2012 International Conference on Information and Network Technology (ICINT 2012)
IPCSIT vol. 37 (2012) © (2012) IACSIT Press, Singapore

Enhancement of Data Service by Finding the Shortest Path of the Server using Swarm Intelligence

G. Venkatesh +, S. Vinoth and G. Nandha Kumar

Anjalai Ammal Mahalingam Engineering College

Abstract. In Network, the quality of a service reduces because of the delay in data service owing to the inability to reach the server quickly. So there is a need for an optimum method which provides easy access to the server. This paper presents the novel technique, Swarm Intelligence (SI). We bring out the Ant Colony Optimization (ACO), a behavior under SI which is used to find the shortest path of the server and also provides solutions for getting response without reaching the server and data delivery.

Keywords: swarm intelligence (SI), ant colony optimization (ACO).

1. Introduction

Swarm Intelligent systems are typically made up of a population of simple agents or boids interacting locally with one another and with their environment. The inspiration often comes from nature, especially biological systems. The agents follow very simple rules and although there is no centralized control structure dictating how individual agents should behave locally and to a certain degree, random interactions between such agents lead to the emergence of "intelligent" global behavior, unknown to the individual agents. Natural examples of SI include ant colonies, bird flocking and bee colony. The application of swarm principles to robots is called swarm robotics, while 'swarm intelligence' refers to the more general set of algorithms. 'Swarm prediction' has been used in the context of forecasting problems. Ant Colony is the efficient optimization process among the swarm intelligent systems for the quick delivery of data & finding the shortest path to reach the server. Ant Colony Optimization (ACO) relies on the behaviour of real ant to find the server. Here we use the behaviour of ants which finds the shortest path through Pheromonic Acid (Finding the Server by Shortest Path), Roaming ant which find the nearest food source after finding the shortest path (Getting Response without Reaching the Server) and collect the food source (Data Delivery).

2. Comparison of Ant Colony with other Colonies

2.1. Bee colony

Bee finds the food source based on the movement of sun, location of nest and flower. Initially the forager bee chooses the food source by finding the angle between the nest, sun and flower. Then it will convey the location of the flower to the scout bee through waggle dance. Scout bee is aware of waggle dance and these bees are responsible for the collection of honey. Forager bee's role is to find the location of food source and schedule the work of the scout bee to collect the honey. This optimization needs a lot of scheduling process and predictions. As a result it will increase the burden in network, whereas in Ant colony, scheduling and predictions are not necessary.

2.2. Bird colony

+ Corresponding author. Tel.: + 919894870583.
E-mail address: venki26blue@gmail.com.

Normally birds fly in "V" Shape. The V-shaped formation that geese use when migrating, serves two important purposes: First, it conserves their energy. Each bird flies slightly above the bird in front of it, resulting in the reduction of wind resistance. The birds in the front take turns, falling back when they get tired. In this way, the geese can fly for a long time before they can stop for rest. The second benefit of the "V" formation is that it is easy to keep track of every bird in the group. Fighter pilots often use this formation for the same reason. This optimization process concentrates only on the efficient way to reach the server and does not worry about the data delivery.

2.3. Better optimization process (ant colony optimization)

The above two colony is suitable only to reach the server and is not much efficient for data delivery. Ant colony rectifies on the two problems mentioned above. Here, the real ant behavior is mapped to data service and efficient way to reach the server. ACO is implemented using simple prediction by referring to the maximum count in the router, to find the shortest path and to efficiently deliver data to the user.

3. Behavior of Real Ant

Real ants are capable of finding the shortest path from the food source to the nest without using visual cues. Also, they are capable of adapting to the changes in the environment, for example finding the new shortest path once the old one is no longer feasible due to a new obstacle. Consider the following figure in which ants are moving in a straight line which connects the food source to the nest:

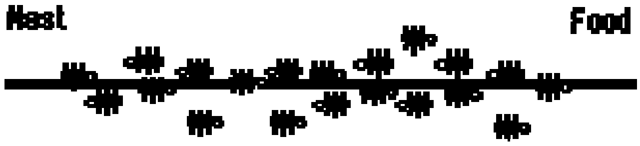

Fig. 1: Identifying the food source

It is well known that the main means used by ants to form and maintain the line is a pheromone trail. Ants deposit a certain amount of pheromone while walking and mostly each ant prefers to follow the direction, which is rich in pheromone rather than a poorer one. This elementary behavior of real ants can be used to explain how they can find the shortest path which reconnects a broken line after the sudden appearance of an unexpected obstacle that has interrupted the initial path.

