A Modified Load Balancing - Ant Colony Optimization (MLB-ACO) algorithm in Cloud Environment

R. Raja
Associate Professor, Department of Computer Science and Engineering,
St. Martin's Engineering College,
Secunderabad, Telengana, India.

S.A.Kalaiselvan
Professor, Department of Computer Science and Engineering,
St. Martin's Engineering College, Secunderabad,
Telengana, India.

ABSTRACT

Cloud computing is a platform that delivers services based on requisition to the customer at any time. Someone wants to leverage these tools to reduce the costs of storage and technology, and the demand on the cloud is continuing to rise. Load balancing is one of the most important problems that cloud networking has experienced for quite a while now. Congruous load balancing will minimize energy utilization and carbon ejection. There are a lot of load balancing algorithms available. Every of such algorithms operate in a variety of ways, including certain positives and demerit points. This paper is focused on dynamic load management for a cloud system and offers a basic description of load balancing and dynamic load management capabilities. In the cloud, we can find an almost optimal solution within a short period of time. Modified Load Balancing - Ant Colony Optimization (MLB-ACO) algorithm is considered to have an optimal load balancing solution in a cloud computing environment. Experimental outcomes exhibit that proposed model exceeds existing models in terms of transmission delay, execution time, reducing energy consumption, increasing resource utilization and decreasing the number of energetic nodes.

Keywords— Cloud Computing, Load Balancing, Dynamic load management, MLB-ACO.

1. INTRODUCTION

Every individual, company uses cloud computing services with the ease of access to the internet. Cloud Computing is the most exciting way to make the concept of "computer services" real, it allows you to store and restore enormous data without thinking about the hardware you use. It offers three services such as Software as Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [1]. Users on request are provided with different physical and virtual resources. Cloud computing requires virtualization to access the resource [2].Virtualization ensures that several operating systems can be implemented on one physical system and the underlying infrastructure can be shared. Virtualization is the abstraction of computer resources. Virtualization offers services linked to real systems. We may produce more insanely large amounts of less regulated servers by virtualization, which in effect decreases the total cost of capacity, electricity and infrastructure [3]. The virtualization platform can be used to quickly scale cloud resources. On-demand users are assigned dynamic cloud resources. With increasing users, the resources available are dynamically decreasing.

The allocation of cloud services on request to users contributes to the challenge of load balancing is defined in figure 1.If workload is not correctly spread, some cloud nodes are highly loaded and some nodes are lowly loaded. Likewise, if cloud resources are not properly distributed, the delivery of service to customers will be delayed [4]. Load imbalance can cause system bottleneck. In order to achieve resource utilization and without delay in the provision of services, the allocation of resources should be carried out in an efficient manner.

