A REVIEW ON LOAD BALANCING ALGORITHM

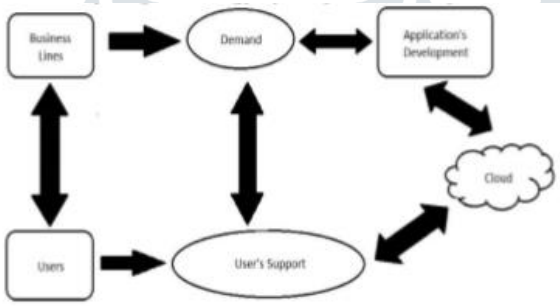
JOY RAKESH Y
Computer Science Department,
CHRIST(Deemed to be University)
Banaglore.

***Abstract*— Cloud Computing in general term means anything which involves hosted services which are connected over the internet. With the advancement of the internet over the years, cloud computing is one of the emerging technology. Due to the increase in usage of cloud computing services, there are certain challenges which the user and the service providers come across. Due to the increase in the load, the resources are not utilized efficiently, this is the main reason for Load Balancing. The main concept of load balancing is to equally divide the load to all the nodes so that the resources are utilized efficiently. This paper gives a brief overview of load balancing, different types of load balancing, different algorithms used, and briefly all the parameters involved.**

Keywords*— *Cloud Computing, Load Balancing, Clustering, Throughput, Resource Utilization.

I. INTRODUCTION

Today everyone is connected to the internet. This helped in emerging of a new technology called "Cloud Computing". It is one of the widely used and most proven technology services. Features of cloud computing such as reliability, scalability and is highly flexible for the end user and the services providers. Following features put more focus towards this technology where scientist and research scholars are shifting towards it. Cloud computing pay per use model is a widely used capital expenditure, which is highly cost-efficient and available 24 * 7. Maintenance is one purpose that made all the company adapt to this technology. Here the resources are provided only when the user request for it.[4] The basic scenario which makes us understand how cloud computing is used in the industry. Suppose the system manager is told to buy computer/software and other equipment. Requirement a person to maintain all the pc is also required. The cost of the company increases. These costs can be reduced when the company shifts from normal assess to a cloud framework.



Load balancing is one of the issues where the research scholars and scientists are working. There are two types of load balancing algorithms static and dynamic load balancing algorithm. A static load balancing algorithm is used is a homogenous environment, which is more stable and produced efficient results in such an environment. But here it is not possible to make a dynamic change during the execution time. Whereas dynamic algorithm is more flexible and yields good results compared to static algorithms. Here the consideration of different attributes in the system is done during the runtime. The dynamic algorithm adapts changes in a heterogeneous environment to give better results. As a result, some algorithms give better results and some not resulting in inefficiency in the performance. Even though cloud computing has many advantages, there are issues to deal such as load balancing among the resources, security, scheduling of task etc. This paper gives a brief insight into how load balancing algorithms can increase efficiency to the user and the service providers.

II. LOAD BALANCING IN CLOUD COMPUTING

The primary aim of load balancing is to distribute the load equally so that all the nodes are utilized efficiently to give better performance and throughput in such a way that no node is under loaded or overloaded. In load balancing, there are certain attributes that are used to measure the performance of load balancing algorithms such as Throughput, Fault tolerance, and adaptability. [6]

Throughput: It is a process that makes sure that an increase in efficiency by executing a maximum number of processes in a minimum amount of compilation time.

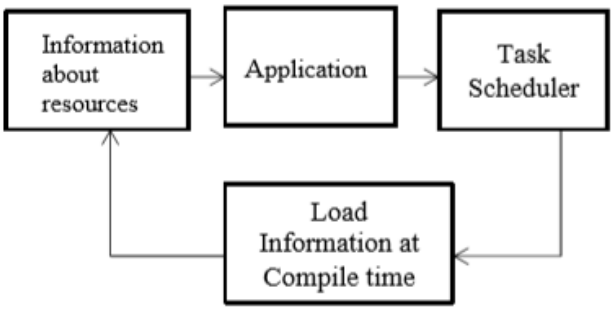
Fault Tolerance: In case a fault occurs in a system it should not stop other systems from working. The algorithms must be capable of handling fault tolerance perfectly.

