A Comparative Nature Inspired Load Balancing Algorithms in a Cloud Computing Environment

Mahendar A
Ph.D. Research Scholar (External)
Faculty of Computer Science and Engineering
J N T U Hyderabad,
Kukatpally, Hyderabad, India
mahi.adapa@gmail.com

Dr. Shahu Chatrapati K
Head, Dept. of Computer Science and Engineering
J N T U H College of Engineering Manthani (JNTUHCEM) Karimnagar Dist, Telangana, India

Abstract— One of the primary issues in cloud computing is implementation of a novel load balancing approach. The demanding thirst for optimal performance of the system is creating research interest in this area. Many Load Balancing algorithms that aim to enhance the overall system performance have been proposed. In this paper, we survey a special group of Load balancing algorithms that have taken inspiration from nature. We provide an overview of the current trends in the field by discussing and comparing these algorithms.

Keywords— Cloud computing, Load balancing, swarm Intelligence

I. INTRODUCTION

The exponential growth of cloud computing in the recent years has attracted research and academia to this field. Load Balancing is a primary issue that needs to be taken care of. Several Load Balancing algorithms have been proposed and investigated; however, there are issues yet to be addressed. Load balancing is “the process of distributing the work load among various nodes of a cloud based system to improve both resource utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while other nodes are idle or doing very little work.”

Load balancing algorithms are divided as static and dynamic based upon the working environment [1], centralized and distributed based upon the control strategy [2]. Static algorithms are effective in stable and homogenous environments where as Dynamic algorithms are effective in dynamic and heterogeneous environments. The centralized strategy requires an arbiter or control node to perform the load balancing act whereas in distributed strategy load balancing is performed by all the nodes of the system. Many

Load Balancing algorithms have been proposed in the recent past. In this paper, we present a survey of Nature Inspired Load Balancing Algorithms that have been specifically developed for hosted environments. These algorithms use the concept of Swarm intelligence for Load Balancing [3]. We consider some of the potentially viable nature inspired algorithms for load balancing in large scale cloud environments. There are two popular classes of Nature Inspired Algorithms available in the literature namely Ant Colony and Honey Bee Colony. We give an overview of these algorithms, discuss their pros and cons and analyze their properties.

The rest of this paper is organized as follows. We discuss the related work in Section II. Then, in Section III we discuss the various challenges and issues of Nature Inspired load balancing in cloud computing environment. Afterwards in Section IV, we review and compare Nature Inspired Load Balancing Algorithms that are available currently in the literature. Section V concludes the paper and highlights future enhancements that can be done in Cloud Load Balancing.

II. RELATED WORK

Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-jaroodi have presented a survey of Load Balancing in Cloud Computing. The paper gives an overview of Load Balancing Algorithms like INS, ESWLC, CLBDM, Ants Colony, Mapreduce, VM Mapping and DDFTP. It compares the algorithms based upon on certain parameters[1]. Martin Randles, David Lamb,A. and Taleb-Bendiab have presented a comparative study of three distributed load balancing algorithms namely Honeybee based load balancing, Biased Random Sampling and Active Clustering. The paper describes and compares the algorithms by performing experiments using simulations set up in Repast.NET[4]. Rich Lee and Bingchian Jeng in their work made a comparitative analysis of Round-Robin, Weighted Round-Robin, Least Connection, Shortest Expected Delay, Resource Best and Resource Fit algorithms using a simulation program based on GNU R[5]. V. Sesum-Cavic and E. Kuhn have presented the advantages of using swarm intelligence in load balancing in their works [6].

III. ANALYSIS OF ISSUES RELATED TO NATURE INSPIRED CLOUD LOAD BALANCING ALGORITHMS

In this section we give an introduction to the major challenges a Nature Inspired Cloud Load Balancing Algorithm must address before it is implemented in the system. These challenges if not addressed properly may affect the performance of the algorithm. These challenges are summarized as follows.
A. Nature of the Cloud

Static Nature Inspired Cloud Load Balancing Algorithms are designed to work with static clouds. The performance of these algorithms is satisfactory in stable cloud environments. However, it is a challenge to design a dynamic Nature Inspired Cloud Load Balancing Algorithm that is flexible and adapts to the dynamic changes in the attributes of the system.

