Evaluate the Performance of Load Balancing Algorithms in Cloud Computing

Jyoti Talwar
Research Scholar
Computer Science and Engineering Department
RIMT-IET, Mandi Gobindgarh

Rupinder Kaur Gurm
Assisstant Professor
Computer Science and Engineering Department
RIMT-IET, Mandi Gobindgarh

Anuj K. Gupta
Professor
Computer Science and Engineering Department
CGC-Landran, Chandigarh

Abstract - Cloud computing is a collection of IT services that have been provided to a person on the network on a leased basis i.e pay-per –use in nature, same as we pay mobile and electricity bills as per our use. For maintaining various data and applications, this technology uses the internet and central remote servers. In cloud computing, Load balancing is the main challenge in which distribution of the dynamic workload across multiple nodes is required so that no single node is overloaded. It helps in proper utilization of resources and therefore enhancing the performance of the system. To assign the client's requests to various Cloud nodes, many algorithms were designed. These approaches aim to improve the overall performance of the cloud and provide the user more satisfying services. This paper will analysis the performance of proposed algorithm and compares the simulation outcomes in terms of throughput, migration time and overhead associated , with the existing approaches i.e. Ant Colony Optimization(ACO), Equally Spread Current Execution(ESCE) and Round Robin(RR).

Keywords- Cloud Computing, Load balancing, Artificial Bee Colony(ABC)Optimization, Hybridized Ant Colony and Bee Colony(HACBC) Optimization.

I. INTRODUCTION

For maintaining various data and applications, Cloud Computing uses the internet and central remote servers. Cloud Computing provides various services to a person on the network on a leased basis i.e pay-per –use in nature, same as we pay mobile and electricity bills as per our use. Cloud Computing is mainly dynamic in nature. Hence, prediction based monitoring is impossible and performance analysis of any application in cloud computing is very much important as well as complex. As fast growing IT market, mostly companies are not able to manage these IT requirements even though they have in-house data centres. Hence, the services of cloud manage these resources efficiently and provide the way to enhance IT facilities without paying for new data centres. By providing centralized storage, bandwidth, memory and processing, this technology helps companies very efficiently computing. The IT resources may consist of various different nature computational web services for example- tax-calculation web service, weather information web service, shipping position web service etc. These web services are reusable and also programmable. With the help of internet these services are available anywhere. This model is composed of five main characteristics, three service models, and four deployment models.

The five main characteristics are as follows.

- Pooling of resources
- On-demand self service
- Universal network access
- Site independence

- Fleet adaptability.

II. ARCHITECTURE OF CLOUD COMPUTING

Cloud computing is rapidly growing technology in the real time environment. It provides various services models and deployment models which are discussed as follows. Figure 1 describes the three main service layers that aggregate the cloud computing. It presents three basic services that are Software as a Service, Platform as a Service and Infrastructure as a Service [2]. The rest of the paper is organized as follows. In section 3, we defined load balancing and its need and challenges.

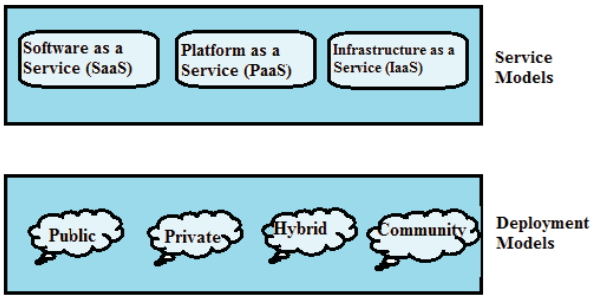

Figure1: Architecture of Cloud Computing

The services models are defined as follows.

I. Software as a Service (SaaS): It use application of provider over a network.
II. Platform as a Service (PaaS): It set up applications created by customers to a cloud.
III. Infrastructure as a Service (IaaS): Rent and lease preparing, storage capacity, network capacity and other fundamental computing resources.

The deployment models are as follows.