Adaptability: The algorithm must be capable of handling requests in a dynamic situation and allocate take in a lesser amount of time.

Basically Load Balancing is divided in two types' Static Load Balancing and Dynamic Load Balancing

A. STATIC LOAD BALANCING

Static load balancing is most suitable when data flow is isolated uniformly among all the servers. In static load balancing is applied for frameworks having very low variations in load. Here the performance of the processors is calculated only at the start of the execution and this calculation requires a better understanding of system resources. At static load, the balancing task is performed by the scheduler based on the load in the compile time [9]. There are several drawbacks such as task one set cannot be moved amid to another node for balancing the load.



Different Static Load Balancing Algorithm

1) ROUND ROBIN ALGORITHM

In this algorithm, the processes are divided between the processors in an equal manner. Whereas the distinctive processes do not follow this mechanism. Here the processes are divided into a session where each session has a timestamp to it [9]. Due to this sometimes the process is either overloaded or lightly loaded. The use of the RR algorithm is where the HTTP request is comparative in nature and conveyed similarity then the round-robin algorithm is utilized. The mathematical model for the Round Robin algorithm is given below [10]:

Create X1, X2;
X1 = store ready process
X2= store blocked process
New process submitted to end of X1
If task time interval finished then
Move to end of X1.
If I/O request or swapped out request is made by process
Then Move process from X1 to X2.
If I/O operation is completed or ready to move from blocked processes then Move process from X2 to X1

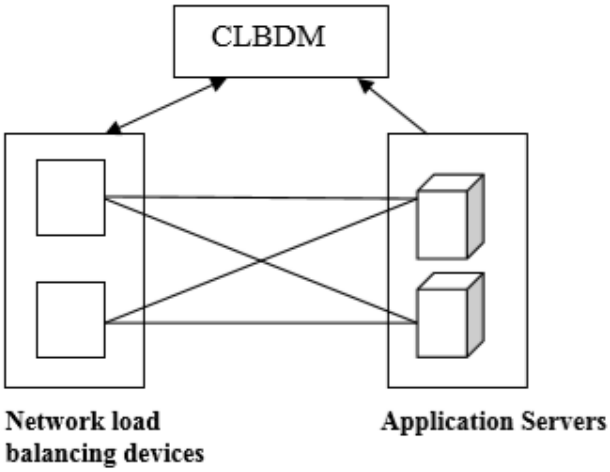
2) Active Clustering Load balancing Algorithm

In this algorithm, the idea of clustering is used. Cluster is a grouping of similar node and non-grouping of dissimilar nodes. Dynamic clustering is an enhanced technique for random sampling. The basic rule in this algorithm is to cluster similar nodes and working is based on grouped nodes. Here gathering the node increase the throughput and performance [2].In this algorithm "matchmaker" a technique where once the execution begins the first node chooses is the neighbor node. The neighbor node is taken as a coordinate node, which associates the neighbor a node that is similar to the initial node [7]. Finally, the match produced node is removed. This procedure is done repeatedly to adjust the heap similarity. This framework execution is enhanced exceedingly by expanding the throughput. There is a proficient usage of assets where there is an expansion in throughput. The mathematical model for the active clustering are as follows.

To compute the distance between the two nodes, Euclidean distance has to be calculated.

3) Central Load Balancing Decision Model

Round robin algorithm is based on the session which switches once it's over on the application layer. The main purpose to overcome the drawbacks in the round robin algorithm. This algorithm consists of attributes such as execution time and threshold value [9]. The working of the central load balancing decision model is that the processes are divided between the processes and these processes are allocated a round robin method to processes. Based on that execution time which is calculated between the node and the request from the user, a comparison of the execution time takes place with the threshold value. The connection between the user's request and the node is terminated only when the execution time is above the threshold value. However one of the disadvantages of the central load balancing decision model is it can enter the looping iteration and give a wrong decision between the nodes which results in poor performance.



