LOAD BALANCING IN CLOUD COMPUTING: APPLICATIONS & COMPLEXITIES ASSOCIATED WITH ALGORITHMS – A SURVEY

L. Thara[1], Dr.R. Gunasundari[2], G. Naveen[3]

ABSTRACT

The significance of cloud computing is massively increasing in the direction of outward utilization of information technology assets. The process of computational control, loading and industry requests as packages and handling social network data with optimal throughput is a challenging task. Cloud computing uses the ideas of scheduling and load balancing to transfer tasks to Virtual Machines for efficiently allocating the requests and effectively sharing the resources. Since it is a vital responsibility of the cloud computing, global results for refining the systems corresponding to the interval of accomplishment and budget are most essential. This paper presents the survey of various load balancing techniques along with their significance and complexities.

Keywords : load balancing, cloud computing, GA, VM, algorithm, efficiency.

[1]Assistant Professor, Dept. of Computer Science, PSG College of Arts & Science, E-mail : kltharavijay@gmail.com,
[2] Research Supervisor, Department of CS, CA & IT, Karpagam Academy of Higher Education,
E-mail : gunasoundar04@gmail.com,
[3] II MCA, Department of Computer Science, PSG College of Arts & Science, E-mail : nav041997@gmail.com
Coimbatore, Tamil Nadu, INDIA

I. INTRODUCTION

IT based companies have been changing their way to purchase and design hardware over cloud computing technology with great utility which can also create software more smart. Though, clouding technology is extensively implemented in the industrial arena, until now various ongoing disputes similar to load balancing, energy management, security and so on are existing. This correspondingly relates to the issue of VM migration. Among various issues, the major stimulating concern in a cloud environment is the load balancing policy. In turn, it is interconnected to unambiguous complications [5]. Figure 1 shows the structure of load balancer in a clouding environment.

Load balancer means eliminating tasks from the severely loaded Virtual Machines and allocating them to the Virtual Machines with less load. Several studies in the field of load balancing were conducted to boost the load across the VMs in cloud environment. As on-demand in service model, weight comes arbitrarily or dynamically in cloud computing settings, which prompts some virtual machines/servers to be loaded largely, while others idle or gradually loaded. This in turn contributes to pitiful performance and make user unconvinced [2].

Consequently the load balancer allocates the active amount of work through various nodes to confirm that no distinct node is either heavily utilized or less utilized. The condition may be measured as a problem of optimization. Hence there is a need of better balancing device which familiarizes its approach to the fluctuating situation and the sorts of responsibilities [3]. As a result, the usage and throughput will be optimized while cutting down the response time.

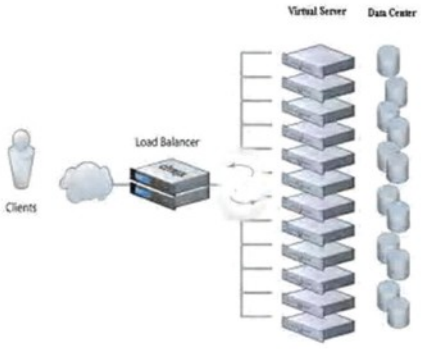

Figure 1 : Structure of load balancer in cloud environment

II. Types & Aspects of Balancer Algorithms

The concept of balancing the load can be explained with 2 primary algorithms: [6]

A. STATIC ALGORITHM

In this algorithm the circulation is separated consistently between the masters (serving machines). There is a need of prerequisite properties which will assist in performing the task of shifting the loads across nodes which is not subject to the present form of the scheme. Static algorithm is suitable for circumstances when the load fluctuation is very minimal.

B. DYNAMIC ALGORITHM

In this algorithm either the server allocated with minimal load among the entire system or a suitable system is examined and desired as a load balancer. To accomplish this task, an interaction with the network is essential which may worsen the network traffic of the system. This approach uses the present form of the system to take decision for managing the load.

