A Framework to Optimize Load Balancing to Improve the Performance of Distributed Systems

Payal
Research Scholar /MMICT&BM
Maharishi Markandeshwar University, Mullana
Ambala, Haryana, India

Atul Garg
Associate Professor/MMICT&BM
Maharishi Markandeshwar University, Mullana
Ambala, Haryana, India

ABSTRACT

Load balancing in distributed systems is the technique to handle maximum requests in minimum time. Load balancing aims to optimize resource use, maximum throughput, minimize response time, increase reliability and scalability. Trying to achieve this is however not an easy task. Various techniques and approaches have been used to achieve this goal. The proposed work is inspired by the Ant Colony Optimization (ACO) technique. The algorithm is designed on a Semi-Distributed system where clusters are formed and each cluster consists of number nodes. Ant is created on these clusters and acts accordingly the needs of the environment. The objective of paper is to develop an efficient load balancing technique that can improve the performance of system.

Keywords

Load Balancing, Semi-Distributed systems, ACO.

1. INTRODUCTION

The development of next generation of computing systems, mobiles and other hand handled devices increased the number of users worldwide. Hence, maintaining the good performance and catering the dynamic requests of such exponentially growing users has become one of most challenging and complicated issues. Apart from sharing data and I/O devices, distributed systems also share the computational power among nodes to improve the system efficiency. To increase the efficiency of network multiple nodes are connected to each other. The work load distribution process is performed by Load Balancing. Load Balancing is extensively used to improve the system performance and resource utilization. Load Balancing is a process of reassigning the total load to the individual nodes of the system for avoiding situation where some nodes are heavily loaded while others are idle [1].

In general Load Balancing mechanism is divided in two categories: Static and Dynamic. Static Load Balancing is performed by considering predetermined behavior of the system. Whereas Dynamic Load Balancing considerers current state of the system. In Dynamic Load Balancing work load is distributed among nodes at run time. Dynamic algorithms are adaptive in nature i.e. the algorithm can be modified as the state of system changes. Applications where workload is unpredictable or change during execution dynamic algorithm are implemented to achieve better performance [4]. In distributed systems Dynamic Load Balancing can be achieved in three forms: Completely Distributed, Centralized and Semi- Distributed. In Completely Distributed form load balancing algorithm is executed by all nodes and responsibility of Load Balancing is shared among them. In Centralized approach algorithm is executed by one single node and all other nodes interact with this central node. Semi- Distributed on the contrary is combination of both Centralized and Completely Distributed approach [21].

Figure 1 shows the placement of Load Balancer in Distributed System. All incoming requests are forwarded to different nodes for execution. Internal process migration is possible in case a selected node is overloaded.

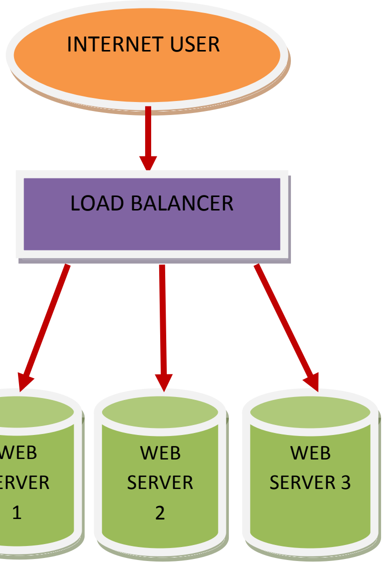

Figure 1: Load Balancer Implementation

Proposed work is done on Semi-Distributed system due to its ability of handling large number of requests and failure of any one node will also not affect the rest system. In Semi-Distributed approach clusters are formed, each cluster having a central node. Central node is divided into secondary nodes. These secondary nodes handle the load sent by overloaded primary nodes [12] [24].

Importance of Load Balancing has lead to the development of various techniques that use behavior of species for finding optimal solution. Swarm Intelligence is a technique of problem solving inspired by behavior of social insects such as wasps, ants or honey bees. These insects lives in colonies and each one specializes in a set of tasks, they divide their work on the basis of specialization. The ants for example have excellent ability of foraging for food [8]. When ants are on the way to search for food, they start from their nest and walk towards the food source. While walking ants deposit a chemical substance pheromone on path indicating its usage. Many ants may travel through different path to the same food source. The ants reinforce the path by depositing more pheromone on it that aids more ants to follow the path and hence identifies it as shortest path to reach food source [16]. Many researchers used Ant Colony in various fields like telecommunications [5], cloud computing [29] [31] [30], routing [19], Load Balancing on client- server [25].