B. Control Mode

Nature Inspired Cloud Load Balancing Algorithms are mostly centralized. Having a single point of control in large scale cloud computing environments makes load balancing a daunting task. Moreover, if the arbiter goes down it brings the whole system to a halt which is not desirable. Hence, distributed or even hybrid algorithms are required. Designing a distributed Nature Inspired Cloud Load Balancing Algorithm that gives optimal performance is a challenge.

C. Resource Awareness

Nature Inspired Cloud Load Balancing Algorithms are designed to work with homogenous resources. But in most cases, a cloud computing environment is a collection of heterogeneous resources. If the algorithm being implemented is not resource aware then its performance will be adversely affected. Hence, developing a Resource aware Nature Inspired Cloud Load Balancing Algorithm is a challenge.

D. Geographical separation of the cloud nodes

Algorithms that perform well with cloud nodes distributed over geography are yet to be devised. There are some additional parameters that must be considered by the algorithm during balancing. The parameters include network speed, distance between processing node and clients, distance between nodes. Hence algorithms that work well with geographically separated cloud nodes are required [7].

F. Replication Model

In order to guarantee the SLA’s Nature Inspired Cloud Load Balancing Algorithms must support data replication. This can be either complete or partial replication. A fully replicated algorithm comes with additional costs as data on replication nodes must be maintained. It doesn’t use the available storage efficiently. On the other hand even though partial replication algorithms utilize the storage resources efficiently, they are very complex to design [8].

G. Ease of implementation

It is desirable that Nature Inspired Cloud Load Balancing Algorithms are easily implemented and operated. Complexity in algorithm’s implementation will raise performance issues. Therefore, Simple algorithms are required [9].

H. Network Overhead

Nature Inspired Cloud Load Balancing Algorithms have to deal with overhead of the network as the so called software ants honey bees continuously traverse through the network to gather information of the cloud nodes. The presence of large number of ants or bees may sometimes degrade the system performance [10].

I. Scalability

The scale of a cloud computing platform can be very large. Nature Inspired Cloud Load Balancing Algorithms must be devised to take in to consideration the scalability factor. They must be flexible enough to work in a situation where a resource can be randomly added. The algorithm must quickly take into consideration the newly added resource and perform load balancing with little effect on system performance.

J. Synchronization

Nature Inspired Cloud Load Balancing Algorithms perform the operation of load balancing with the help of agents like artificial ants or honey bees. The algorithm generates a large number of agents. These agents continuously traverse through the cloud and monitor it. Achieving synchronization among the agents is a challenging task for the algorithms.

IV. REVIEW OF NATURE INSPIRED CLOUD LOAD BALANCING ALGORITHMS

In this section we give a review of Nature inspired cloud load balancing algorithms that are currently available in the literature. Nature inspired cloud load balancing algorithms can be categorized in to two classes namely, Ant Colony Inspired Algorithms and Honey Bee Inspired.

Algorithms. We first discuss the Ant colony Inspired Algorithms that have been developed for cloud load balancing. Then later discuss the Honey Bee Inspired cloud load balancing algorithms. TABLE I shows a comparison of Nature Inspired Load Balancing Algorithms that are reviewed in this paper.

A. Ant Colony Inspired Load Balancing Algorithms.

All Ant Colony Inspired Load Balancing Algorithms are based on ACO algorithm [11, 12]. Individual ants are quite simple insects but colony of ants collectively perform a variety of tasks such as building anthills, foraging for food with great reliability and consistency [13,14]. This social behavior of ants has inspired researchers to solve many computational problems that includes even the problem of Load Balancing a Cloud. We discuss three popular ant colony inspired load balancing algorithms.

