IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 30-36
www.iosrjournals.org

A New Approach for Dynamic Load Balancing in Cloud Computing

Anjali[1], Jitender Grover[2], Manpreet Singh[3], Charanjeet Singh[4], Hemant Sethi[5]
[1,2,3,4,5]*(Department of Computer Science & Engineering, MM University, Sadopur, Ambala, India)*

ABSTRACT: *Load Balancing is crucial for various operations in distributed situations. Now a days, Cloud Computing is getting a lot of consideration. Users are requesting for more administrations as well as effective results. To achieve this, load balancing is necessary hence, it turned into an extremely interesting for research. In distributed environment, adequate amount of resources are required which are used in such a way that resources are not over-utilized or under-utilize in any circumstance. Many researchers suggest different approaches for load balancing. This paper shows the new approach for Dynamic Load Balancing using the concept of Agent. In this new approach, an entity known as mobile agent performs the basic task. Mobile agent is a software program which executes independently. This paper also compares the proposed protocol with the traditional scheme used for load balancing and the results concludes that the proposed approach greatly reduces the communication cost of servers, accelerates the rate of load balancing which indirectly improves the Throughput and Response Time of the cloud.*

Keywords - *cloud computing, mobile agent, load balancing.*

I. INTRODUCTION

Cloud computing is receiving a great deal of attention. Cloud computing provides an effective services where a user can obtain storage space and various different resources. Earlier, it was very difficult for some organizations to increase number of instances of their computing resources like hardware, software or storage media for increasing their computation power or storage. [1] [2] But, the advancement of technology brings concept of cloud computing into the existence which helps to overcome such kind of difficulties. It makes possible for users to access their data at anytime from anywhere. Small organizations can store user's information in the cloud; this eliminates the cost of purchasing and storing memory devices [3].

The evolution of cloud computing also increases the number of users and their demands. User needs to increase the resources in cloud system to improve the performance of his task. The increased number of demands and services directly affects the workload of servers. To maintain the performance of cloud computing it is necessary to decrease the workload. One solution to overcome such issues is to balance the load by applying load balancing algorithms. [4] Load balancing divides the work of one server into the available servers. In this way more work gets done in the same time.

The two basic tasks of load balancing algorithm are resource allocation and task scheduling. This will leads to the easy availability of resources and reducing the cost of using resources. The cloud computing takes place when numerous computers are using services which are distributed over the network (internet) and connected to the data center (private/public). Every service is loosely attached. If one service gets failed then it will not affect the other services. Cloud computing is performed in two phase as frontend and back end. [5] The front end is a client who gets served by those services which are provided by the back end which is the cloud system. Various researchers work on the concept of load balancing and provides different algorithms which helps to reduce the workload of the server.

There are two types of working environment i.e. static and dynamic. In static environment, resources installed are homogeneous i.e. similar type of resources are used. In this, the user requirements are known prior to run time. In dynamic environment, resources installed are heterogeneous i.e. different type of resources are used. In this, no prior knowledge is required.

The proposed approach for Dynamic Load Balancing in Cloud Computing uses agents. The use of mobile agent in the proposed algorithm shows better results than existing load balancing algorithms. The whole working is done with the help of agents known as a regular agent and a mobile agent. Mobile agent is a program which migrates from one machine to another. [6] With the help of this an executable code is moved to a new host. It runs independently according to the interest of client. Mobile agent adds to regular agent. There are different features of mobile agent which makes it unique are, the capability of learning and mobility [15] [16]. In addition to this, [17] mobile agents also have the benefits of bandwidth conservation, reduction of completion

time, load balancing, dynamic deployment etc. [7] [8] Mobile agent is used for monitoring, information retrieval, remote control and dynamic systems.

II. RELATED WORK

For the balancing the load, different researchers provides different algorithms. Depending upon the working nature, load balancing algorithms are categorized into two broad categories such as static algorithms and dynamic algorithms [9]. In static algorithms, all the information is available to the algorithm which runs before any real computation starts. Hence, it arises some problems with static load balancing. It can be used in some networks where information needed is less.