I. Private Cloud: Cloud that is owned or leased by an enterprise.
II. Community Cloud: Cloud which shared infrastructure for some specific community.
III. Public Cloud: Cloud that is sold to the public, mega-scale infrastructure.
IV. Hybrid Cloud: Cloud which is combination of two or more Clouds.

III. LOAD BALANCING

Load balancing is one of the main issues related to cloud computing. The load can be a memory, CPU capacity, network or delay load. It is always required to share work load among the various nodes of the distributed system to improve the resource utilization and for better performance of the system. This can help to avoid the situation where nodes are either heavily loaded or under loaded in the network. Load balancing is the process of ensuring the evenly distribution of work load on the pool of system node or processor so that without disturbing, the running task is completed. The goals of load balancing [9] are to:

- Improve the performance
- Maintain system stability
- Build fault tolerance system
- Accommodate future modification.

There are mainly two types of load balancing algorithms:

a. *Static Algorithm*

In static algorithm the traffic is divided evenly among the servers. This algorithm requires a prior knowledge of system resources, so that the decision of shifting of the load does not depend on the current state of system. Static algorithm is proper in the system which has low variation in load.

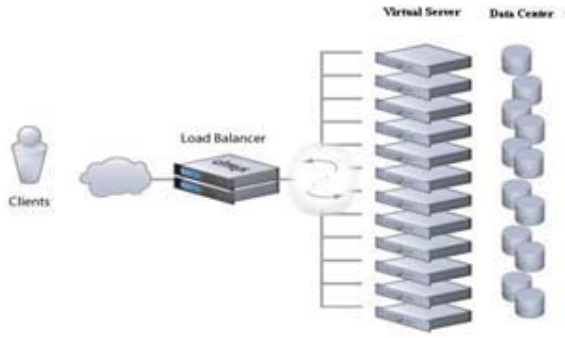

Figure 2: Load Balancing in Cloud Computing [3]

b. Dynamic Algorithm

In dynamic algorithm the lightest server in the whole network or system is searched and preferred for balancing a load. For this real time communication with network is needed which can increase the traffic in the system. Here current state of the system is used to make decisions to manage the load.

3.1 NEED OF LOAD BALANCING IN CLOUD COMPUTING

Load balancing in clouds is a mechanism that distributes the excess dynamic local workload evenly across all the nodes. It is used to achieve a high user satisfaction and resource utilization ratio [10], making sure that no single node is overwhelmed, hence improving the overall performance of the system. Proper load balancing can help in utilizing the available resources optimally, thereby minimizing the resource consumption. It also helps in implementing fail-over, enabling scalability, avoiding bottlenecks and over-provisioning, reducing response time etc.

Apart from the above-mentioned factors, load balancing is also required to reduce the energy consumption and environmental impact of their utilization in clouds which can be done with the help of the following two factors:

• *Reducing Energy Consumption* - Load balancing helps in avoiding overheating by balancing the workload across all the nodes of a cloud, hence reducing the amount of energy consumed.

• *Reducing Carbon Emission* - Energy consumption and carbon emission go hand in hand. The more the energy consumed, higher is the carbon footprint. As the energy consumption is reduced with the help of Load balancing, so is the carbon emission helping in achieving Green computing.

3.2 CHALLENGES IN CLOUD COMPUTING LOAD BALANCING

Before we could review the current load balancing approaches for Cloud Computing, we need to identify the main issues and challenges involved and that could affect how the algorithm would perform. Here we discuss the challenges to be addressed when attempting to propose an optimal solution to the issue of load balancing in Cloud Computing. These challenges are summarized in the following points:

1. *Spatial Distribution of the Cloud Nodes*

Some algorithms are designed to be efficient only for an intranet or closely located nodes where communication delays are negligible. However, it is a challenge to design a load balancing algorithm that can work for distributed nodes. This is because other factors must be taken into account such as the speed of the network links among the nodes, the distance between the client and the task processing nodes, and the distances between the nodes involved in providing the service. There is a need to develop a way to control load balancing mechanism among all the spatial distributed nodes while being able to effectively tolerate high delays[5].