Zehua Zhang and Xuejie Zhang have proposed a Load Balancing Mechanism based on Ant Colony and Complex Network Theory (LBMACCN) in open cloud computing federation (OCCF) [15]. The mechanism aims to solve the issue of complex and dynamic load balancing in OCCF. Underload load balancing, Overload balancing, Pheromone updating, and Network Evolution are the major modules of LBMACCN. The operation of the algorithm suits for heterogenc cloud environments. The algorithm is designed to work with distributed clouds. The fault tolerance and the
scalability factor of the mechanism are excellent. The algorithm requires a full replication model. It has to deal with network overhead problem because ants keep on traversing across the network which sometimes may lead to delays. Synchronization of the ants and Static nature of the algorithm are issues to be considered.

Kumar Nishant et al. - have proposed an algorithm that performs the task of load balancing of nodes in cloud using the Ant colony optimization [10]. The algorithm is an improved version of algorithm presented in [15]. It aims at efficient load distribution among the cloud nodes such that the ants never come across a dead end during network traversal for building an optimum solution set. The algorithm suits for heterogeneous and distributed clouds. The fault tolerance and the scalability factor are excellent. The issue of synchronization between ants is solved. The algorithm requires a full replication model and has to deal with network overhead problem. It is a Static Load Balancing Algorithm.

Both the above mentioned algorithms work in the following manner, ants and pheromones are generated from a head node once a request is generated. In order to gather node information for task scheduling ant behavior is used. The ants start their forward route from the ‘head’. A forward movement means that the ant is searching for overloaded nodes whereas reverse movement indicates it is searching for an underloaded node. The authors of [10] have introduced a new feature termed ‘suicide’ to the ants. Once the target node is found the ant will be terminated preventing unnecessary backward movement.

Kun Li et al. - have proposed LBACO (Load Balancing Ant Colony Optimization) algorithm for load balancing in cloud based systems [16]. The algorithm is simulated using CloudSim version 2.1 toolkit package. The algorithm is based on ACO algorithm [11, 12]. The algorithm considers the past task scheduling time to carry out new scheduling strategy. It reserves the current optimal solution and uses it to make a decision in future Load balancing scenarios. The experiment discussed in [16] proves that LBACO is more effective when compared with ACO algorithm. The algorithm suits for dynamic, distributed and heterogeneous clouds. The algorithm assumes that the tasks are mutually independent, preemptive and computationally intensive.

B. Fast adaptive load balancing method

D. Zhang et al. [18] proposed a binary tree structure that is used to partition the simulation region into sub-domains. The characteristics of this fast adaptive balancing method are to be adjusted the workload between the processors from local areas to global areas. According to the difference of workload, the arrangements of the cells are obtained. But the main workload concentrates on certain cells so that the procedure of adjusting the vertices of the grid can be very long because of the local workload can be considered. This problem can be avoided by the fast load balancing adaptive method. Here the region should be partitioned by using the binary tree mode, so that it contains leaf nodes, child nodes, parent nodes etc. There were partition line between the binary tree and the indexes of the cells on the left are smaller that of right and the indexes on the top are smaller than the bottom. Calculate the workload based on the balancing algorithm. This algorithm has a faster balancing speed, less elapsed time and less communication time cost of the simulation procedure. Advantages are Relative smaller communication overhead relative smaller communication overhead, faster balancing speed, and high efficiency and the disadvantage is it cannot maintain the topology that is neighboring cells cannot be maintained.

C. Honey Bee Inspired Load Balancing Algorithms

Honey Bee inspired technique is used as a search technique in many computing applications where the system is highly scalable and dynamic [17]. A bee hive works as follows. There are two roles a bee can play namely forager bee and follower bee. Forager bees search for a suitable source of food, when found, they return back to the colony and advertise the same using a “waggle dance”. The quality, distance and quantity of the food are judged based upon the intensity of the dance. Follower bees are sent out to harvest the discovered food. The waggle dance is used to determine whether more bees are required to harvest or the exploited source to be abandoned. The same phenomenon is being applied in load balancing of nodes in cloud computing environment. We discuss 2 popular bee inspired Load Balancing Algorithms that are available in the literature.