4) Min-Min Load Balancing Algorithm

In this algorithm, it begins with all the unsigned tasks. First of all minimum compilation time for all the task are found [10]. The tasks having minimum execution time is chosen. The execution time for all the tasks is redefined for the machine. Here the fundamental issue is Starvation.
Min-Min Load balancing algorithm is given below:

```
Procedure Minmin(Task Ti)
{
        Find execution_Time for each task
        Store the execution_Completion_Time of task Ti  in orderQueue
        {
                for each task Ti in orderQueue
                {
                        obtain minimum completionTime from orderQueue;  assign task to vm;  update the
                        execution_Completion_Time;
                }
                Until orderqueue empty;
        }
}
```

5) Map Reduced Based Entity Resolution Model

Lars Kolb discovered map reduced based entity resolution model [8]. This algorithm is divided into two parts: map() and Reduce(). Map() function performs the sorting and cleaning with the help of part() method which breaks down the large dataset into the smaller datasets. Comp() method is used to compare the similar tasks and groups using the grouper with the help of reducing () method. Here the overloading of processes is reduced to minimal with the use of part() method [13].

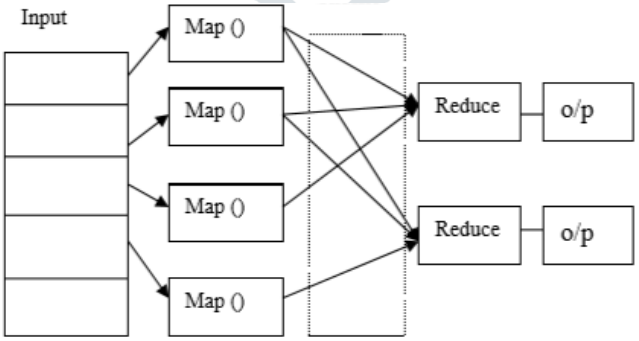
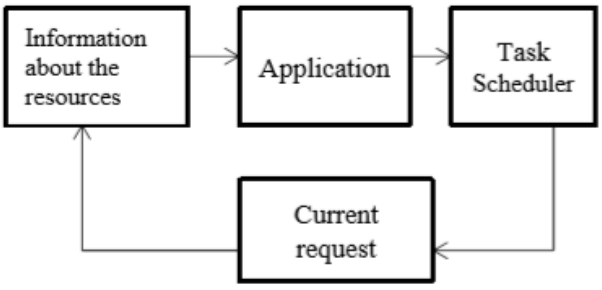


TABLE I COMPARISON OF STATIC LOAD BALANCING ALGORITHMS

Algorithm	Parameters	Challenges
Round robin	Waiting Time	Less Resource Utilization
Active clustering load balancing	Response Time	Every time iteration takes place for allocation for VMs
Central load balancing decision model	Throughput	Provides wrong decision when it entered into the loop.
Min-min load balancing	Waiting time	Starvation
Map reduced based entity resolution model	Resource Utilization	Requires higher computation time as it has been divided by Part() method.

B. DYNAMIC LOAD BALANCING

The Dynamic Load balancing algorithms are which processes the request of the user dynamically and accepts the current request from the client node. Dynamic Load balancing algorithms do the job in the execution time and they are faster. Work is allotted to host or a hub [15]. The work load that is a number of processes, data transfer capacity, etc. at every post is figured powerfully.



Dynamic Load balancing Algorithms are as follows:

1) Equally Spread Current Execution Algorithm

In this algorithm, the processes are handled with priority. The dynamic load adjusting calculation is performed and the priority for each process is allocated. Here the priority of the process is decided by checking the span or lifetime of the process. This algorithm sends out its load randomly by first checking the size of the processes and then the workload is exchanged with the virtual machine, which is lightly loaded. The spectrum technique is used where the load balancer spreads on a distinctive node [11].