2. *Storage / Replication*

A full replication algorithm does not take efficient storage utilization into account. This is because the same data will be stored in all replication nodes. Full replication algorithms impose higher cost since more storage is needed. However, partial replication algorithms could save parts of the datasets in each node (with a certain level of overlap) based on each node's capabilities such as processing power and capacity [6]. This could lead to better utilization, yet it increases the complexity of the load balancing algorithms as they attempt to take into account the availability of the dataset's parts across the different Cloud nodes.

3. *Algorithm Complexity*

Load balancing algorithms are preferred to be less complex in terms of implementation and operations. The higher implementation complexity would lead to a more complex process which could cause some negative performance issues. Furthermore, when the algorithms require more information and higher communication for monitoring and control, delays would cause more problems and the efficiency will drop. Therefore, load balancing algorithms must be designed in the simplest possible forms [7].

4. Point of Failure

Controlling the load balancing and collecting data about the different nodes must be designed in a way that avoids having a single point of failure in the algorithm. Some algorithms (centralized algorithms) can provide efficient and effective mechanisms for solving the load balancing in a certain pattern. However, they have the issue of one controller for the whole system. In such cases, if the controller fails, then the whole system would fail. Any Load balancing algorithm must be designed in order to overcome this challenge [8]. Distributed load balancing algorithms seem to provide a better approach, yet they are much more complex and require more coordination and control to function correctly.

IV. RELATED WORKS

Shanti Swaroop Moharana,Rajadeepan D. Ramesh & Digamber Powar.(May2013) –"Analysis of Load Balancers in Cloud Computing"depicts that evolutionary algorithms are slower in nature for finding the optimal solutions since there is a need of handling the population movements[11].

Pandey S, Wu L, Guru S, Buyya R.(2010) –"A Particle Swarm Optimization (PSO) based heuristic for scheduling workflow applications in Cloud Computing environments." which is being inspired by the movement of birds in flocks or fishes in school. In PSO search network, each solution is considered as "particle". Each iteration updates the particle by considering the two "best" values i.e pbest and gbest and these two values are need not be evaluated in ACO algorithm. Unlike PSO technique, ACO reaches with guaranteed convergence to the optimal solution. Moreover ACO is more applicable for problems that require crisp results and PSO is applicable to problems that are fuzzy in nature[12].

Tanya Prashar & Rajeev Kumar (2015)- "Performance Analysis of Load Balancing Algorithms in Cloud Computing" analysis the comparison of Load Balancing Algorithms –RR, ACO, ESCE, BCO using average response time, total cost, average DC processing time[13].

Daniel Grosu, Anthony T. Chronopoulos et. al. (2005)-"Non-cooperative load balancing in distributed systems" proposed a novel distributed load balancing technique. The major benefits of the proposed load balancing technique are the distributed structure, less complexity and optimal allocation of virtual machines for each user request[14].

M. Houle, A. Symnovis and D.Wood,(June2002)- "Dimension-exchange algorithms for load balancing on trees" designed algorithms for statically balancing the load on trees, considering that the total load is fixed[15].

A. Y .Zomaya &Y. H.Teh.(2001) – "Observations on using genetic algorithms for dynamic load-balancing" states that there are many examples of evolutionary algorithms, where genetic algorithm is one of them that is used for scheduling in a network. These algorithms may comprise a memory to retain the last status that helps in reducing the no of agents close to locations in optimal solutions that have been discovered earlier[16].

Y. Hu, R. Blake and D. Emerson,(1998)-"An Optimal Migration Algorithm for Dynamic Load Balancing" proposes an efficient algorithm for data migration in dynamic load balancing by calculating the Lagrange multiplier associated with the Euclidean form of transferred weight. This work can minimize the data movement in homogenous environments in an efficient manner, but it does not support the distributed heterogeneous environments [17].