Martin Randles, David Lamb and A. Taleb-Bendiab have proposed a Honey bee inspired Load Balancing algorithm [4]. The working of the algorithm can be explained as follows. The servers in the cloud take the role of either foragers or harvesters. An advert board that mimics the waggle dance of the bees is maintained. A server successfully executing a request will post on the board. A server that reads the board follows the chosen advert, and then serves the request; thus mimicking the honey bee. The server not reading the advert board serves a random virtual server’s queue request; thus mimicking the forager bee. The total colony profit is calculated based upon just-serviced virtual server profit. If the just serviced virtual server’s profit is high then the forager server will keep on posting an advert for it until the calculated profit becomes low. The algorithm is suitable for large scale, dynamic and heterogeneous cloud environments. The algorithm suffers with Network Overhead and replication issues. Given an optimum profit calculation method, this algorithm provides a good distributed solution to the problem of Load balancing in cloud computing environment.

Jing Yao and Ju-hou He, have presented a Load Balancing Strategy for cloud computing based on Artificial Bee algorithm [17]. The algorithm is an improved version of the algorithm proposed in [4]. The previous mechanisms were considering only lightly loaded nodes for balancing where the technique in [17] considers all resources in the cloud iteratively. The algorithm includes some extra operations that improve the quality of service. Experimental results show that in a system with a fixed number of systems and
increasing number of requests the algorithm generates an improved throughput whereas a change in number of servers and fixed number of requests the original ABC algorithm performs well. The algorithm is dynamic, distributed and suitable for heterogeneous cloud systems; whereas, it suffers in a highly scalable system. Network Overhead and replication model should also be considered before implementing this algorithm in a cloud based system.

D. Heat Diffusion Based Dynamic Load Balancing

Yunhu.et al. [19] proposed an efficient cell selection scheme and two heat diffusion based algorithm called global and local diffusion. Considered the distributed virtual environments there were various numbers of users and the load accessing by the concurrent users can cause problem. This can be avoided by this algorithm. According to the heat diffusion algorithm; the virtual environment is divided in two large numbers of square cells and each square cell having objects. The working of the heat diffusion algorithm is in such a way that every node in the cell sends load to its neighboring nodes in every iteration and the transfer was the difference between the current node to that of neighboring node. So it was related to heat diffusion process. That is the transfer of heat from high to low object, when they were placed adjacently in local diffusion algorithm, there were local decision making and efficient cell selection schemes are used. Here they simply compared the neighboring node loads to the adjacent node loads. If load is small then the transfer of load becomes possible. When global diffusion algorithm considered, it has two stages that is global scheduling stage and local load migration stage. From various experimental results the global diffusion algorithm becomes the better one. Advantages are communication overhead is less, high speed and require little amount of calculations. Disadvantages are network delay is high and several iterations are taken so there was a waste of time.

E. Decentralized Scale-Free Network Construction and Load Balancing in Massive Multiuser Virtual Environments

Markus et al. [20] addressed the concept of overlay networks for the interconnection of machines that makes the backbone of an online environment. Virtual online world that makes the opportunities to the world for better technological advancements and developments. So the proposed network that makes better feasibility and load balancing to the dynamic virtual environments. This proposed system developed hyper verse architecture that can be responsible for the proper hosting of the virtual world. There were self organized load balancing methods by which the world surface is subdivided in to small cells, and it is Managed by a public server. In this cells various hotspots so that the absolute mass of the object in the cell can be calculated by the public server. Hotspot accuracy is better when increasing the network load. The proposed algorithm cannot avoid the overloaded nodes but find out the number of links that assigned to each node while joining the network. The advantages are the network becomes reliable; the network becomes resilience, efficient routing, and fault tolerant. The disadvantage is the overload ratio at the beginning is higher so that public servers are initially placed randomly so some time is used for balancing the load.

