ACO ALGORITHM FOR LOAD BALANCING IN SIMPLE NETWORK

Mrs Minal.Nerkar
Faculty of Dept of Computer Engineering.

Bhagyashree Kale,Shivani Bhutada,Chetan Darshale,Poonam Patil
Dept of Computer Engineering.
Savitribai Phule Pune University,India
AISSMS IOIT
Maharashtra,Pune,India

Abstract

Cloud computing is a technic of providing networked,on-line,on demand services to the user which pay on the basis of usage.Cloud computing has gained most importance in past years and will be one of the most required technic in future.There are many issues related to cloud such as scalability,performance ,secu-rity,etc..Load balancing one of the most important is-sue of cloud computing,which is the process of distri-bution of workload among different nodes or proces-sor.The purpose of load balancing is to improve the performance of a cloud environment through an appro-priate distribution strategy.In this paper we are demon-strating an algorithm which will be helpful for balanc-ing dynamic load on cloud an developed using the basic idea of ant colony optimization.

1. INTRODUCTION

1.1. Definition of Load Balancing

Load balancing is dividing the amount of work that a computer has to do between two or more computers so that more work gets done in the same amount of time and, in general, all users get served faster.

A Load Balancer allows users to intelligently distribute traffic to a single IP across any number of servers using a number of different protocols. This means that the processing load can be shared across many nodes, rather than being limited to a single server increasing performance during times of high activity. It increases the reliability of your web application and allows you to build your application with redundancy in mind. If one of your server nodes fails, the traffic is program-matically distributed to your other nodes without any interruption of service.

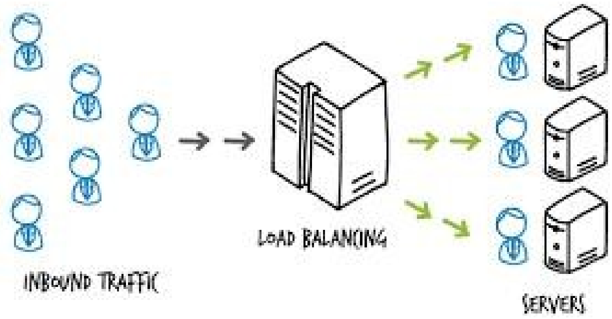


Figure 1. Load Balancer

1.2. When and Why to use Load Balancing

Load balancing, by its very nature, is the solution for more than one problem. You can use load balancing to keep your site up through traffic spikes, or grow with you as your site gains popularity.The two most common uses for Load Balancing:

ISSN: 2278 – 1323 All Rights Reserved © 2015 IJARCET

Limiting your points of failure
Failover and redundancy:
By limiting your available points of failure, you increase your uptime. If you load balance between two or more identical nodes, in the event that one of the nodes in your cluster experiences any kind of hardware or software failure the traffic can be redistributed to the other nodes keeping your site up. If you are extremely concerned with uptime, load balancing between two identical nodes that can independently handle the traffic to your site al-lows for failure in either one, without taking your site down.

Load Distribution
Growing beyond a single server configuration: As your site gains popularity you will outgrow the power of even the most robust servers, and require something stronger than a single server configura-tion. Upgrading from a single server, to a dual server configuration (1 Web server, 1 Database server) configuration will only allow for so much growth. The next step is to combine the power of multiple servers with Load Balancing.

1.3. Motivation of Proposed System

In computer science and operations research,the ant colony optimization algorithm(ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs.In the natural world ants (initially) wander randomly and upon finding the food return to their colony while laying down pheromone trails.If an-other ant find such a path they are likely not to keep traveling at random,but instead follow the trail return-ing rein- forcing if they eventually find the food. When one ant finds a good path(i.e short path) from the colony to a food source,other ants are more likely to follow that path and positive feedback eventually leads to all ants following a single path. The idea of ACO is mimic this behavior with ”‘simulated ants walking around the net-work representing the problem to solve”.

This paper will focus on a ACO algorithm and its structure and how it can be used as load balancer .In Section 2, the phases will be introduced in the context of load balancing for file downloading system which can be extended to cloud.In Section 3 will have experimental results and analysis.In Section 4, will have conclusion and future work.

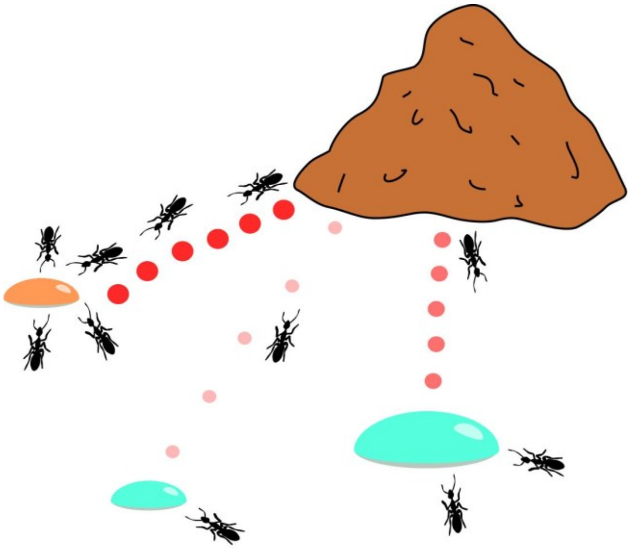


Figure 2. Ants And Their Behavior

2. PROPOSED SYSTEM

The proposed system is a demonstration that can be extended to the cloud.In the proposed system we are developing a ACO server which handles multiple client request for the file and processes client request by using ACO algorithm.The scope of the system is limited to a particular type of download and that is File Download-ing from Virtual Nodes.