2. PROPOSED WORK

Rapid growth in use of computer and Internet has increased the number of resource sharing applications and tremendously increased the amount of load across Internet. Problem is resolved to an extent by deploying multiple servers and distributing the load in effective manner. Distributed approach offered various advantages such as minimum storage, maximum resource utilization and continuous availability of system but still has disadvantage of extra communication overhead which makes it suitable for only those systems where nodes work individually with minimum communication [24]. Semi-Distributed system approach comes in middle between Centralized and Completely Distributed approach. It takes the best of each approach and avoids their major drawback of bottleneck and over communication. In Semi-Distributed policy nodes are divided in equal clusters. Each cluster adopts a centralized approach where central nodes take charge of load balancing within the cluster. Clusters together adopt a distributed approach and exchange information with each other to achieve global Load Balancing. Research shows that Semi-Distributed system produces better performance as compared to other two polices and is best in system having large number of nodes [18].

To make distributes systems more effective Ant Colony Optimization (ACO) is applied on it. Ali et al., [22] proposed ACO algorithm for Load Balancing in distributed systems by applying color ant colony to distinguish ants of different nests. In this mechanism load information is dynamically updated at each ant movement and proved better than its traditional approach. Similarly, Shagufta et al., [30] presented forward - backward ant movements to reach destination in minimum time and Dimple & Atul [25] proposed a framework where ant is created on both client and server side. Their work managed to improve the server performance. The researchers in [5][16][19][29] also presented various approaches of ACO in load balancing but despite of being providing better performance and efficiency no one worked on the Semi-Distributed platform. While applying ACO, the destruction of an ant before reaching its destination is not considered. Nothing is proposed in order to get acknowledgement from the server on successful execution of request. The main purpose of the work is to implement ACO to optimize a Semi-Distributed approach in Distributed Systems. The ant colony system proposed reacts according to the state of the node. The primary ant, semi ant and return ant are created in certain conditions.

In the proposed work Active Ant (AA) will be created on server side immediately after receiving request from client. Passive Ant (PA) is also created along with AA which will move along the AA and will keep track of its movement. AA will search for an under loaded node and will assign task to it. To find state of node an average value is calculated firstly. Number of jobs in the waiting queue of a node is compared with the average value.

$$A > W_i, \quad \text{Under loaded}$$

$$A < W_{i,} \quad \text{Over Loaded}$$

$$A = \text{Average Weight} \quad , \quad W_i = \text{Weight of node I}$$

Once AA assigns job to selected node, it will create a Return Ant (RA) which will take acknowledgement and notify client for successful execution of request. In case AA gets destroyed before reaching destination, PA will track back to the starting point and request for new AA to accomplish the task. On the other hand if PA gets destroyed then AA will create its new PA.

Figure 2 presents the high level view of proposed work.

