IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-3, Issue-1 E-ISSN: 2347-2693

Load Balancing of node in Network using Ant Colony Optimization

Vaishnavi Aher,Sayali Khairnar,Madhuri Shinde,Priyanka Shirole

Department of Computer Engineering,
Pune Vidhyarthi Griha's Collage of Engineering, Nashik, Maharashtra, India

www.ijcaonline.org

Received: Dec /16/2014 Revised: Jan/1/2015 Accepted: Jan/24/2015 Published: Jan/31/2015

Abstract— In this paper, we used algorithm for load balancing of workload among the various nodes of grid network by using Ant Colony Optimization (ACO). Load balancing is one in all the most difficult problems in networking that is need to distribute the dynamic work across multiple nodes to confirm that no single node is overloaded. The main contribution of our work is to balance the whole system load whereas making an attempt to maximize and minimize the parameter. Actually when two or more co-positioned ants exchange their knowledge, they extend their movement radius to a bigger domain and this causes better awareness of the environment. There is a similar idea which is inspired from the pheromone deposits from the ants while wandering. Other ants can pursue the ant by using this pheromone deposited. This idea is applied in most of ant colony optimization problems.

Keywords— Ant Colony Optimization (ACO), Pheromone updation table, Grid Networks, Load Balancing.

I. INTRODUCTION

A network is recognized by varied nodes that perform computation according to the requests of the client. As the requests of the client will be random to the nodes they will vary in amount and so the load on every node also can vary. Therefore, every node in a network can be unsteady loaded of tasks in keeping with the quantity of work requested by the clients. Load balancing is a technique to extend work between two or lot of computers, network links, CPU's, hard drives, or other resources, for getting optimal resource utilization, maximum throughput, and minimum response time.[1] Load balancing helpful once addressing redundant communications links, as an example, an organization might have multiple web connections guaranteeing network access even if one in all the connections ought to fail. A fail over arrangement would mean that one link is selected for traditional use, whereas the second link is used provided that the primary one fails. With load balancing, each links may be in use all the time. A device or program decides that which are the ready links to send packets on, being careful to not send packets on any link if it's unsuccessful. The ability to use multiple links at the same time will increase the available bandwidth. Major telecommunications firms have multiple routes through their networks or to external networks. They use a lot of refined load balancing to shift traffic from one path to a different to avoid network congestion on any specific link, and typically to reduce the value of transit across external networks or improve network dependability.

Ant Colony Optimization (ACO) is a relatively new computational and behavioral paradigm for resolution improvement and combination problems; it's supported the principles that management the behavior of natural systems. In this paper, Ant based algorithm for load balancing in Grid is projected. The analysis work aims on improving the approach ants search the most effective resources in terms of minimizing the time interval of every task and at identical time equalization the load on available resources.[4] Generally, Ant Colony Optimization (ACO) is a general-purpose heuristic algorithm, which can be used to solve different combinatorial optimization problems. In ACO, the search activities are distributed over artificial ants, which mimic the behavior of real ants. The advantages of that system are positive feed-back, distributed computation, and the use of a constructive greedy heuristic Positive feed-back refers to the power to speedy discovery of excellent solutions. The ACO is also a population –based approach in which parallelization can easily be achieved.[3]

II. PROPOSED ALGORITHM

A. *Nature of Real Ant*

Many aspects of the collective activities of social insects, like ants, self-organizing. This suggests that complicated group behavior emerges from the interactions of individuals who exhibit straightforward behaviors by themselves. Examples of these collective activities among ants finding food and building nests. The results of self-organization area unit world in nature, however surface from interactions primarily based entirely on native data. To realize this, self-organization depends on many components: (i) Positive feedback (ii) Negative feedback (iii) multiple interactions. The capabilities of a single ant are very limited compared to those of a colony. In some species, ants are principally blind and that they communicate poorly. However together, ants will establish the shortest route between a supply of food and their nest and with efficiency move the food to their home.