2.1. Block Structure Of ACO Server

The ACO Server has basically has the main database and ant objects. The ant objects has various parameters like pheromone values,next neighbour,state,etc. The ant object is send to the virtual nodes.The updated values of the ant object are stored in the database.The database typically has the virtual nodes ip address,port number,their capacity,the request serve by which node,their state,etc.

2.1.1. Pheromone Updating Formula. The ant will use two types of pheromone for its movement. The type of pheromone being updated by the ant would signify 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:

The figure shows how the ant moves from the overloaded node to underloaded node. The overall impact on ant colony optimization algorithm depends on the this parameter greatly.It depends on threshold value.If incoming file request is less than threshold

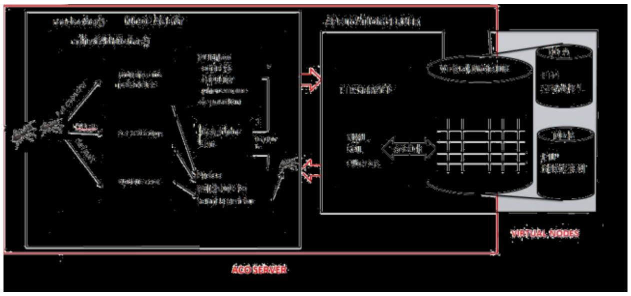


Figure 3. ACO Server Structure

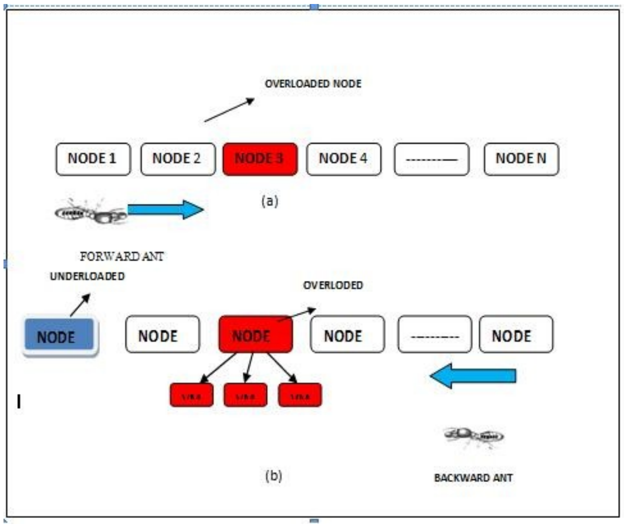


Figure 4. Depicting movement of Ant

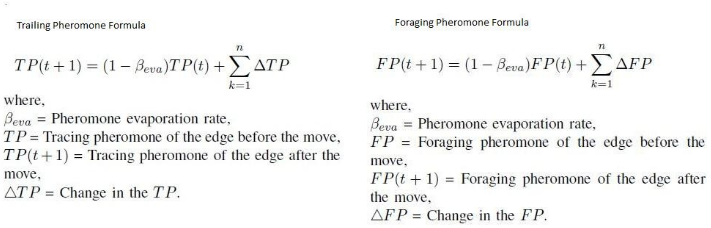
value then its an underloaded node and else it is over-loaded node.

Foraging Pheromone(FP)

In a typical ACO the ant uses foraging pheromones to explore new food sources. In our algorithm the ant would lay down foraging 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.

Trailing Pheromone(TP)

In a typical ACO the ant uses trailing pheromone to discover its path back to the nest. However, in our algorithm the ants would use this to find its path to the under loaded node after encounter-ing overloaded node. Therefore, after an ant en-counters an overloaded node it will try to trace back the under loaded node through the trailing pheromone.



Trailing Pheromone Formula

$$TP(t+1) = (1-\beta_{eva})TP(t) + \sum_{k=1}^{n} \Delta TP$$

where,
β_{eva} = Pheromone evaporation rate,
TP = Tracing pheromone of the edge before the move,
$TP(t+1)$ = Tracing pheromone of the edge after the move,
$\triangle TP$ = Change in the TP.

Foraging Pheromone Formula

$$FP(t+1) = (1-\beta_{eva})FP(t) + \sum_{k=1}^{n} \Delta FP$$

where,
β_{eva} = Pheromone evaporation rate,
FP = Foraging pheromone of the edge before the move,
$FP(t+1)$ = Foraging pheromone of the edge after the move,
$\triangle FP$ = Change in the FP.