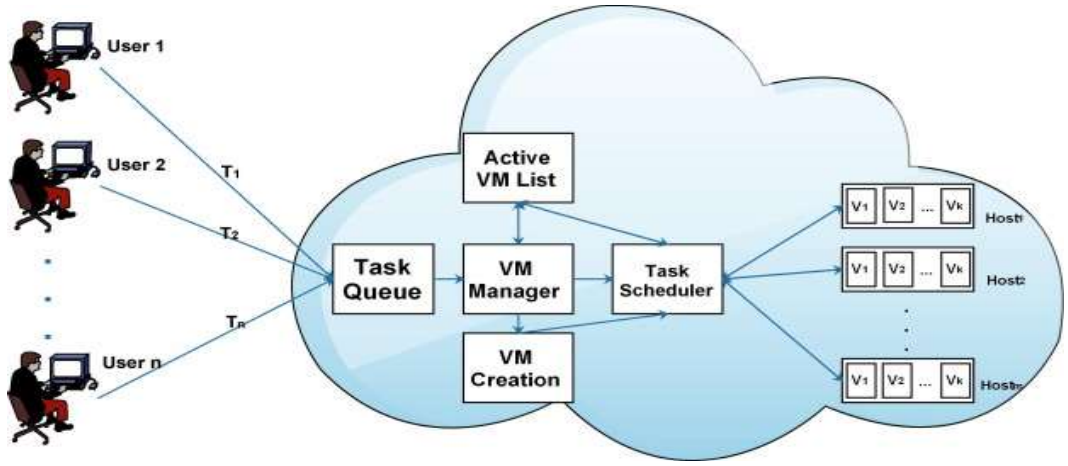

Figure.1 Load Balancing in Cloud Computing Environment

Nodes can logically be divided into clusters and load balance tasks are shared between clusters. Every single cluster will assign the load to the nodes of the cluster. This can be hierarchically arranged. Different load balance methods have been introduced effectively to spread load across the available machines in the cloud environment.

This article is structured as follows. Section II explains load management. Section III defines the different category of Load Balancing Algorithm. Section IV describes traditional Ant Colony Optimization. Section V includes the proposed load balancing algorithm called Modified Load Balancing – Ant Colony Optimization (MLB-ACO). Section VI and VII describes simulation and experimental analysis.

2. RELATED WORK

A. LOAD BALANCING

Load balancing is a way to distribute task effectively and equally over all functional nodes in the cloud. Through switching the workloads between various nodes will increases the overall system efficiency. Sometimes not properly used resources are overheated, causing carbon emissions. Carbon ejection can be reduced by utilizing energy properly [5]. Throughput, performance, scalability, reaction time, resource consumption and fault tolerance are several calculation indicators which can be used to assess load balancing strategies. These parameters help one to check if the basic load balancing technique is adequate for load balancing [6]. Growing virtual machine in the cloud network can run in the same quantity through a good load balancing process. Load balancing is therefore necessary to optimize efficiency by minimizing response time. It also reduces energy use in a healthy and balanced climate. The energy consumption is minimized with the aid of Load Balancing, thereby reducing carbon emissions. It aims to achieve green computing. Efficient load balancing would ensure efficient load management on nodes, improved overall system performance, enhanced user interaction, quicker response, reliability of the system, reduced carbon emissions.

B. CLASSIFICATION OF LOAD BALANCING ALGORITHMS

Basically load balancing algorithms can be categorized into two separate sections called static load balancing algorithms and dynamic load balancing algorithms**.**

C. Static Load Balancing Algorithms

When assigning tasks to nodes in a static load balancing algorithm, node status and feature in previous tasks will not be checked [7]. The assignment process is based solely on the system's previous knowledge of the property and resources of the node such as processing power, storage capacity and memory access. Although a node's above mentioned properties are considered before assigning a mission, they cannot adjust during runtime to the dynamic changes in the attributes and the allocated load on the node [8]. Several Generic Algorithms include the Round Robin Load Balancing Algorithm (RR), the Min-Min Load Balancing Algorithm (LB Min-Min), the Min-Max Load Balancing Algorithm (LB Min-Max).

D. Dynamic Load Balancing Algorithms

For heterogeneous and complex environments, dynamic algorithms have better results. These are more flexible algorithms. Dynamic algorithms can take into account dynamic attribute changes. The main advantage is the tasks are chosen based on current status that will lead to improving the system's performance [9]. The complex Load Balancing algorithm is distributed and non-distributed in two respects. All nodes in the system interact in a distributed algorithm and the task is divided among nodes, but all of them operate independently to perform the task in a non-distributed algorithm. Some Dynamic Algorithms are Throttled Load Balancer Algorithm, Biased Random Sampling Algorithm, Exponential Smooth Forecast based on Weighted Least Connection, Join Idle Queue Algorithm.

E. ANT COLONY OPTIMIZATION

ACO is inspired from the ant colonies that employment together in foraging behavior. Actually the real ants have inspired many researchers for their work [10, 12], and therefore, the ants approach has been employed by many researchers for problem solving in various areas. This approach is called for by the name of its inspiration ACO. Ants work together to find new food sources and at the same time use existing food sources to move the food back to the nest.