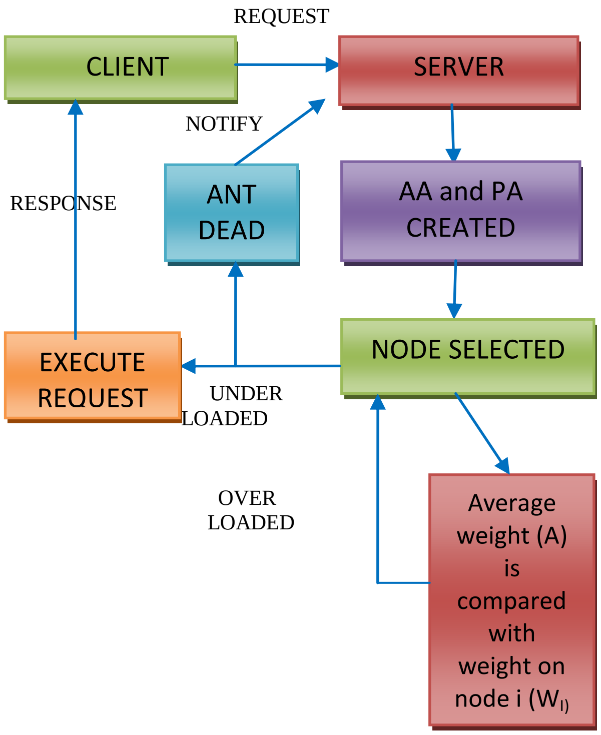

Figure 2: High Level View

Algorithm and flowchart for the same are given in figure 3 and figure 4.

```
Step 1: Client Send Request to Server
Step 2: AA and PA are created
Step 3: Select node randomly
Step 4: If node is under loaded go to step 5
          Else go to step 3
Step 5: If AA reaches destination create RA go to step 9
          Else go to step 6
Step 6: If AA gets destroyed than PA will go to starting
            point and request for new AA
    Else if PA gets destroyed than AA will create new PA
          itself
Step 7: End if
Step 8: Go to step 3
Step 9: Exit
```

Figure 3: Algorithm of Proposed Work

Working Algorithm

Client send request to server. Server creates AA and PA in step 2. In step 3 a node is selected randomly by AA. If the node is under loaded than request is forwarded to it and after successful execution at step 5 RA is created and acknowledgement is send to client. If AA gets destroyed before reaching destination node than at step 6 PA will go to starting point and request for new AA. After that new node is selected again and procedure is followed starting from step 3.

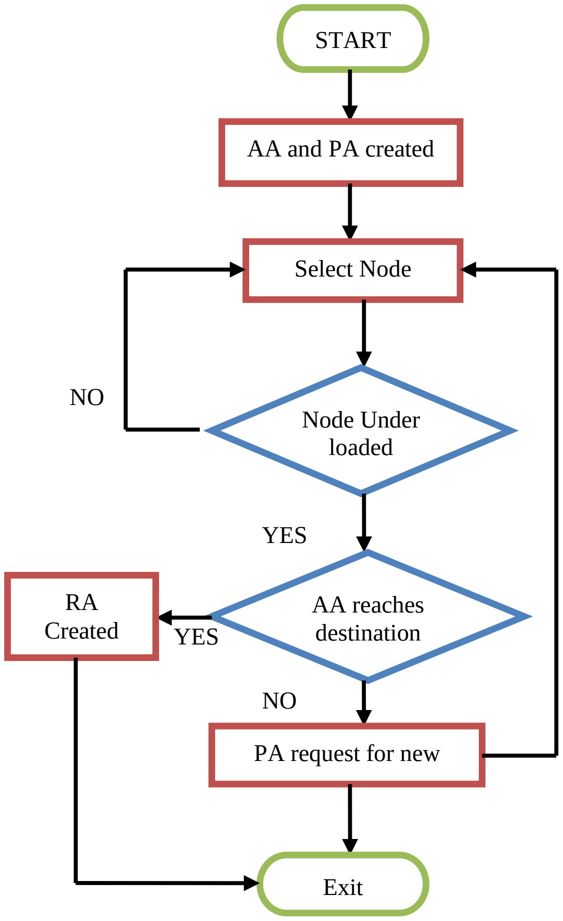

Figure 4: Flowchart of Proposed Work

3. CONCLUSION

Balancing the load on all nodes of systems is one of the most important tasks as it increases system efficiency, performance and scalability. To achieve this, a mechanism is proposed by implementing ACO on semi-distributed systems. Semi-Distributed approach is more reliable and provides better resources utilization, fault tolerance and response time as compared to other approaches. In the proposed work, server creates Active Ant (AA) and Passive Ant (PA) on receiving request from client. AA performs load balancing and PA keeps track of movements of AA, it also informs about the destruction of AA. Return Ant (RA) is created to send response to client. Sending acknowledgement to client and informing about ant destruction are the significant features which differentiate the proposed work from other similar techniques. Future work is to compare the algorithm with other techniques and to implement the mechanism in more realistic environment.

4. REFERENCES

[1] P.L.McEntire, J.G.O'Reily "Distributed Computing: concepts and implementation. New York: IEEE Press, 1984.

[2] Y.Wang and R.Morrois," Load balancing in distributed systems", IEEE Trans.Computing.C-34, no.3, Mar.1985.

[3] Derek L. Eager et al., "Adaptive Load Sharing in Homogenous Distributed Systems", IEEE Vol.12, 1986.

[4] Y.Lan and T.YU," A Dynamic Central Scheduler Load Balancing Mechanism", Proc. of 14th IEEE Conf. on Computers and Communications, 1995.