V. PROBLEM FORMULATION

The problem of load balancing occurs when the cloud clients try to access and send the request to the same cloud server while other cloud servers don't receive the service request from the cloud clients which leads to the unbalanced workload on the cloud data centers. Therefore, it causes the development of numerous algorithms for scheduling and load balancing. But unlike swarm intelligent algorithm, one of the classes of evolutionary strategies like genetic algorithm does not support multiple users at an instant of time. Many authors have just focused on the availability of nodes and only few factors are taken into consideration like node's memory, processing capacity, etc. Thus we added the some more factors like virtual machine bandwidth, virtual machine computing capacity which is being calculated in respect of millions of instructions per second, no of processors in a virtual machine and image size of a virtual machine therefore all these factors will easily provide the fittest resource for the job to be processed in a cloud. In some research papers of ABC, FCFS priority concept has been considered that could minimize the overhead associated and to minimize the migration time. Hence Shortest Job

First Criteria has been considered to maximum throughput of the cloudlets. In our approach the ant net concept of forward and backward movement is also taken into account for distributing the workload on the nodes, whereas ABC is applied for searching the optimal path towards the best suitable resource in the cloud network.

VI. EXISTING WORK

Existing Algorithms of Load Balancing are as follows:

i. Round Robin (RR) Algorithm-

The round robin algorithm in the cloud computing is quite similar as the round robin scheduling performed in the process scheduling. This algorithm performs on basis of random choice of the VMs. The data center controller (DCC) allots the service calls to a pool of VMs in a cyclic manner.

The initial client request is assigned to a randomly selected VM from the group of VMs and then the DCC allots the requests in a round manner. Once the VM is allocated, it is moved to the bottom of the pool of VMs [13].

ii. Equally Spread Current Execution Load (ESCE)-

The algorithm of ESCE needs a load balancer that tracks the jobs which are asked for execution. The main requirement of Load balancer is to put the tasks in the job pool and assign them to the distinct VMs. The balancer keeps monitoring the job queue frequently for new tasks and then assign them to the pool of free VMs. The load balancer also handles the list of tasks assigned to the virtual servers, which helps them to check which VMs are free and need to be allocated to the new tasks. The name itself clearly defines about this algorithm that it works with equally distributing the work load on distinct virtual machines [13].

iii. Ant Colony Optimization (ACO)-

ACO arises from the way real ants naturally behave. Initially, real ants move randomly in search of food and upon finding the food resource it returns to their colony while laying down pheromones on its path. If new ants also discover such pheromone concentrated track, they will also follow the pheromone trails instead of wandering randomly, return and reinforce it, if they ultimately search the food resource. When one ant encounters a shorter distance from the ant colony to the food resource, other ants also follow that length due to bio-inspired nature of ants, thus produces a positive feedback i.e. finally makes all the ants to follow a single track. From an algorithmic point of view, the pheromone evaporation process is useful to avoid the convergence for a local optimal solution [13].

Primary algorithm of ACO is given as:

Step1: Initialization of ants with pheromones.
Step2: Locating the ants
Step3: Selection of next state.
Step4: Checking of Load Balance.
Step5: Pheromone Updation step.
Step6: If stopping criteria is met, then stop the execution, else repeat from step 2nd.

iv. Bee Colony Optimization (BCO)-

The algorithm of artificial bee colony (ABC) is confined to the activities of honey bees for searching the nectar as well as for sharing the information with other bees. In this algorithm, there are generally three types of bees present, i.e. onlooker bees, scouts and employed bees. The employed bees settle down on the food resource and retain its surroundings in the memory; while onlooker stake this information from the employed bees and choose the food resource accordingly. On the other hand the scouts are responsible for discovering the new food resources. The main constituent of the beehive is the dancing area where information is being shared among the bees. This information is related to the location and quality of food resources. This dance is known as “the waggle dance” [13].