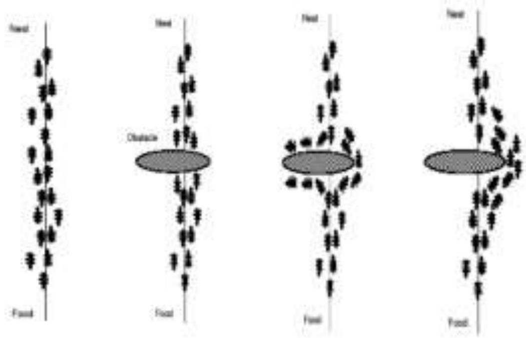

Figure. 2 Natural Behavior of ANT

For many years the ethologists were in trouble because they questioned how even a blind ant was able to follow their fellow ants and entered the food sources exactly. They find that on moving from one node to another, the ants leave a pheromone trail. The ant eventually arrived at the sources of food by tracking the pheromone tracks. The pheromone strength may differ based on various considerations such as nature of the food supply, distance from the food, etc. To pick the next node the ants use these pheromone tracks. The ants may even alter their paths when they encounter any obstacles in their way. Figure.2, gives a brief description of this situation. This behavior of the ants was used in various optimization algorithms, in which the ants obey one another by a pheromone network. The ants traversing from one node to another change the path's pheromone trail, so that a path will become more feasible if more ants traverse on it. Paths with the highest strength of pheromones have the shortest distance from the point to the best source of food. The behavior of those ants alters a solution set separately.

3. PROPOSED ALGORITHM - *Modified Load Balancing – Ant Colony Optimization (MLB-ACO)*

The hardware performance of the node resources must be considered to achieve effective load balancing. First, assign tasks according to the capacity of the node to each node reasonably. Hardware performance is defined by two key factors — computing power and the bandwidth of resource nodes for system communication. As hardware efficiency, the following equation can be expressed.

$$P_i = CP_i + BW_i \qquad (1)$$

In Equation (1), CP_I represents the computing power and BW_I represents the communication bandwidth of resource node i.

Load indicators of resource node consist mainly of CPU utilization, memory usage and system occupancy of bandwidth. Equation (2) can be used to express the load balancing value of the resource nodes.

$$LB_i = C_i + M_i + B_i \qquad (2)$$

In Equation (2), LB_i represents the load balancing value C_i represents the CPU utilization rate M_i represents the memory usage and B_i represents the bandwidth occupancy of resource node i.

The average load value of all resource nodes in SWIM is expressed in Equation (3).

$$LA_i = \sum_{i=1}^{n} LB_i/n \qquad (3)$$

Therefore, Equation (4) describes the resource node load standard deviation.

$$LS_i = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(LB_i - LA_i)^2} \qquad (4)$$

The device load balance level can be expressed by measuring the LS regular load node default. The higher the LS value the more the device load is unequalled. The smaller the LS value, the more the system load is balanced.

The initial secretion matrix is updated on the chosen best path to direct future ants a lot of powerfully toward higher answers. There are two kinds of pheromone updation. The local pheromone updating rule is given in equation (5).

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\tau_0 \qquad (5)$$

The local pheromone updating rule is applied to decrease the value of i.e., to reduce the attraction for the later ants. Thus local pheromone updating rule helps to enhance the diversity of the algorithm. After all ants have built their solutions, global pheromone updating rule is applied on the best path using equation (6).

$$\tau_{ij} = (1-\rho)\tau_{ij} + \rho\Delta\tau_{ij}^{best} \qquad (6)$$

where τ_{ij} means if best ant k travels on edge i, j.

The hardware output parameters of the network node and the device average load gap to change the pheromone τ_{ij}, as described in Equation (7).

$$\Delta\tau_{ij}^{best} = \sum_{k=1}^{n} \tau_{ij}^{k} \qquad (7)$$

We also need to consider the completion time factor of all tasks in order to maximize the utility and try to minimize the completion time of the task. So, it estimates the expected completion time.