In practical approach, dynamic load balancing algorithms are more powerful. It requires asynchronous communications and relies on knowledge as well as run time properties. Some of the different schemes proposed by the different authors are mentioned below.

Data duplication is the main problem while retrieving information. To minimize this problem authors in [10] proposed a new scheme which is known as INS (Index Name Server). This algorithm is used to find the optimization selection point depending upon the number of parameters. This algorithm is useful in many ways but apart from its success it also has some disadvantages. The main issue is its complicated implementation and lacks in future behavior identification.

In [11], authors proposed a new algorithm for the purpose of load balancing known as CLBDM (Central Load Balancing Decision Model). Although there are various algorithms used in scheduling and out of all, the most famous algorithm is Round Robin algorithm [12]. CLBDM is an improved version of Round Robin. The basic difference between the two is connection formation and termination. In improved version, connection time is calculated and if it exceeds the connection terminates. It reduces the requirement of human administration. The problem facing by this algorithm is a single point failure.

Ant colony algorithm for load balancing has been used by some authors for their research. Authors in [13], tried to improve this algorithm for cloud environment or distributed environment. According to authors this can also be used for complex networks. Best-case scenarios are: it balances a load in such a way that the problem of single point failure does not occur.

The collection of information is also becomes fast. Due to the existence of enormous ants in the network, this algorithm is also suffered from some problems that are: network overhead problem and the change in the nodes status. For cloud computing, authors in [14] presented a new scheme for Load Balancing which relies on the Artificial Bee Algorithm, which is a bionic method based on the gathering behaviour of honeybee. Through imitation of behaviour of honey bees, it optimizes the amount of nectar (i.e., system throughput) to reach the maximum throughput.

III. SYSTEM MODEL

Figure 1 shows the proposed system architecture. In figure, there are number of clients ranging from 1 to n. All these clients are attached with cloud service providers. The service provider consists of three units i.e. virtual machines, management unit, and few servers ranging from 1to n. Servers are considered as shared pool of resources. At the shared pool of servers, agent complete one cycle in two walks:

- In First walk it moves from initiation server to last server and gathers information from all servers, for making appropriate decision for load balancing and
- In second walk it balances the server's load on the basis of average load of the cloud.

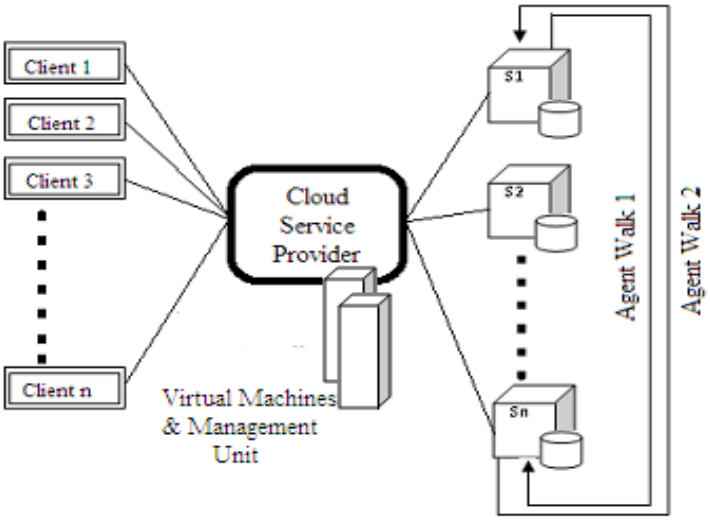

Fig. 1: Architecture of proposed Agent Based Dynamic Load Balancing

Agent Walk1:
Before the walk1 description some parameters are described which will be used by the agent during walk1. These are as follow:

- Suppose there is a shared pool of servers S, which consists N number of servers, S= {S_0, S_1,....., S_n } and Each server S_i (i=0, 1, …., n) has variable number of jobs.
- Next parameter for average number of jobs is: AVERAGE, which represents, required number of jobs for balanced status of the server, which will be calculated as follows:

$$\text{AVERAGE} = \frac{\sum J_i}{N};\ (0 < i < n)$$

Whereas,
J_i = Number of jobs at i^{th} server
N = Total number of servers

Now let's move towards the description of first walk. Figure 2, shows the pictorial representation of Agent Walk1. In Agent Walk1, agent is activated at any random server and finds number of jobs in queue at that server. Agent will repeat this process for all servers of that shared pool. After that it will calculate AVERAGE. On the basis of AVERAGE, it will sense the server's status in terms of overloaded and under loaded. Now it will make a list that contains server's status, which will be decided as follow:

For each server S_i
{
If (J_i > AVERAGE)
Then transfer jobs STATUS as "Overloaded"
Else
Set server's STATUS as "Underloaded"
}

Agent is still at the last server with the record of all the servers' load and it is ready for backtracking i.e. traveling from last server to first server. In this way agent will complete Agent Walk1 or half cycle.

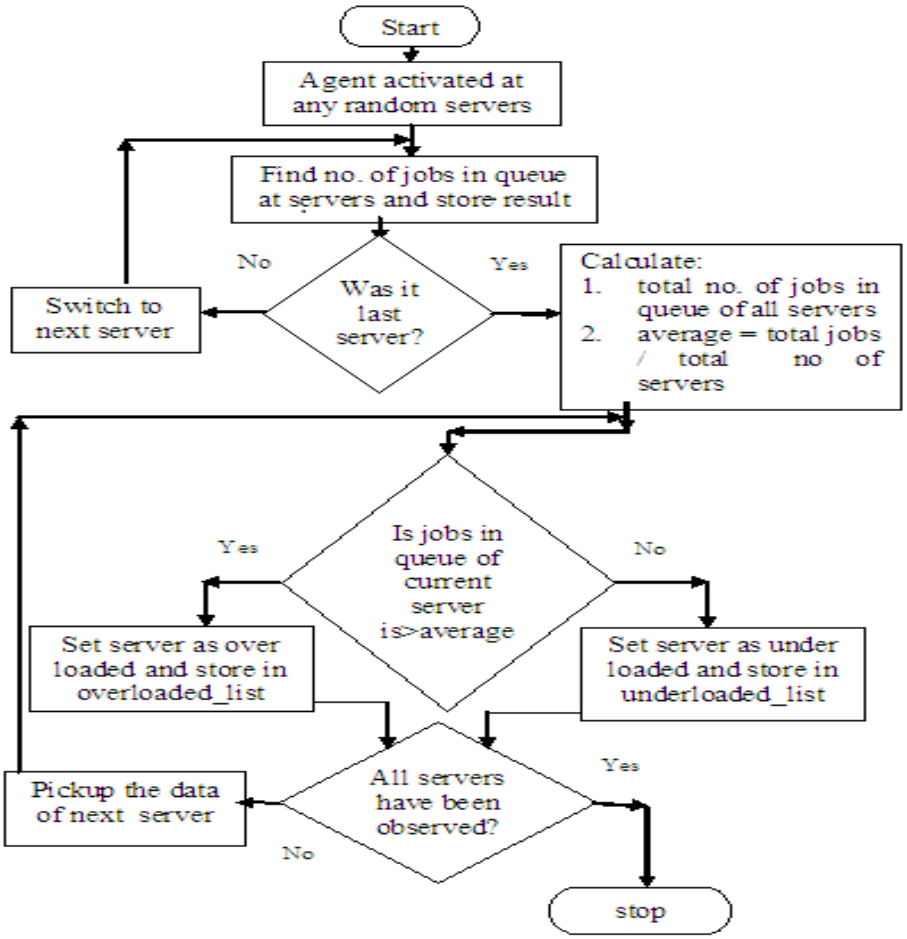

Fig. 2: First walk of agent from first to last server for gathering load information.

Agent Walk 2:
Figure 3, shows the pictorial representation of Agent Walk 2. Agent will start backtracking from last server to first server for balancing load of servers. At each server it will check the condition,

For each server S_i
{
If (STATUS == "Overloaded")
Then transfer jobs to "Underloaded" Servers
Else
Receive jobs from "Overloaded" Servers
}

If server is overloaded then finds the number of jobs to be transmitted over the under loaded server and transfer it and if server is under loaded then finds number of jobs that server can receive and migrates the jobs from heavily loaded servers. Whereas number of jobs to be transmitted from overloaded server and number of jobs that an under loaded server can receive will calculated as follow,
Number of jobs to be Transmitted or Received =

$$|J_i - \text{AVERAGE}|$$

Agent will perform this operation until it reaches at the first server with balancing all servers' load including first server also. In this way agent will balance the load without interrupting the system's work.