2) Throttled Load Balancing Algorithm

In this algorithm, the load balancing concept is based on finding the correct virtual machine for dealing with a specific occupation. Here job manager has a rundown of all the virtual machines, using this ordered show it allocates the desired work given by the client to its fitting machine [10]. In this algorithm, if the client request is suited for a particular machine such as size and accessibility of the machine, the job is assigned to a particular machine. If there are no virtual devices to run the job then the job anger queues the request till an appropriate match is found [14].

3) Join-Idle Queue Load Balancing Algorithm

In this algorithm, it uses distributed dispatcher by the first load balancing the ideal processors across the dispatcher and then assigning jobs to the processes to reduce the queue length at every processor. This algorithm is mainly used for large scale systems. The main purpose of the load balancing algorithm is that it is used for dynamically scalable web services. Therefore, efficiently reduce the system load, incures communication overload at job arrivals and does not increase actual response time [8]. The disadvantage of this algorithm is it is not scalable.

4) CARTON Load Balancing Algorithm

This algorithm is basically divided into two parts that are first is load balancing and second is distributed rate limiting (DRL) [6]. The work of load balancing is to utilize the load and equally distribute the job between their nodes so that its related cost can be minimalized and the performance can be increased. The DRL is used in this algorithm for fair resource allocation. It additionally adjusts the server capabilities for dynamic workload which ensures that execution level at all the servers is equal. This algorithm is simple to execute with low calculation and correspondence overhead [16].

5) Honey Bee Foraging Load Balancing Algorithm

The idea behind the honey bee foraging load balancing algorithm is forging the behavior of honeybee. There are two kinds of honeybees finders and reapers. The finders go outside in search of the source. Once they find the source they come back to the honeycomb and gives a signal demonstrating the quality and the mound of honey available [1]. At that point, reapers go outside and harvest the honey from the source. After gathering the honey from the source they get back to the comb and do a waggle to demonstrate how much food is left. Similar to this the servers are embedded as virtual servers and each virtual servers has a process queue attach to it [4]. Every server calculates the benefits which are practically equivalent to the quality. If profit is high the server stays else it comes back to the forage. This node requires every node to keep up a different queue [6]. The calculation of profit on every node causes overhead. The calculation here does not demonstrate a huge change in throughput.

TABLE II COMPARISON OF DYNAMIC LOAD BALANCING ALGORITHMS

Algorithm	Parameters	Challenges
Equally Spread Current Execution	Throughput, Response	Time VM are assigned randomly
Throttled Load Balancing	Throughput, Resource Utilization	Requires maximum VM and completion time is high
Join-Idle Queue Load Balancing	Response Time	Less Resource utilization
CARTON Load Balancing	Distributed Rate Limiting	Less Resource Utilization
Honey Bee Foraging Load Balancing	Throughput	More time to allocate the particular VMs

III. BENEFITS OF LOAD BALANCING

1. Scalability: The main advantage of load balancing is that it is highly scalable meaning any number of servers or nodes can be added without damaging or causing a disturbance, and even though many nodes are added the performance is good and efficient through load balancing in the node or the server in the cloud [9].

2. Performance: Load balancing is used to increase the efficiency helps the users and the service providers to respond faster compared to the usual compilation time. Moreover, the execution time gets reduced to a greater extent through caching mechanism and efficient compression techniques [10].

3. Availability: Load balancing mechanism provides services efficiently. In case of unavailability of the servers, the load will be further distributed efficiently [1].

4. Reliability: The reliability of the cloud service is protected by the redundancy of the server through which application is hosted at the cloud hub in the world. When there is a failure, the service will not stop functioning and the service will be changed to another cloud location [16].

IV. ISSUES RELATED TO LOAD BALANCING

Even though the load balancing algorithm is mostly used to increase the performance by evenly distributing the load among the node there are some issues at the time of load balancing [13]. The issues related to load balancing are:

- Sometimes a fair distribution of the cloud resource does not exist that results in overutilization or underutilization of the nodes [10].
- When the user demands change, which will gradually decrease in the performance of the node.