Ants communicate with one another through the employment of pheromones. As ants traverse a trail, they deposit pheromones. Pheromones are chemical substances that attract other ants and are deposited by ants on the ground as they travel. Ants move randomly, but once they encounter a secretion path, they decide whether or not or to not follow it. If they are doing therefore, they lay down their own secretion on the path still, reinforcing the pathway. The probability that an ant chooses one path over other increases proportional to the amount of pheromone present. The more ants that use a given path, the more attractive that trail becomes to subsequent ants.[2]

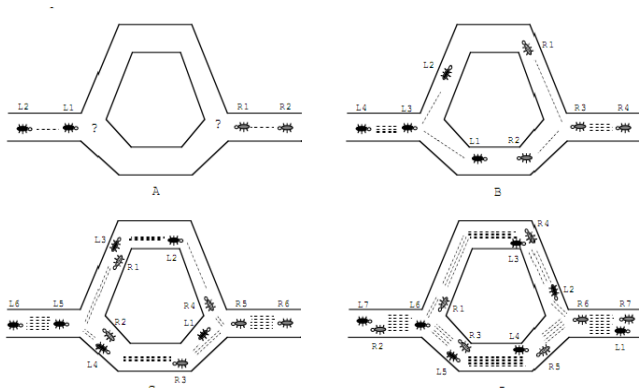

Figure 1: Real ants.

B. Ant Colony Optimization

In this system we use ACO for load balancing purpose, in an organization there are multiple clients they request to download the data from server simultaneously. The Congestion is occurring in network ultimately the response time is more. So, for proper distribution of the load among the node of the network, we use a intermediate system as a ACO server. whenever the client request for the file download from the server it first transfer to the ACO server, after arriving the request it check for response time from pheromone table and according to response time it given to the server ,in this way load on the network is distributed and time required for download is also reduce.

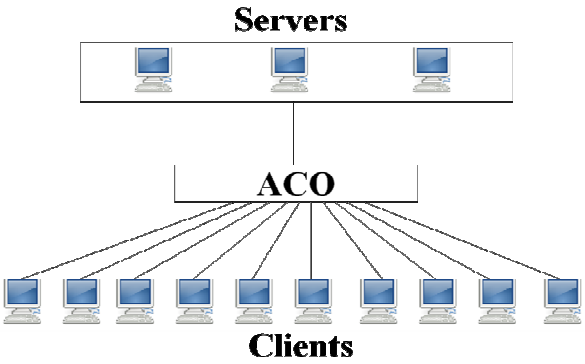

Figure 2:ACO System.

C. Equations

The In paper, the ACO algorithm is used for load balancing. The aims at efficiently distribute the load among the nodes and such that the ants never encounter a dead end for movements to nodes for building an optimum solution. The ants in our proposed algorithm will continuously originate from the Head node. These ants traverse through the network in such a way that they know about the location of underloaded or overloaded nodes in the network.

These Ants at the side of their traversal are going to be change a secretion table, which can keep a tab on the resources utilization by every node. We also have proposed the movement of ants in two ways in which kind of like the classical ACO, that are as follows:

1) Forward movement-The ants constant move within the forward direction within the network encountering overload node or under loaded node.
2) Backward movement-when an ant encounters an overloaded node in its movement when it has previously encountered an under loaded node then it will turn to previously under loaded node to check if the node is still under loaded or not and if it finds it fixed under loaded then it will reconstruct the work to the under loaded node. The vice-versa is also possible.
The main task is distributing the work among the node. The ants traverse the network, selecting nodes for their next step through the formula given below, where the probability *Pk* of an ant, which is currently on node r selecting the neighboring node s for traversal, is:

$$Pk\,(r,s) = \frac{[T(r,s)][\eta(r,s)]^{\wedge}\beta}{[T(r,u)][\eta(r,u)]^{\wedge}\beta} \qquad (1)$$

Where,
r=Current node.
s=Next node.
T=Pheromone concentration of the edge,
η=The move for the ant (if the move is from an under loaded node to overloaded node or vice-versa the move will be highly desirable)
β=Depends upon the relevance of the pheromone concentration with the move distance.