Basic algorithm of Bee Colony is given as:
Step1: Initialization of bee population with their random solutions.
Step2: Evaluate the fitness function and recruit employed bees.
Step3: Calculate the fitness function and recruit the onlookers
Step 4: Move the Scout bees.
Step5: Evaluate the optimal solution.
Step6: Check stopping condition, if met, then end the execution, otherwise repeats from 2nd step.

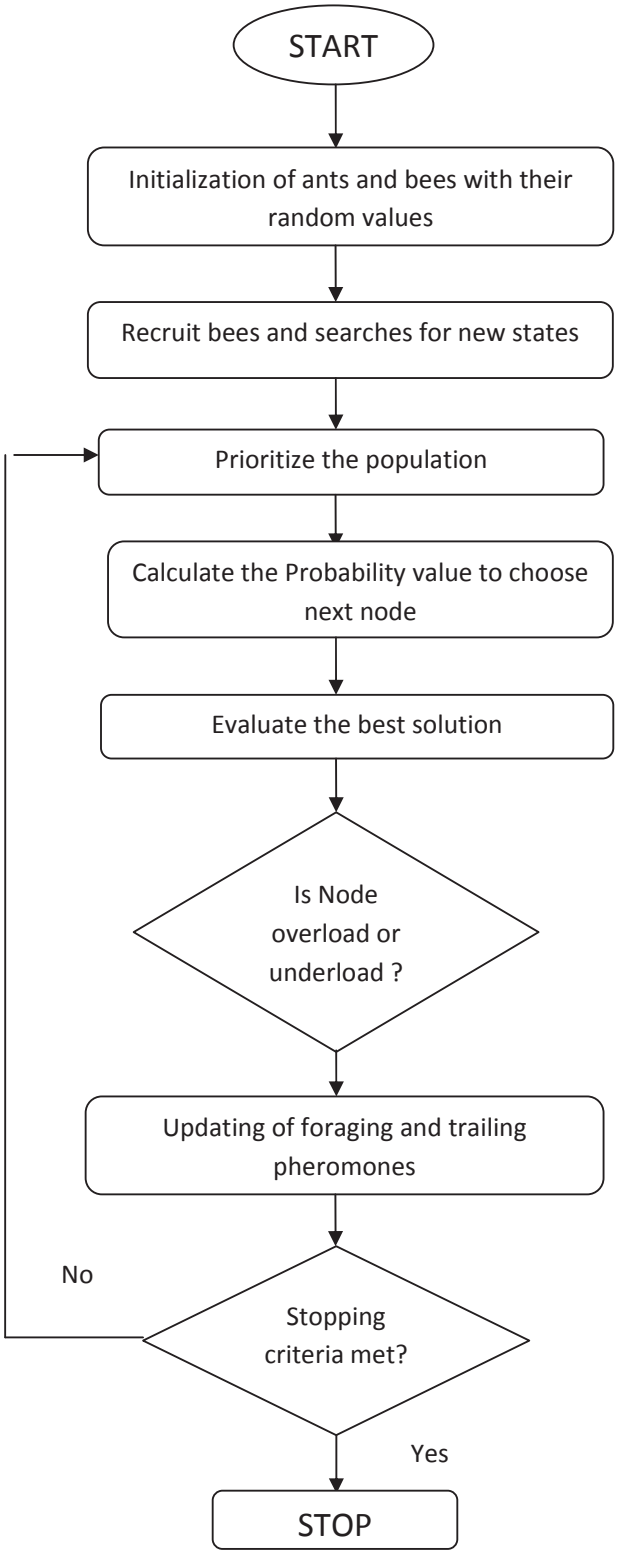

Figure 3: Flowchart of Proposed methodology

VII. OBJECTIVES OF THE STUDY

The main objectives of the study are as follows:

1. To make efficient scheduling and uniform distribution of workload among cloud virtual resources at an excellent performance rate.
2. This will work out on two basic swarm intelligent meta-heuristic algorithms, including ant colony optimization and priority based ABC to design and produce an efficient load balancing algorithm within a distributed heterogeneous environment known as cloud.
3. There are no. of physical servers known as data centres having multiple virtual machines. The proposed algorithm is decentralized to elude bottlenecks and a single failure point.
4. The performance will be evaluated using the parameters throughput, migration time, overhead associated.