V. CHALLENGES OF LOAD BALANCING ALGORITHM

Performance: When the load is equally divided between the node the performance of the nodes increases. This deals with better efficiency of the system. When the system performance is good it ensures satisfaction to the user [9]. The performance metrics are as follows:

1. Response Time: It is defined as the time taken to respond by the algorithm to the user. When response time is less it denotes good performance.
2. Resource Utilization: When the resources are utilized properly which ensures good performance by the system. This is used to measure if the node is under loaded or overloaded.

Fault Tolerance: A behavior that improves the system and when there is a chance of failure it should not affect the other system. The algorithm should be designed in such a way that even if one system stops working it should not affect another system [9].

Overhead: It is defined as the time taken to migrate from one virtual machine to another which increases in communication cost. A good load balancing show has less overhead [8].

VI. CONCLUSION

Load balancing algorithms are implemented to basically increase the efficiency and performance by equally distributing the load equally. The main objective of this paper is to give a brief description of load balancing that is about static and dynamic load balancing algorithms. Here a summarization of all the methodologies is addressed with respect to load balancing algorithm in cloud computing. In this paper, we have also discussed different challenges for the efficient load balancing algorithms and comparison of both static and dynamic algorithms.

REFERENCES

[1] Ariharan V,Sheeja Manakatlu “Neighbor Aware Random Sampling (NARS) algorithm for load balancing in Cloud computing” IEEE-2015.

[2] Surbhi Kapoor, Dr. Chetna Dabas “Cluster Based Load Balancing in Cloud Computing” IEEE 2015.

[3] Abhinav Hans, Sheetal Kalra ”Comparative Study of Different Cloud Computing Load Balancing Techniques” IEEE 2014.

[4] Reena Panwar, Prof. Dr. Bhawna Mallick “Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm” IEEE 2015.

[5] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and Kuwar Pratap Singh “Load Balancing of Nodes in Cloud Using Ant Colony Optimization” International Conference on Modelling and Simulation IEEE 2012.

[6] J.Prasanna, Ajit jadhav, and V.Neel Narayan “Towards an Analysis of Load Balancing Algorithms to Enhance Efficient Management of Cloud Data Centres," Springer, 2016.

[7] Aarti Singha, Dimple Junejab, “Manisha Malhotra “Autonomous Agent Based Load Balancing Algorithm in CloudComputing” Elsevier,2015.

[8] Tushar Desai, Jignesh Prajapati “A Survey Of Various Load Balancing Techniques And Challenges In Cloud Computing” IJSTR,2013

[9] Reena Panwar , Prof. Dr. Bhawna Mallick ”Load Balancing in Cloud Computing Using Dynamic Load Management Algorithm”IEEE 2015

[10] Sandeep Negi, “An Improved Round Robin Approach using Dynamic Time Quantum for Improving Average Waiting Time” International Journal of Computer Applications Vol. 69 : No 14, May 2013.

[11] Labdah ALGhafran and Zulkefli Bin Muhammed Yusof , “An Efficient Load Balancing Clustering Algorithm for Mobile Ad Hoc Networks” International Journal of Computing, Communications & Instrumentation Engg. (IJCCIE) Vol. 2, Issue 2 (2015) ISSN 2349-1469 EISSN 2349-1477.

[12] A.kumar “Load Balancing in Cloud Data Center Using Modified Active Monitoring Load Balancer” IEEE-2016.

[13] Shang-Liang Chen , Yun-Yao Chen ∗, Suang-Hong Kuo “ CLB: A novel load balancing architecture and algorithm for cloud services.

[14] https://docs.hpclIoud.com/eucalyptus/4.2.2/installguide/euca_components.html

[15] Garima Rastogi, Dr Rama Sushil, “Analytical Literature Survey on Existing Load Balancing Schemes in Cloud Computing”, IEEE 2015.

[16] Garima Gupta, Vimal Kr.Kumawat, P R Laxmi, Dharmendra Singh, Vinesh Jain,” A Simulation of Priority Based Earliest Deadline First Scheduling for Cloud Computing System”, IEEE 2014.