4. SIMULATION ENVIRONMENT

Simulation is done by using CloudSim toolkit package to perform the simulation of Modified Load Balancing – Ant Colony Optimization (MLB-ACO) algorithm. The experiment was conducted to correlate existing scheduling algorithms such as FCFS, Classical ACO, and LBACO [13-17] with the proposed MLB-ACO algorithm. In CloudSim architecture contains n users such as user1, user2 ….user n and all users have an independent task. All users who have a separate task are shown as T1, T2 ...Tp. The tasks distributed to a number of VMs are scheduled by a user. There are q virtual machines with n data centres. Any number of hosts may be included in each data centre. We find that each data center has j hosts. CIS provides data center broker, virtual machine scheduler, and VM assignment policy. The data center broker is responsible for monitoring activities that interact with each other. VM Scheduler delivers policies for each data centre broker, virtual machines, and VM allocation policy. The VM policy determines the scheduling capability of the tasks on each virtual machine. An efficient algorithm is used to optimize task scheduling and improve the quality of services.

5. EXPERIMENTAL RESULT

Data Centre contains 50 virtual nodes. Head node with ID 10 is set, because it has a maximum number of neighbors in the server network simulation. Suppose there are 10,000 requests spread over the nodes and the calculated threshold is 370. The threshold value depends on the number of requests and the availability of resources.

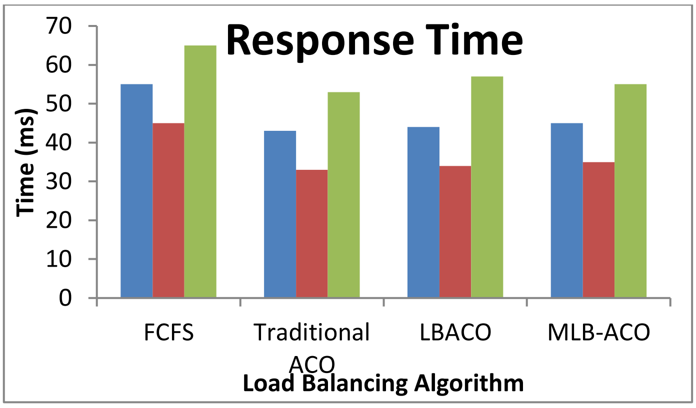

Figure. 3 Response Time Analysis with different Load Balancing Algorithm

The figure.3 shows the response time analysis with the proposed MLB-ACO for various types of load-balancing algorithms such as FCFS, Modern ACO and LBACO. Nodes operate similarly and data transmitted without impacting system performance has been checked with MLB-ACO to all existing algorithms. It provides the best performance shown in figure. Figure.4 demonstrates a comparison of the average time interval for various numbers of simulated tasks utilizing current FCFS, Modern ACO and LBACO with our proposed MLB-ACO algorithm. The diagram shows that the average time in MLB-ACO is always less than that in FCFS, LBACO and Traditional ACO. MLB-ACO is always better than the FCFS, Traditional ACO, and LBACO. The average minimum duration is always better.

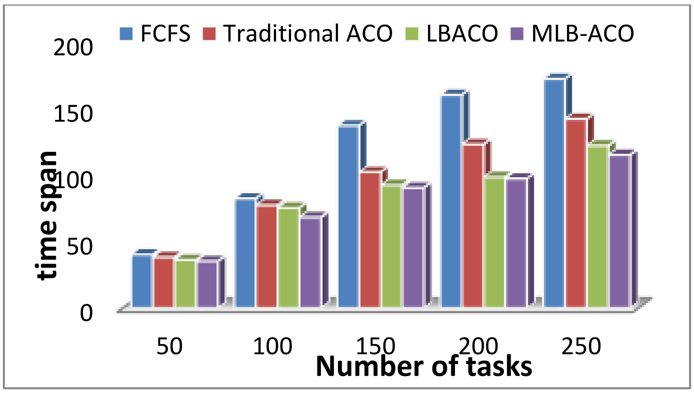

Figure. 4 Comparison of time span on different number of tasks

6. CONCLUSION

A new Modified Load Balancing Ant Colony Optimization (MLB-ACO) algorithm has been presented in this paper for task scheduling in cloud environment. We had designed a new algorithm and compared the results with existing algorithms like FCFS, Traditional ACO and LBACO. The proposed algorithm is simulated using CloudSim by different parameters and the results are compared. We evaluated that the approach presented is doing higher than the previous algorithms. Observational test results suggest that utilizing an improved MLB-ACO method, the time period has been shortened significantly by growing the amount of tasks relative to FCFS, Standard ACO and LBACO.