C. ASPECTS

- ***Throughput:*** The aggregating process time for it execution in specific volume of time.
- ***Makespan:***A complete processing time when will the job arriving and destroying.
- ***Response Time:*** The task taken time for its arriving in the system. Resource
- ***Utilization:*** The proper way of assigning in the resource for performing.
- ***Overhead:*** Interlink between the different neighborhoods nodes on its task movement.
- ***Fault Tolerant:*** Avoid task replication in any other device and find disconnected node.

  Migration Time: The seeking time for moving one node to another node.

III. Genetic Algorithm

This algorithm is a kind of searching algorithm, which provides an ideal outcome to a problem by searching a solution space [5]. The special feature of this algorithm is that the way it conducts the searching process. First, a "population" of probable results are generated for the specific problem by the genetic algorithm. Then they are allowed to "evolve" with numerous generations in order to attain optimal results.

A. GA Functions:

The functionality of GA is accomplished in 4 major functions as described below:

1. Size of the Population: This function constructs 2 lists of arrays. The first array comprises the list of VM's using which jobs are to be arranged to implement and the second array encompasses list of requests to be performed.

2.Objective Function: In this function, all probable VM's will be fetched, so as to accomplish the task.

3. Mutation Function: This step allocates job to the device queue in correspondence to an estimation needed by the jobs and the capability of devices to perform the job. In other words, this function discovers an optimal VM to execute the relevant job.

4. Fitness Function: This step consequently invokes the appropriate VM which fulfills the condition by executing the mutation function as soon as the job is to be completed [5].

Garima Joshi and Verma .S.K have proposed an algorithm which considered the value of cost as fitness function of a distinct node whereas carrying out load balancing and the approach was computer-generated by means of MATLAB [2]. They came out with an improvised Genetic Algorithm Load-Balancing issue, that can be worked out by allocating X number of jobs to Y amount of elements to be executed on cloud. Every processing unit will have a processing unit vector *(Puv)* presenting the importance of processing unit consumption.

Similarly, a job unit vector *(Juv)* represents each job submitted by cloud user. Thus, the attribute of different situation can be represented by $Juv = f(Ts, N, \alpha,)$ where Ts, represents the nature of service needed by the job, Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS) [4].

B. GA Applications

GA aims to stabilize the load of the portable cloud organization even though trying to diminish the time of execution or approachability of jobs with abridged amount of relocations of VM's and develops the resource consumption by sharing the capability of a data centre into n amount of VM's completing the number of demands in parallel with increased efficiency. The GA based methods were used for balancing the capacity of the framework and the proposed system is programmed to map the appropriate VMs to carry out the desired demands with condensed response time. This load balancer has been developed with the cloud sim emulator.

The results of simulation show that the implemented algorithm provides efficient solution with reduced timeslot that consumed through the process of restoration and suspension [5].

Subasish Mohapatra, proposed a GA based load balancer (LBGA) with real assessment implied GA using a novel programming mechanism, in his research, "On Solving Some Issues in Cloud Computing"[7]. Genetic Algorithm works with a population of the probable keys characterized in the form of chromosomes. All chromosomes are composed of variables known as genes and they map to a fitness value of the candidate problem. Jobs come at dissimilar breaks to be processed and are located in the queue for execution. Makespan is considered as the head objective function of the proposed algorithm. The calculating node is denoted as a VM. Though, a distinct node may be allocated with many VMs as dedicated calculating components. High average node consumption guarantees that the load is well balanced through the entire nodes. Typical deployment is accomplished by sharing the job execution times of every VM by the Makespan value. The consumption of the distinct VM (UMi) can be given by:

$$UMi = Ti / Makespan$$

where, Ti is the task completion time.

GA has been proposed as a load balancing technique for cloud computing by Kousik Dasgupta et al. to find a global optimum processor for job in a cloud. A simple GA is composed of 3 procedures: selection, genetic operation and replacement. The benefit of this technique is that it can manage a huge search space, appropriate to compound objective function and can avoid being trapped into native optimal result. The process of GA is repeated till either the fittest chromosome (optimal solution) is found or the termination condition (maximum number of iteration) is exceeded [3].

