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing", Elsevier (CIMTA) 2013.

[21] Anamika Jain, Ravinder Singh, " Review of Peer to Peer Grid Load Balancing Model Based on Ant Colony Optimization with Resource Management" Volume 3, Issue 4, April 2013 IJARCSSE.

[22] Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, "Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach",Elsevier (C3IT) 2012.

[23] 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 (lCWET),pp.370-375, February 2011.

[24] A. M. Nakai, E. Madeira, and L. E. Buzato,"Load Balancing for Internet Distributed Services Using Limited Redirection Rates", 5th IEEE Latin-American Symposium on Dependable Computing (LADC),pp.156-165 2011.

[25] Xi. Liu, Lei. Pan, Chong-Jun. Wang, and JunYuan. Xie, "A Lock-Free Solution for Load Balancing in Multi-Core Environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (lSA), pp.1-4 2011.

[26] Hu, 1. Gu, G. Sun, and T. Zhao, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", Third International symposium on parallel architecture, algorithms and programming(PAAP),pp.89-96,2010.

[27] Hao Liu, Shijun Liu, Xiangxu Meng, Chengwei Yang, Yong Zhang, LBVS:A Load Balancing Strategy for Virtual Storage, IEEE International Conference on Service Sciences, 2010.

[28] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318,pp.271-277,2010.

[29] Z. Zhang, and X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", Proceedings of 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China,pp.240- 243, May 2010.

[30] V. Nae, R. Prod an, and T. Fahringer, "CostEfficient Hosting and Load Balancing of Massively Multiplayer Online Games", Proceedings of the lith IEEE/ ACM International Conference on Grid Computing (Grid), IEEE Computer Society,pp.9- 17, October 2010.

[31] R. Stanojevic, and R. Shorten, "Load balancing vs. distributed rate limiting: a unifying framework for cloud control", Proceedings of IEEE ICC, Dresden, Germany, pp. 1-6, August 2009.

[32] Y. Zhao, and W. Huang, "Adaptive Distributed Load Balancing Algorithm based on Live Migration of Virtual Machines in Cloud", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Republic of Korea,pp.170-175, August 2009.

[33] A. Singh, M. Korupolu, and D. Mohapatra, "Server-storage virtualization: integration and load balancing in data centers", Proceedings of the ACM/IEEE conference on Supercomputing (SC), November 2008.