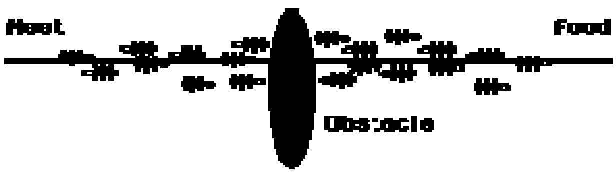

Fig. 2: Obstacle in path

In fact, once the obstacle has appeared, the ants which are just in front of the obstacle cannot continue to follow the pheromone trail and therefore they have to choose between turning right or left. In this situation we can expect half of the ants to choose to turn right and the other half to turn left. The very same situation can be found on the other side of the obstacle.

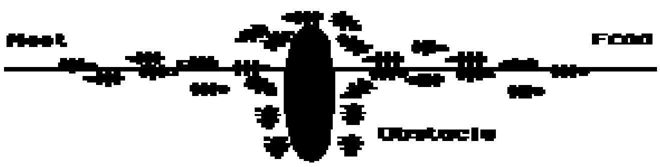

Fig. 3: Choosing the shortest path

It is interesting to note that those ants which by chance, choose the shortest path around the obstacle will more rapidly reconstitute the interrupted pheromone trail compared to those which choose the longer path. Hence, the shortest path will receive a higher amount of pheromone in the time unit and this in turn, will cause a higher number of ants to choose the shortest path. Due to this positive feedback (autocatalytic) process, all the ants will choose the shortest path very soon.

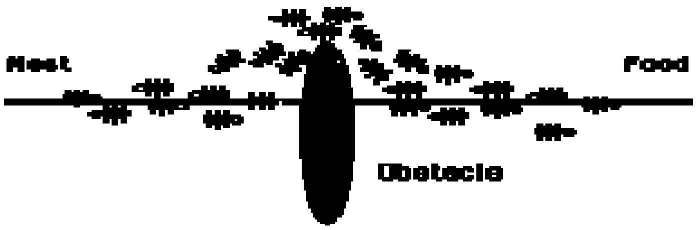

Fig. 4: Following the shortest path

The most interesting aspect of this autocatalytic process is that finding the shortest path around the obstacle seems to be an emergent property of the interaction between the obstacle shape and ants' distributed behavior: Although all ants move at approximately the same speed and deposit a pheromone trail at approximately the same rate, it is a fact that it takes longer to contour obstacles on their longer side than on their shorter side which makes the pheromone trail accumulate quicker on the shorter side. It is the ant's preference for higher pheromone trail levels which makes this accumulation still quicker on the shorter path. The problem with the way in which ants find the shortest path naturally involves the addition of a new shortest path after the ants have converged to a longer path. Because as the pheromonic has been built on the older longer path the ants will not take the newer shortest path. This would be a major problem if the algorithm is applied to dynamic problems in computing. However a simple solution can be created. If the virtual pheromone evaporates then the longest path will eventually be abandoned for the shortest one by the roaming ant. This is because some ants will still take the shortest path by chance (periodically roaming) and so pheromone will build up and eventually overtake the longer one.

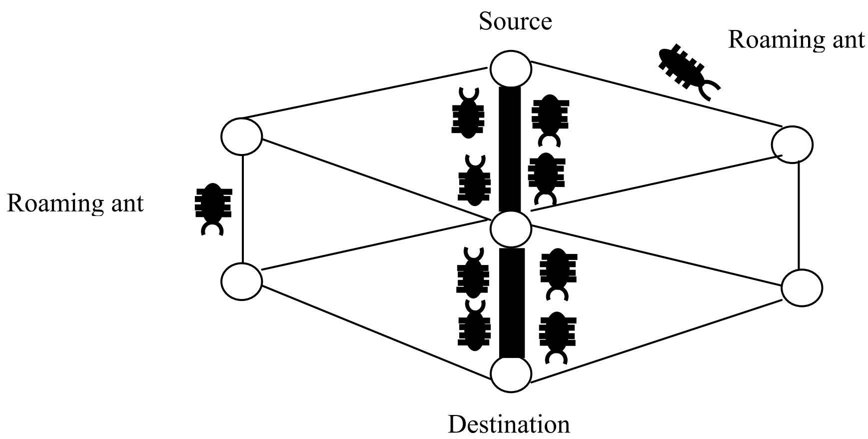

Fig. 5: Finding new shortest path by the roaming ant

4. Ant Colony for Enhancement of Data Service

Mapping of this behaviour of real ant to data service will reduce the time delay in request and response from client and server. By studying the real ant behavior, three things should be followed while processing and delivery of data.