REFERENCES

1. Sajjan R.S, Biradar Rekha Yashwantrao, “Load Balancing and its Algorithms in Cloud Computing: A Survey” International Journal of Computer Sciences and Engineering Volume-5, Issue-1, January 2017.
2. Rajwinder Kaur and Pawan Luthra,”Load Balancing in Cloud Computing”, Association of Computer Electronics and Electrical Engineers, 2014, DOI: 02.ITC.2014.5.92.
3. Ms.Sajjan R.S., Ms.Noorjahan Mohammed Rafique, “Cloud Resource Virtualization: A Survey”, International Journal for Research in Applied Science & Engineering Technology (IJRASET) Volume 4 Issue II, February 2016.
4. Venubabu Kunamneni, “Dynamic Load Balancing for the Cloud”, International Journal of Computer Science and Electrical Engineering (IJCSEE) Vol-1 Iss-1, 2012.
5. Garima Rastogi, Dr Rama Sushil, “Analytical Literature Survey on Existing Load Balancing Schemes in Cloud Computing”, International Conference on Green Computing and Internet of Things (ICGCloT) 2015.
6. Klaithem AI Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela AI-Jaroodi, "A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms", IEEE Second Symposium on Network Cloud Computing and Applications, 2012.
7. Shanti swaroop moharana, rajadeepan d. Ramesh, digamber powar, "Analysis Of Load Balancers In Cloud Computing", International Journal of Computer Science and Engineering (UCSE) ISSN 2278-9960 Vol. 2, Issue 2, May 2013.
8. Divya Chaudhary, Rajender Singh Chhillar, "A New Load Balancing Technique for Virtual Machine Cloud Computing Environment", International Journal of Computer Applications (0975 - 8887) Volume 69-No.23, May 2013.
9. R. Kanakala; V. K. Reddy; K. Karthik, " Performance analysis of load balancing techniques in cloud computing environment”, 2015 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT), Pages: 1 - 6, DOI: 10.1109/ICECCT.2015.7226052
10. M. Dorigo, V. Maniezzo and A. Colorni, “Ant System: Optimization by a Colony of Cooperating Agents”, IEEE Transactions on Systems, Man, and Cybernetics, PP. 29-41, 1996.
11. C.W. Chiang, Y.C. Lee, C.N. Lee and T.Y. Chou, “Ant Colony Optimization for Task Matching and Scheduling”, IEE Proceedings on Computers and Digital Techniques, 153 (6), pp. 373-380, 2006.
12. M. Dorigo, M. Birattari and T. Stutzle, “Ant Colony Optimization-Artificial Ants as a Computational Intelligence Technique”, IEEE Computational Intelligence Magazine, pp. 1-12. 2006.
13. Gang Li and Zhijun Wu, “Ant Colony Optimization Task Scheduling Algorithm for SWIM Based on Load Balancing”, Future Internet 2019, 11, 90.
14. Kumar Nishant et all, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization” 14th International Conference on Modelling and Simulation, 2012.
15. Prachi Verma, Sonika Shrivastava, R.K.Pateriya, “Enhancing Load Balancing in Cloud Computing by Ant Colony Optimization Method”, International Journal of Computer Engineering In Research Trends, Volume 4, Issue 6, June-2017, pp. 277-284.
16. Arvinda Kushwaha, Mohd Amjad, Arvind Kumar, “Dynamic Load Balancing Ant Colony Optimization (DLBACO) Algorithm for Task Scheduling in Cloud Environment”, International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8 Issue-12, October 2019.
17. Shagufta khan 2Niresh Sharma, “Ant Colony Optimization for Effective Load Balancing In Cloud Computing”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 2, Issue 6, November – December 2013.