[5] Ruud et al., "Ant-like Agents for Load Balancing in Telecommunications", Agnets'97, 1997.

[6] A.shaout and P.McAuliffle,"Job Scheduling using Fuzzy Load Balancing in Distributed System", Electronics Letters Vol.34, 1998.

[7] E.Bonabeau et al.,"Swarm Intelligence: From Natural to Artificial Systems", Oxford press, 1999.

[8] M.Dorigo et al.,"Ant Colony Optimization: A New Meta-Heuristic", proc. of 3th Fuzzy Sets and Systems, 1999.

[9] A.Y.Zamaya et al., "Observations on Using Genetic Algorithms for Dynamic Load balancing", IEEE Trans. on Parallel and Distributed Systems Vol.12, No.9, 2001.

[10] O.Babaoglu et al., "Anthill- A Framework for the Development of Agent based Peer to Peer Systems.", Procs. Of 22th IEEE Intl. Conf. on Distributed Systems, 2002.

[11] A.Montressor and H.Meling,"Load Balancing through Swarm of Autonomous Agents", Proc. of 1st ACM Int. Joint Conf. 2002.

[12] M.T.Islam, R.K, "A Parallel Ant Colony Optimization Algorithm for All-Pair Routing in MANET", Procs. of 3rd Intl. Symp. , 2003.

[13] Abubakar et al., "Evaluation of load balancing strategies", National Conference on Emerging Technologies, 2004.

[14] Daniel Grousa, Anthony T.," Non-Cooperative load balancing in distributed systems", Journal of Parallel and Distributing Computing, 2005.

[15] M.Amini and H.Deldari, "Grid Load Balancing using An Eco Systems of Ants", Procs. of 24th IASTED Intl. Conf. , 2006.

[16] Bing Qi," Ant Algorithm Based Load Balancing for Network Sessions", ICNC 3rd Intl. Conf. on Natural Computation, 2007.

[17] Siriluk et al.," An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment", IJCICE Vol., No.4. 2007.

[18] Parveen Jain and Daya Gupta," An algorithm for dynamic load balancing in distributed systems with multiple supporting nodes by exploiting the interrupt service", IJRTE, Vol. 1, No.1, May 2009.

[19] S.prasad et al., "Swarm Based Intelligent Routing for MANET", IJRTE Vol.1, No.1, 2009.

[20] Hamid and Sam," A Survey on Ant Based Algorithms for Mobile Ad-Hoc Networks", Intl. Conf. on Signal Processing Systems, 2009.

[21] Ali M.Alakeel," A Guide to Load Balancing in Distributing Computer Systems", IJCSNS, Vol. 10, No.6, June 2010.

[22] Ali et al.' "Load Balancing of Distributed Systems based on Multiple Ant Colonies Optimization", AJAS, Vol.7, No.3, 2010.

[23] Yan et al., "An Efficient and Stable Cluster System based on Improved Load Balancing Algorithm", Procs. Of IEEE 2010.

[24] Ali M.Alakeel, "A Fuzzy Dynamic Load Balancing Algorithm for Homogeneous Distributed Systems", World Academy of Science, Engineering and Technology, Vol.6, 2012.

[25] Dimple Juneja and Atul Garg," Collective Intelligence based Framework for Load Balancing of Web Servers", IJICT, Vol. 3 No.1, 2012.

[26] Atul Garg and Dimple Juneja, "A Comparison and Analysis of Various Extended Techniques of Query Optimization", IJICT Vol.3 No.3, 2012.

[27] Nitin and A.Tijare, "Dynamic Load Balancing in Parallel Computing based on Execution Time", IJARCSSE Vol.2 No.10, 2012.

[28] Dinesh and P.Venkatakishna,"Honey-Bee Behavior Inspired Load Balancing of tasks in Cloud Computing Environment", Applied Software Computing, Vol.13, 2013.

[29] Rajan Kumar and G.Sahoo,"Load Balancing using Ant Colony in Cloud Computing", IJITCS Vol.3, No.5, 2013.

[30] Shagufta and Niresh," Effective Scheduling Algorithm for Load Balancing using Ant Colony Optimization in Cloud Computing.", IJCSIT Vol.4, No.2, 2014.

[31] Preeti Kushwah,"A Survey on Load balancing Techniques using ACO Algorithms", IJCSIT Vol.5, No.5, 2014.