Genetic Algorithms are mainly pertinent to large problems which are nonlinear, and perhaps isolated in nature. As the probabilistic progress of the solution, GA cannot ensure an ideal solution. Still, they are expected to be close to the universal optimum [7].

IV. Honey Bee Behavior Inspired Algorithm

L.D.Dhinesh Babu, et al. [1], considers an algorithm known as "Honey Bee Behavior - Inspired Load Balancing" that works well by dispensing the load correspondingly across virtual machines. This algorithm accomplishes load balancing so effectively that it decreases the sum of waiting time of jobs in the queue.

A. Honey Bee Algorithm Functions

It is entirely stimulated by the hunting behaviour of honey bees. As soon as any overloaded VM is established, then the task is to be removed to the under loaded VM to balance the load. It depends upon two circumstances i.e., either it discovers the VM or it might not find the appropriate machine. When it finds a proper VM, it generates a positive signal and generates a negative signal when it fails. Moreover the job has to discover the best virtual

machine when it finds a group of similar virtual machines. This is accomplished using QoS standards termed as job precedence. This means that the job identifies machine allocated with fewer jobs of similar precedence. Once the task identifies the suitable machine, it is assigned to the corresponding machine then the information are loaded into it.

Suppose the job fails to identify an appropriate virtual machine, it drives to the waiting state where it gets experienced with the information updates by other tasks. The procedure of finding the proper VM for the waiting task starts again once it confirms the information and competes with other challenging jobs to discover their desirable virtual machines and become distributed to them.

The data is restructured after the job is being distributed to the suitable VM. As the new tasks arrive, the process of finding the suitable VM for each task to balance the load continues till all the tasks are assigned to appropriate machines and the method is properly balanced based on the capacity as well as preferences.

B. Pros and Cons of Honey Bee Algorithm

PROS	*CONS*
Low Overhead	*Fewer throughputs*
Less migration time	*Less Fault tolerant*
Less Response time	*Less Scalable*
Optimum Resource utilization	*Low priority task stay continuously in the queue*

V. Conclusion

This paper presents a study on how loads are balanced in cloud environments, deployment of various algorithms to rectify the complexities and their applications. The vital functionality of balancing the load is to fulfil the client prerequisite by allocating the load dynamically amongst the nodes and to create determined resource deployment by reallocating the entire load to discrete nodes. Thus the process ensures efficient resource distribution and increased performance.

VI. References

1. DhineshBabu L.D., P. Venkata Krishna, Honey Bee Behavior Inspired Load Balancing Of Tasks In Cloud Computing Environments, Applied Soft Computing Journal, Vol. 13, 2013.

2. Garima Joshi and Verma .S.K, "*Load Balancing Approach in Cloud Computing using Improvised Genetic Algorithm: A Soft Computing Approach*", International Journal of Computer Applications (0975 – 8887) Volume 122 – No.9, July 2015.

3. KousikDasgupta, BrototiMandal, ParamarthaDutta, Jyotsna Kumar Mondal, Santanu Dam, "*A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing*", International Conference on Computational Intelligence: Modeli=ng Techniques and Applications, 2013.

4. Lars, K., Andreas,T., Erhard,R., "*Load Balancing for Map Reduce-based Entity Resolution*", in Proc. 28th International Conference on Data Engineering (ICDE), IEEE, pages 618-629, 2012.

5. Neha Gupta, Parminder Singh, "*Load Balancing Using Genetic Algorithm in Mobile Cloud Computing*", International Journal of Innovations in Engineering and Technology, Vol. 4 Issue 1 June 2014.

6. RadhikaPathak, "*A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms*", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 5, Issue 1, January 2017.

7. SubasishMohapatra, Department of Computer Science and Engineering National Institute of Technology Rourkela Rourkela, Odisha, "*On Solving Some Issues in Cloud Computing*", dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering, November 2015.