4.1. Shortest path to reach the server (shortest path to reach the food source by the ant with the help of pheromonic acid)

Clients are connected to the server through a lot of nodes, thus the shortest path can be found out by selecting the path which has lesser number of nodes. This simple logic can be used by the artificial ant to find the shortest path. The following diagram will explain how the artificial ant finds the shortest path. When the user is connected with the network, the ants will travel in different paths and reach the server. As described above, client is connected to the server through nodes like Routers, switch etc. On reaching each and every node, the ant updates the count value in the router if the ant travels through it. Ant can find the shortest path by comparing the count value in the router (count value increases if more ants use the same path). While traveling through the router it will update the pheromone count in the router. If an obstacle is caused during the travel of ant then the ant will choose the nearest node (as each and every node is connected to the nearest node) that is suitable to reach the server by referring to the next maximum value in the pheromone count. Thus the shortest path is found out by the ant. The client request and server response are transmitted through the path which is found out by the ant. The diagram shows the single client request to the server through virtual Ant.

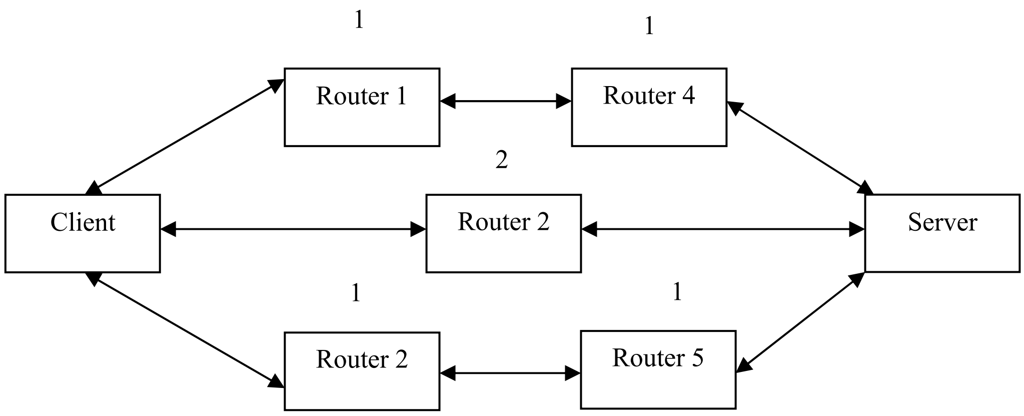

Fig.6: Shortest path to reach server by the artificial ant.

4.2. Data delivery without reaching the server (Roaming Ant which finds the nearest food source after finding the shortest path)

The ant finds the shortest path to the food source. After finding the shortest path, there may be a food source which is available nearest to the nest (newer shortest path) compared to the path selected as shortest one (older shortest path). Then the ant colony is not efficient for the shortest path problem. So to avoid this ant follows one simple rule to find the new shortest path. There may be roaming ants which role is always to wander around to find the new food source or the new shorter path. Similarly, if another user's request in the request queue is same as the previous one, then it can directly obtain the response from the response queue without reaching the server. This is also done by the artificial ant by analyzing the request queue. When the requested data is available in the response queue then the roaming ant will find that out and deliver the data. This will fasten the delivery of data and reduce the burden of the server.

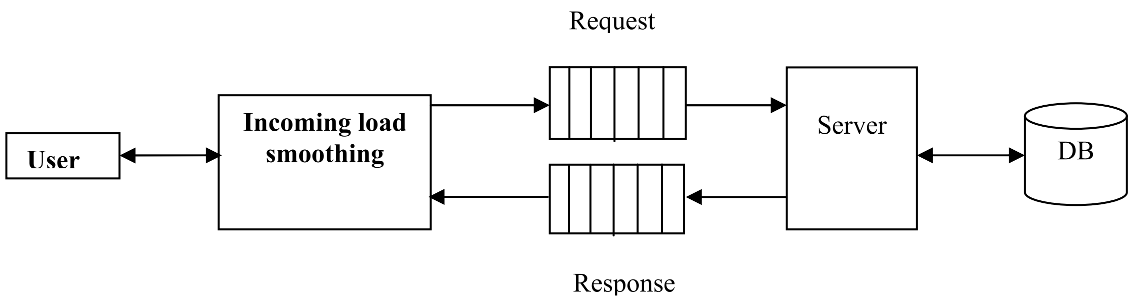

Fig. 7: Finding the difference between food and obstacle.

4.3. Data delivery (collecting the food and storing it in the nest)

This is a simple process for the ant to collect the food source and reach the nest. The ant follows the same path to reach the nest as it reaches the food source earlier. Thus the data delivery is easy if data source is easily available. This is made possible by artificial ant which is used to find the shortest path.