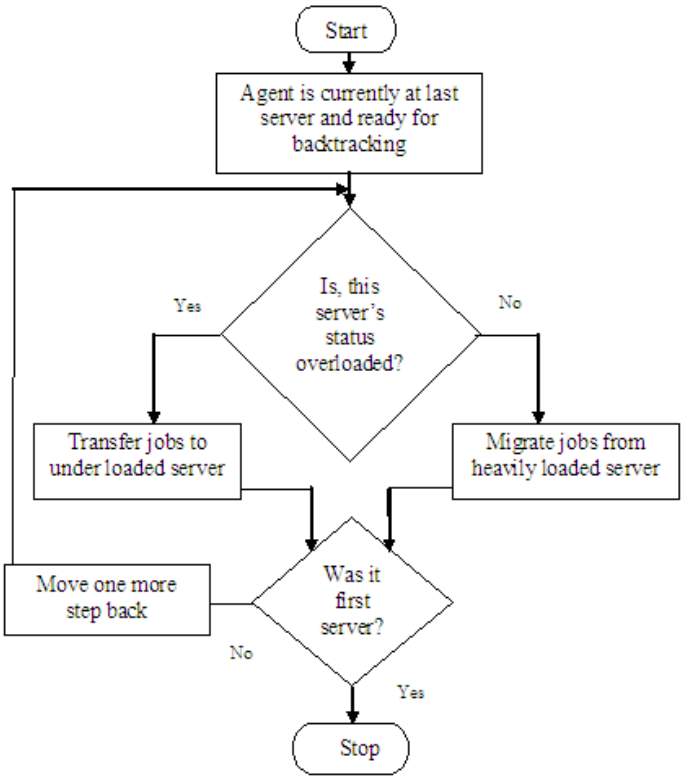

Fig. 3: Second walk of agent from last server to first server for balancing load

IV. RESULTS AND DISCUSSION

The proposed Dynamic Load Balancing Algorithm and one existing Centralised Server based load balancing schemes have been implemented using MATLAB. Parameters used for implementation are given in Table 1. This implementation is used to show the advantages of proposed load balancing scheme over existing one.

TABLE I. Simulation Parameters

Parameters	**Values**
No of Servers	10, 20, 30, 40, 50
CPU Time Consumption (Centralized Server based load balancing scheme)	10 Units
CPU Time Consumption (Agent Based Dynamic Load Balancing)	1 Unit
CPU Time Consumption (Process of balancing load)	50 Units
Platform	Matrix Laboratory 2014
Operating System	Windows 7

(a) In Centralized Server based load balancing scheme, CPU Time consumption is 10 Units as Server to Server communication takes lots of CPU Time in passing load information to Central Server and vice-versa.

(b) In Agent Based Dynamic Load Balancing scheme CPU Time consumption is 1 Unit as Agent-Server communication costs very less in passing load information to Agent and vice-versa.

(c) Time consumption by both the schemes in the process of balancing load after the collection of load information from all the servers is taken as equal that is 50 Units of time.

Implementation is done on the basis of scalability for 10, 20, 30, 40, 50 servers having random number of jobs for each server in a cloud. In Figure 4, the CPU Costs for both proposed and existing load balancing schemes are depicted on the basis of scalability. The graph shows that the CPU time unit consumed in Load balancing in proposed algorithm is much lesser than the Centralized Server based load balancing.