Figure 5. Formula

2.2. Flow of Proposed System

There are three phases of the proposed system.The our as follows:

Client Phase

1. The client will initially login in the system.
2. The client will decide what operation to per-form i.e download,upload,or viewing a file.
3. If it is download operation the client will se-lect the files to be downloaded.
4. And the request will be send to the ACO server.
5. And client will be waiting for the files to be downloaded.

ACO Server Phase

1. The administrator will login in the system.
2. Will select the algorithm to handle client re-quest i.e ACO Algorithm or Simple Load Bal-ancing Algorithm
3. If its ACO Algorithm will perform the follow-ing:
   – The server will initialize the ant object

– The server will send the ant object to virtual node and will update the pheromone values based on virtual node state(underloaded,overloaded,moderate).

– The ant will give all information to the ACO Server.

– The ACO server on receiving the information will send the intended virtual node request to serve the client request.

4. Else will server request on first com first base.

Virtual Node Server(FTP Server)

1. Initially the Virtual node will request the ACO server to add it in network.
2. It will send the capacity and its port number.
3. On receiving the ACO server request to serve a client it will serve the client request.

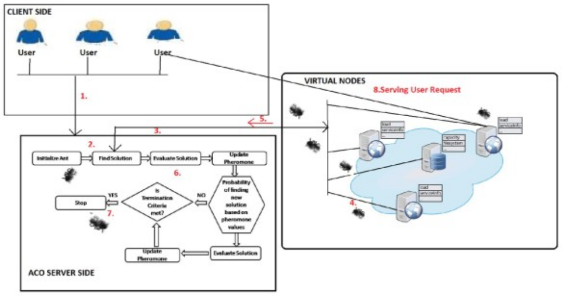


Figure 6. Flow Of System

3. EXPERIMENTAL RESULTS

The graph shows the analysis of number request to the time ratio at which request is completed.

It is observed that initially ACO algorithm is not effective but once the pheromone values are up-dated after a period of time it shows effective re-sults.

The ACO Algorithm initially behaves as first com first serve.

The ACO Algorithm is information based and shows better results if more information there.

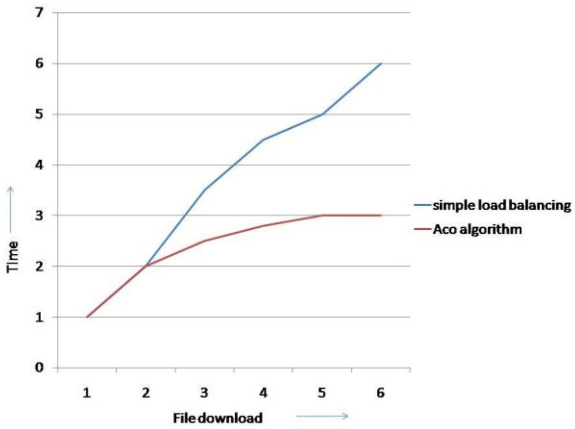


Figure 7. Analysis Of Algorithm

4. FUTURE WORKS AND CONCLUSION

4.1. Future Works

The implementation of ACO can done in Heteroge-neous Network

Implementation of ACO in Cloud Network

4.2. Conclusions

This is a modified approach of ant colony optimiza-tion that has been applied from the perspective of net-work systems with the main aim of load balancing of nodes. This modified algorithm has an edge over the original approach in which each ant build their own in-dividual result set and it is later on built into a complete solution.

References

[1] S. Banerjee, I. Mukherjee and P.K. Mahanti, Cloud Computing Initiative using Modified Ant Colony Frame-

work, World Academy of Science and Technology, 56, pp.221-224, 2009.

[2] Y. Li, A Bio-inspired Adaptive Job Scheduling Mecha-nism on a Computational Grid, International Journal of Computer Science and Network Security, 6(3B), pp. 1-7, 2006.

[3] Z. Zhang and X. Zhang, A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation, Proceedings of the 2nd International Conference on Industrial Mechatronics and Automation, pp. 240-243, 2010.

[4] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nu-aimi and Jameela Al-Jaroodi, A Survey of Load Bal-ancing in Cloud Computing: Challenges and Algo-rithms , 2012 IEEE Second Symposium on Network Cloud Computing and Applications, 978-0-7695-4943-9/12, 2012 IEEE.

[5] Abhijit A. Rajguru, S.S. Apte, A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters Interna-tional Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-1,Issue-3, August 2012.

[6] Bhaskar Prasad rimal ,Eunmi choi,Ian Lumb,A Taxonomy and Survey of Cloud Computing system , Proceedings of the Fifth International Joint Conference on INC, IMS and IDC, 978-0-7695-3769-6/09 2009 IEEE

[7] S. Banerjee, I. Mukherjee and P.K. Mahanti, Cloud Computing Initiative using Modified Ant Colony Framework , World Academy of Science and Technology, 56, pp.221-224, 2009.

[8] Tharam Dillon, Chen Wu and Elizabeth Chang,Cloud Computing: Issues and Challenges, Proceedings of the 24th IEEE International Conference on Ad-vanced Information Networking and Applications, DOI 10.1109/AINA.2010.187,1550-445X/10 2010 IEEE.