The ants continuously originating at an interval of $\triangle$t the overload incurred by network would increase as the number of paths followed by the ants would increase so would the cost for their maintenance and thus the network performance would take a beating. Therefore, we would keep their numbers in a limit.[5] We can keep their numbers in a limit by setting a suicide timer on the ant, which when reduce to zero the ant will terminate itself. The selection of timer value would base on the size and number of nodes in the network. The overload would depend too much on the interval $\triangle$t, the smaller the overload larger the overhead

and vice-versa. However, higher the number of ants more frequent would be the data changes and load balancing and thus network efficiency. For this reason, if we could limit the number of ants in the network for a good trade-off between the need to keep collecting fresh data and decrease variance, and they require to avoid congestion of the ants as well.[1]

D. Types of Pheromone

The ant has two types of pheromone for its movement. The types of pheromone specify the type of movements of the ant and would tell about the kind of node the ant is searching for. The two types of pheromones updated by the ants are as follows:

1) Foraging Pheromone (FP)-In ACO the ant uses foraging pheromones to explore new food source. In this algorithm the ant would lay down pheromone after encountering under loaded nodes for searching overloaded nodes. Therefore, after an ant comes up to an under loaded node it will try to find the next path through foraging pheromone. The formula for updating this pheromone would be:

$$FP(t+1) = (1-\beta eva)FP(t) + \Sigma nk=1 \Delta FP \quad (2)$$

Where,
βeva = Pheromone evaporation rate,
FP = Foraging pheromone of the edge before the move,
FP(t + 1) = Foraging pheromone of the edge after the move,
ΔFP = Change in the FP.

2) Trailing Pheromone (TP)-In a typical ACO the ant uses trailing pheromone to discover its path back to the nest. However, in this algorithm the ants would use this to find its path to the under loaded node after encountering overloaded node. Therefore, after an ant encounters an overloaded node it will try to trace back the under loaded node through the trailing pheromone. The formula for updating this pheromone would be:

$$TP(t+1) = (1-\beta eva)TP(t) + \Sigma n\ k=1 \Delta TP \quad (3)$$

Where,
βeva = Pheromone evaporation rate,
TP = Tracing pheromone of the edge before the move,
TP(t+1) = Tracing pheromone of the edge after the move,
ΔTP = Change in the TP.[1]

Before you begin to format your paper, first write and save the content as a separate text file. Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you?

Finally, complete content and organizational editing before formatting. Please take note of the following items when proofreading spelling and grammar:

III. RESULT ANALYSIS

A. Pheromone Table Updation

We replaced the routing tables in the network nodes by tables of probabilities which we will call 'pheromone tables', as the pheromone strengths are represented by these probabilities. Each node has a pheromone table for every possible destination in the network, and each table has an entry for each neighbor. For example, a node with four neighbours in a 20-node network has 19 pheromone tables with four entries each. One could say that an n-node network uses n different kinds of pheromones. The entries in the tables are the possibilities which influence the ants' selection of the next node on the way to their destination node. Figure 4 shows a feasible network configuration and a pheromone table. For example, ants tour from node 1 to node 3 have a 0.39 possibility of choosing node 2 as their next node, and 0.41 of choosing node 4. 'Pheromone laying' is represented by 'updating probabilities.

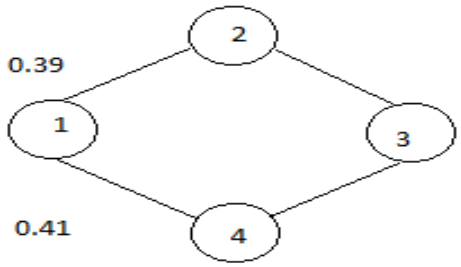

Figure 3: Using ants for network management.

Table I
Relation between Next node and Destination node.

		Next node	
		2	4
Destination node	2	0.85	0.05
	3	0.39	0.41
	4	0.05	0.85

Every time during the simulation, ants can be originated from any node in the network. Every ant has a random destination node. Ants travel from node to node, select the next node according to the probabilities in the pheromone tables for destination node. Arriving at a node, they update the possibilities of that node's pheromone table entries

corresponding to their source node i.e. ants lay the type of pheromone related to the node they were originated from. They modify the table to increase the probability pointing to their previous node.

B. Result

As per the graph we observe that average no of hops in each simulation are less in case of ant colony based algorithm . For the initial calls there is not much difference between, this is because the pheromone tables of each node in case of ant colony optimization technique are not normalized for initial calls made on the network. But as more calls are placed on the network the pheromone tables get normalized and there is not much variation in the pheromone tables, therefore the graph, in case of ant colony based algorithm, remains almost saturated. Hence from the results we inferred that ant colony based algorithms perform better.