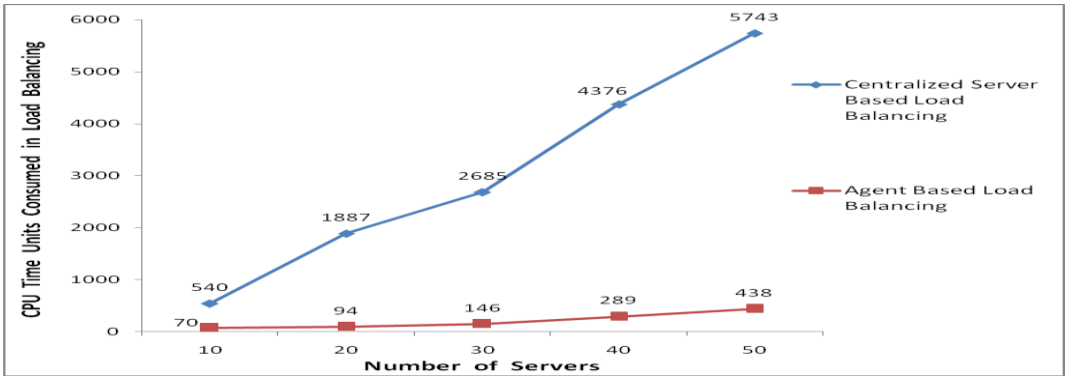

Fig. 4: Performance comparison in terms of CPU Time Unit Consumed in Load Balancing

Figure 5: Performance comparison in terms of Throughput

Throughput comparison of proposed algorithm and Centralized Server based load balancing is shown in Figure 5. The graph shows better Throughput results in proposed algorithm as compared to the existing one.

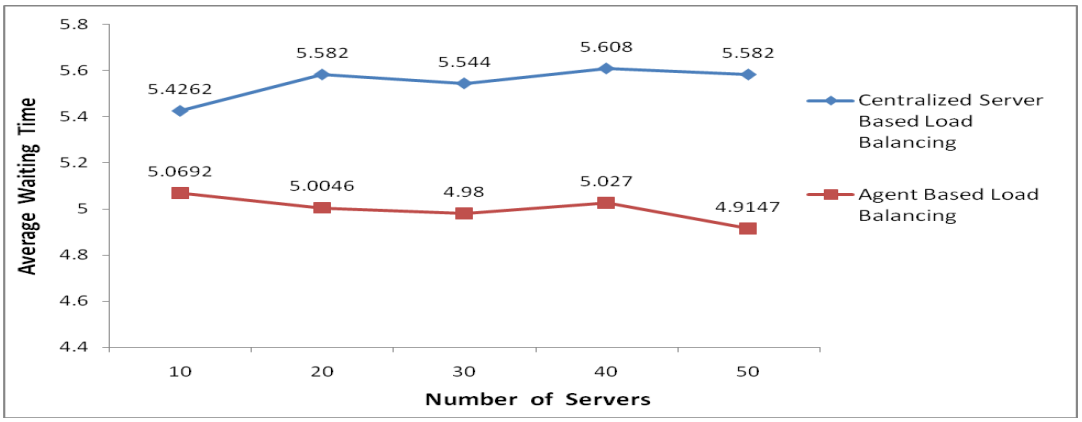

Fig. 6: Performance comparison in terms of Average Waiting Time

Comparison of Average Waiting time of proposed algorithm and Centralized Server based load balancing is shown in Figure 6. The graph shows that the proposed algorithm performs better. The Throughput and Average waiting time illustrated in Figure 5 and Figure 6 respectively also confirms the above said fact that the CPU can utilize saved time (illustrated in Figure 4) in entertaining other jobs which is in queue and that will directly improves the throughput and reduces average waiting time of the jobs.

In proposed system CPU Cost decreases because in Agent Based Dynamic Load Balancing scheme, Agent moves from server to server, consumes negligible CPU Time and saves server's time of communication that it has to use while communicating with Central Server deployed for load balancing in existing system. This

CPU Time can be used to entertain other jobs in queue by the server and it automatically will increase Throughput & decrease Response Time of the cloud.

V. CONCLUSION

This paper presents a scheme for Dynamic Load Balancing for cloud computing. After comparing it with traditional load balancing scheme, this paper concludes that proposed scheme is better and makes system independent form constant monitoring of the servers for load balancing, which is the requirement of existing dynamic load balancing algorithms in cloud computing and paper also observed that this method greatly reduces the communication cost of servers, accelerates the rate of load balancing and improves the Throughput and Response Time of the cloud. Existing load balancing schemes depends upon the bandwidth or traffic condition to transfer the load information from one server to another but in purposed load balancing scheme lightweight agent can move easily from one server to another without affecting the network's load too much.