5. Architecture

The following architecture describes the data processing and delivery from client to server by implementing the ant colony optimization. The request sent by the client is tested first for incoming load smoothing where the irrelevant requests and exceptions are handled before reaching the server (Identifying the difference between the food and obstacle by the ant). Then the request is checked in the request queue. If it is available there, then the data is delivered from the response queue directly (Finding the newer food source which is nearer than the earlier one). If the data is not available in the queue then the request is passed to the server through the shortest path by comparing the pheromone count in the router (Finding the shortest path by ant). The data is delivered to the client (food is stored in the nest).

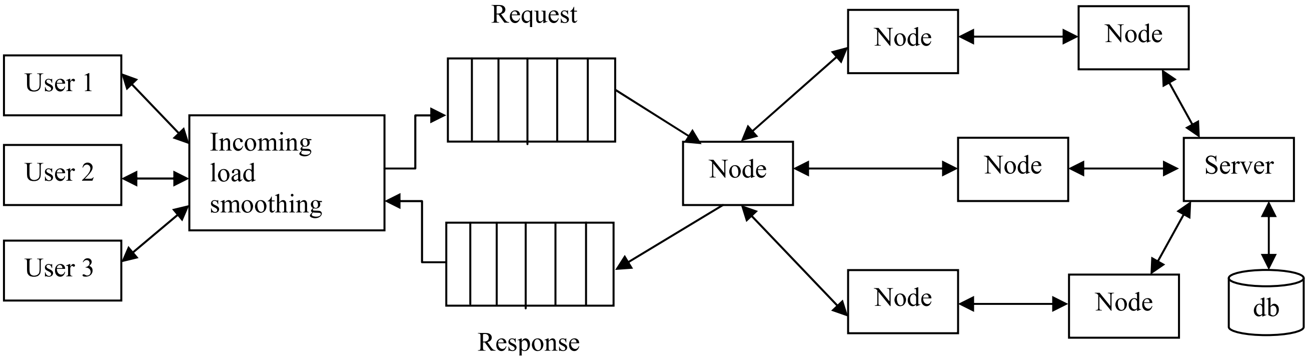

Fig. 8: Architecture.

6. Conclusion

This paper describes how the Ant colony optimization is suitable to reduce the service delay in data. Ant colony optimization efficiently handles the shortest path and reduces the burden of the server by delivering the data directly if it is available in the response queue. So this optimization process will provide the better way to increase the data service. This process efficiently handles the obstacle in the path and will provide the data delivery. However, in the future further enhancement of this approach can be implemented for more dynamic data. In the response queue the older data is available, which is not a suitable data source for the more dynamic data in the server. In the future, this optimization process will be developed for highly dynamic data.

7. Acknowledgements

First and foremost we would like to thank our **Principal Dr. A. Arokiasamy** who has been very instrumental in guiding us throughout the research work. **Mr. K. Velmurugan (HOD of CSE)** who was the main source of inspiration for making us do the project**, Mr. R. Lakshmanan (Computer Technologist , iTech)** who has contributed so much for making this paper a successful one. We would be failing in our duty if we don't thank **Mr. R. Roger Samson and Ms. S. Farjana Farvin.(Asst. professors)** and all our CSE department staff members. Thank you All.

8. References

[1] Beckers R., Deneubourg J.L. and S. Goss (1992). Trails and U-turns in the selection of the shortest path by the ant Lasius niger. *Journal of theoretical biology*, **159**, 397-415.

[2] Van der Zwaan, S. and Marques, C. 1999. Ant colony optimization for job shop scheduling. *In Proceedings of the Third* Workshop on Genetic Algorithms and Artificial Life (GAAL 99).

[3] J.L. Hellerstein, Y. Diao, S. Parekh, and D.M. Tilbury, Feedback Control of Computing Systems. John Wiley and Sons, 2004. 566 IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 4, APRIL 2011 Fig. 16. Normalized average data accesses.

[4] K.D. Kang, J. Oh, and Y. Zhou, "Backlog Estimation and Management for Real-Time Data Services," Proc. 20th Euromicro Conf. Real-Time Systems, 2008.

[5] K.D. Kang, S.H. Son, and J.A. Stankovic, "Managing Deadline Miss Ratio and Sensor Data Freshness in Real-Time Databases," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 10, pp. 1200-1216, Oct. 2004.

[6] M. Amirijoo, J. Hansson, and S.H. Son, "Specification and Management of QoS in Real-Time Databases Supporting Imprecise Computations," IEEE Trans. Computers, vol. 55, no. 3, pp. 304- 319, Mar. 2006.