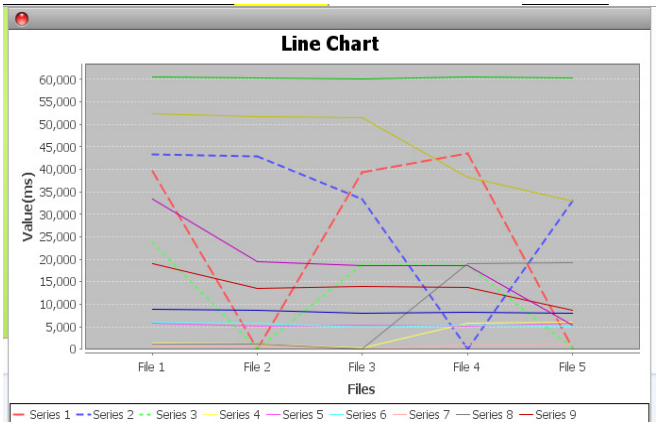

Figure 4:Download time of files in ms.

IV. CONCLUSION

This is a improved approach of ant colony optimization that has been applied from the perspective of grid network systems with the main aim of load balancing of nodes. In this paper, ant colony optimization algorithm depend on Load balancing has been proposed to initiate the service of load distribution among the network architecture. The main benefit of this approach lies in its detections of overloaded and under loaded nodes and thereby performing operations based on the identified nodes. In this approach the task of identification of nodes perform by the ants and tracing its path consequently in search of different types of nodes. The pheromone table is used to show overloaded and under loaded node. The pheromone update mechanism has been proved as a efficient and effective tool to balance the load. This alternation approach supports to minimize the make span of the cloud computing based services and portability of servicing the request also has been converged using the ant colony optimization technique. Our objective for this paper is to develop an effective load balancing algorithm using Ant colony optimization technique to maximize or minimize different performance parameters like CPU burden, Memory capacity, Delay or network load for the clouds of different sizes.

ACKNOWLEDGMENT

The people we would to thanks are many for their help in presenting this project seminar. we specially want to thanks Prof. M. T. JAGTAP ,HOD COMPUTER DEPT And our project guide Prof. J.Y.KAPADNIS whose advise have inspired and shaped much of the content of this project also guiding us on this project work and spending hours working to improve the content of this project. Also we want to thanks to all the staff members who help us to improve. So majority of our credit goes to all staff members. We want to thanks to our parents for having the wisdom to share their thoughts with us, to guide us and present ourself to represent this project and to get the work done on time with more energy and in easy way. We also want to thanks our friends for their patience and encouragement. Finally last but not the list we would to own our heartiest thanks to the great almighty for giving us all the strengths in life.

REFERENCES

[1] Kumar Nishant, Pratik Sharma, Vishal Krishna, Chhavi Gupta and Kuwar Pratap Singh Nitin and Ravi Rastogi," Load Balancing of Nodes in Cloud Using Ant Colony Optimization", *2012 14th International Conference on Modelling and Simulation*", **978-0-7695-4682-7/12, © 2012** IEEE.

[2] Sandip Kumar Goyal , Manpreet Singh," Adaptive and Dynamic Load Balancing in Grid Using Ant Colony Optimization", *International Journal of Engineering and Technology (IJET),* ISSN : **0975-4024**, Vol 4 No 4 Aug-Sep **2012**.

[3] M. LAl-Dahoud Ali and Mohamed A. Belal," Multiple Ant Colonies Optimization for Load Balancing in Distributed Systems", *ICTA'***07**, April **12-14,** Hammamet, Tunisia.

[4] A. D. Ali, and M. A. Belal, "Multiple ant colonies optimization for load balancing in distributed systems," in *Proc. Inter. Conf. (ICTA'07),* **2007**.

[5] R.A. Arnous, H.A. Arafat and M.M. Salem, Improving the Load Balancing within the Data Network via Modified AntNet Algorithm, Proceedings of the 5th International Conference on Information and Communication Technology, pp. **189-195, 2007.**