REFERENCES

[1] P. Mell and T. Grance, "The nist definition of cloud computing (draft)", Tech. rep., National Institute of Standards and Technology, 2011.
[2] V.Venkatesa Kumar & K. Dinesh, "Job Scheduling Using Fuzzy Neural Network Algorithm in Cloud Environment", Bonfring International Journal of Man Machine Interface, Vol. 2, No. 1, March 2012.
[3] Alexa Huth and James Cebula."The Basics of Cloud Computing", Carnegie Mellon University. Produced for United State Computer Emergengy Readingness Team, a government organization, 2011.
[4] Young Joon Lee, Geon Yong Park, Ho Kuen Song, and Hee Yong Youn, "A Load Balancing Scheme for Distributed Simulation Based on Multi-Agent System",IEEE Annual Computer Software and Applications Conference Workshops (COMPSACW), pp. 613-618, July 2012.
[5] Wentao Liu, "Research on Cloud Computing Security Problem and Strategy", IEEE International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, pp. 1216-1219, April 2012.
[6] Buchanan, W.J., Naylor, M., Scott, A.V., "Enhancing network management using mobile agents," Seventh IEEE International Conference and Workshop on the Engineering of Computer Based Systems, Edinburgh, pp. 218-226, April 2000.
[7] Sandeep Rana1 & Niraj Singhal, "Design of Human Powered Directories using Mobile Agents", International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 1 Issue11, pp 016-021, February 2013.
[8] Divya Rastogi, Abhay Bansal & Nitasha Hasteer, " Techniques of load balancing in Cloud Computing: A Survey", International Conference on Computer Science and Engineering (CSE), Bangalore, April 2013.
[9] Lee, R. and B. Jeng, "Load-balancing tactics in cloud," IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), pp.447-454, October 2011.
[10] T-Y., W-T. Lee, Y-S. Lin, Y-S. Lin, H-L. Chan and J-S. Huang, "Dynamic load balancing mechanism based on cloud storage", IEEE International Conference on Computing, Communications and Applications (ComComAp), pp:102-106, January 2012.
[11] Radojevic, B. and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments" 34th IEEE International Convention on MIPRO, pp. 416-420, May 2011.
[12] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", 24th IEEE International Conference on Advanced Information Networking and Applications Workshops, pp. 551–556, 2010.
[13] Zehua Zhang & Xuejie Zhang, "A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation", IEEE International Conference on Industrial Mechatronics and Automation (ICIMA), Volume 2, pp.240-243, May 2010.
[14] Jing Yao & Ju-hou He, "Load Balancing Strategy of Cloud Computing based on Artificial Bee Algorithm", IEEE International Conference on Computing Technology and Information Management (ICCM), Volume 1, pp. 185-189, April 2012.
[15] Jitender Grover, Shikha and Mohit Sharma, "Cloud Computing and Its Security Issues - A Review", presented in IEEE Fifth International Conference on Computing, Communications and Networking Technologies (ICCCNT 2014), Hefei, Anhui, China, DOI: 10.1109/ICCCNT.2014.6962991, pp. 1-5, July 11-13, 2014.
[16] Shivangi Katiyar, Jitender Grover and Nipun Mehta, "Agent Based Performance and Reliability Enhancement Scheme for Cloud Computing using Markov Model", presented in The International Conference on Futuristic Trends in Computational Analysis and Knowledge Management (INBUSH ERA-2015), Amity University, Greater Noida, Uttar Pradesh, India, 25-27 Feb, 2015.
[17] Jitender Grover and Shivangi Katiyar, "Agent Based Dynamic Load Balancing in Cloud Computing", IEEE International Conference on Human Computer Interactions (ICHCI'13), DOI: 10.1109/ICHCI-IEEE.2013.6887799, pp. 1-6, 23-24 August 2013.