F. Load Balancing in Dynamic Structured P2P Systems

Brighten et al. [21] proposed an algorithm for load balancing in dynamic peer-to-peer system and other hybrid Environments. In most peer-to-peer system the non uniform of objects in the space and algorithm, the load information of the peer nodes are stored in different directories. These directories help to schedule reassignment of the virtual servers to develop a better balance. Greedy heuristic algorithm used to find out a better solution for the proper utilization of the nodes. The huge number of virtual servers in the system helps to increase the utilization. The various load information in to the corresponding pool and then the virtual server assignments are to be done. This proposed algorithm should be applied to different types of resources like storage, bandwidth etc. It was designed to handle the various situations like varying load of the node, node capacity, entering and leaving of nodes and also insertion and deletion of the nodes. Advantages are high node utilization and increasing scalability. Disadvantage is the reassignment of the virtual server is difficult.

V. Issues Related to Swarm Intelligence Based Load Balancing Algorithms

In this section we give an introduction to the major challenges related to swarm intelligence based Cloud Load Balancing Algorithm that must be addressed before it is implemented in the cloud. These issues if not tackled properly may down the performance of the algorithm. These challenges are summarized as follows:

- If static swarm intelligence based load balancing algorithms are employed for static clouds. These algorithms perform well in stable cloud environment. However, Cloud environments are mostly dynamic and it is a challenging problem to design dynamic and flexible swarm intelligence based load balancing algorithm.
- Cloud environment comprises of heterogeneous resources but swarm intelligence based load balancing algorithms are of homogenous resources Resource aware load balancing algorithm can improve performance of the system so developing resource aware swarm intelligence based load balancing algorithm is a challenging task.
- Swarm intelligence based load balancing algorithms are generally centralized. Large scale cloud environment with single point of failure is a fearsome problem. Designing distributed swarm intelligence based load balancing algorithms is a challenging task.
- Swarm intelligence based load balancing algorithms generates large number of agents like particle swarm, honey bee or artificial ants. These agents help in
performing the load balancing task. These agents also monitor the cloud. Establishing synchronization among these agents is a difficult task. As swarm intelligence based load balancing algorithms produce large number of agents. These agents distributed in the network and monitor the cloud nodes. This large number of agents adds the overhead in the cloud network which can down the system performance.

- Cloud computing platform is very huge in general. Therefore, swarm intelligence based algorithm must consider the scalability factor and the algorithm must be able to cope up with dynamic resource allocation without affecting the system performance.

### TABLE II Comparison of Nature Inspired Load Balancing Algorithms

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Distributed</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Heterogeneous</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Geographical</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Separation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity</td>
<td>EASY</td>
<td>EASY</td>
<td>MEDIUM</td>
<td>EASY</td>
<td>MEDIUM</td>
</tr>
<tr>
<td>Network</td>
<td>YES</td>
<td>LESS</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Scalability</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Synchronization</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
</tbody>
</table>

### VI. CONCLUSIONS AND FUTURE WORK

We present a survey of Comparative Nature Inspired Load Balancing Algorithms that are available in the literature. We have discussed the challenges that these algorithms must address. The pros and cons of the algorithms have also been depicted. We have drawn out comparisons between the algorithms by taking into account the necessary parameters. All the algorithms discussed in the paper suffer with one or the other issue. Hence there is scope for improvement in the algorithms. Therefore, as part of future work we plan to propose our own Comparative Nature Inspired Load Balancing Algorithm for cloud computing that addresses the issues discussed earlier.

### REFERENCES


[16] Kun Li, Gaochao Xu, Guangyu Zhao, Yushuang Dong and Dan Wang “Cloud Task scheduling based on Load Balancing Ant Colony Optimization” Sixth Annual ChinaGrid Conference 2011