VIII. METHODOLOGY

The methodology is as follows:

1. The initialization of the parameters following with the construction of ant solution using Cloud sim and Cloud Analyst simulator.
2. Evaluating the fitness function of bees.
3. Calculating the probability value.
4. Scheduling of the tasks to the resources.
5. Balancing of workload is carried out using the pheromone updating technique of the hybrid algorithm.

IX. CONCLUSION

This paper described load balancing approaches of Cloud Computing and challenges in Cloud Computing Load Balancing. It also reviewed various load balancing algorithms. The proposed algorithms were based on one or more approaches. It was observed that there are no such algorithm which can hold all the metrics of Load Balancing. Each and every algorithm was designed to achieve specific objective. Therefore, such algorithm must be designed which can handle different types of workload and suitable for all types of environments.

REFERENCES

[1] Basic concept and terminology of cloud computing -http://whatiscloud.com
[2] L.Wang, J.Tao ,M.Kunze, "Scientific Cloud Computing: Early Definition and Experience", the 10th IEEE International Conference Computing and Communications 2008
[3] Nidhi Jain Kansal, Inderveer Chana "Cloud Load Balancing Techniques : A Step Towards Green Computing " IJCSI -International Journal of Computer Science Issues, Vol. 9, Issue 1, No 1, January 2012
[4] Load Balancing in Cloud computing , http://community.citrix.com/display/cdn/Load+Balancing
[5] Buyya R., R.Ranjan and R N. Calheiros, "Inter Cloud: Utility-oriented federation of cloud computing environments for scaling of application services," inproc. 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP), Busan, SouthKorea, 2010.
[6] Foster, I., Y.Zhao, I. Raicuand S. Lu, "Cloud Computing and Grid Computing 360-degree compared," inproc. Grid Computing Environments Workshop, pp :99-106, 2008.
[7] Grosu,D., A. T. Chronopoulos and M.Leung, "Cooperative load balancing in distributed systems," in Concurrency and Computation: Practice and Experience,Vol. 20,No.16,pp: 1953-1976,2008.
[8] Ranjan, R., L.Zhao, X.Wu ,A.Liu,A. Quiroz and M.Parashar, "Peer-to-peer cloud provisioning: Service discovery and load-balancing," in Cloud Computing - Principles, Systems and Applications, pp:195-217, 2010.
[9] D. Escalnte, Andrew J. Korty, "Cloud Services: Policy and Assessment" , Educause review July /August 2011
[10] 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, May 2010, pages 240-243.
[11] Shanti Swaroop Moharana, Rajadeepan D. Ramesh & Digamber Powar.(May2013) "Analysis of Load Balancers in Cloud Computing"
[12] Pandey S, Wu L, Guru S, Buyya R.(2010) –"A Particle Swarm Optimization (PSO) based heuristic for scheduling workflow applications in Cloud Computing environments."
[13] Tanya Prashar & Rajeev Kumar (2015)- "Performance Analysis of Load Balancing Algorithms in Cloud Computing"
[14] Daniel Grosu, Anthony T. Chronopoulos et. al. (2005)-"Non-cooperative load balancing in distributed systems"
[15] M. Houle, A. Symnovis and D.Wood,(June2002)- "Dimension-exchange algorithms for load balancing on trees"
[16] A. Y .Zomaya &Y. H.Teh.(2001) – "Observations on using genetic algorithms for dynamic load-balancing"
[17] Y. Hu, R. Blake and D. Emerson,(1998)-"An Optimal Migration Algorithm for Dynamic Load Balancing"