G. T. Chavan., M. A. Mahajan / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 2, March -April 2013, pp.356-359

Load Balancing in P2P networks using DHT based systems and Ant based systems: A Comparison

G. T. Chavan, M. A. Mahajan
Asst. Prof. Sinhgad College of Engineering, Pune
M.E. Student Sinhgad College of Engineering, Pune

Abstract

The heterogeneity in peer to peer (P2P) networks can be its advantage as well as disadvantage. The heterogeneity may be about amount of storage, computing power, connectivity etc. But P2P systems make it possible to harness resources such as the storage, bandwidth and computing power of large populations of networked computers in a cost effective manner. There are many approaches to balance these systems in a cost-effective way. Here we are giving focus on two main approaches; the Distributed Hash Table (DHT) based systems and Ant based systems. Both approaches are having its own advantages and shortcomings.

Keywords—load balancing, p2p networks, DHT, Ant based Systems

I. INTRODUCTION

Peer to Peer network:-

Peer to peer network is way of implementing the networking between the computers where all are equivalent in nature i.e. all share equivalent responsibilities of processing data. A peer to peer network can also be created in ad-hoc connection; where a couple of computers connected via a Universal Serial Bus to transfer files, or it can also be a permanent network like a network setup in a small organization.

Peer to peer systems doesn't have any centralised control or any hierarchy in operation, so peer to peer systems are distributed systems where the software running at each node is equivalent in functionality.

Load Balancing:

Load balancing is sharing the work between two or more computers, CPUs, network links etc. The load balancing is done to maximize throughput- by getting the work done from all the resources available in the system. As many resources are working simultaneously the resource utilization will be higher and the response time will gradually minimize. Load balancing can also be useful when dealing with redundant communications links. To achieve balancing first the nodes should be available for processing and the different routes between them should also be searched.

Distributed hash tables (DHTs) are example of decentralized distributed systems. DHTs provide a lookup for the data items similar to a hashing where (*key*, *value*) pairs are stored in the DHT, and any node can efficiently get the value associated with a given key. Responsibility of the mapping for keys to values is divided among the nodes, in such a way that a change in mapping causes a minimal amount of disruption. Figure 1 shows the structure of the DHT where the data items are combined with the hash function and the result generated will be stored in the node as a (key, Value) pair.

DHTs form an infrastructure that can be used to build P2P networks

II. DHT WORKING

In the DHT based systems each data item which is to be stored on the computers is having its unique id. The identifier space is partitioned among the nodes on

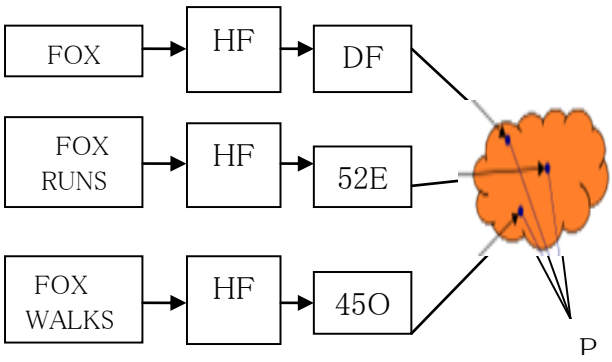

Fig.1. The structure of Distributed Hash Table.

the system, called as zone. Every node on the system is responsible for storing all the items that are mapped to an identifier in its portion of the space. The data items can be accessed and stored on the zone by using a set of functions i.e. put(ID, item): stores the item associated with ID, and get(ID): retrieves the item corresponding to ID. The load balancing is done using the virtual servers. The virtual server is part of the node. A single node can be partitioned into multiple virtual servers. A virtual server is also responsible for the identifier

space. As the node is a combination of multiple virtual servers it is responsible for multiple identifier spaces. The basic advantage of using virtual servers is that when the node gets overloaded at that time a virtual server can be transferred to any other node making the source node lighter. This mechanism is supported by DHT by making two calls leave source node and join destination node. This gives advantage over the scheme of having only one virtual server per node where a node can transfer his load to its neighbours only.

There are multiple ways to do load balancing in the DHT based systems, some of which are explained below.

The first method is having one to one mapping. In this scheme the light node will select any ID randomly and will start searching for the node who is responsible of that ID. When the target node is identified, its load is checked. If the target node is heavy then the light node will initiate the transfer operation and will accept the load of the heavy node till its threshold value. The advantage of the scheme is that it is simple to implement; the light node is doing probing not the heavy node so the heavy node is free from the Burdon of probing. The second method is having the mapping of one to many. In this scheme one directory maintains the list of light nodes. The heavy node sends information of its target load and the load of every virtual server it has to the node which maintains the directory.

Then the directory node will search for the best possible light nodes that will accommodate the virtual servers of the heavy nodes and then will do the transfer accordingly. This procedure is repeated till all the heavy nodes will become light nodes. The third scheme is many to many. In this scheme there is pool of virtual servers, where every node will submit the list of virtual servers to it. Then from the pool the heaviest virtual server is selected and transferred to the light node by making sure that the destination node will not become heavy. If there is no node available to transfer the virtual server then from the available list of light nodes, a virtual server is selected and transferred to the pool and the heaviest virtual server is transferred to the node.

Disadvantages of the systems based on DHT are:

The disadvantage of the systems based on the DHT is that in these systems the processing required for the load collection and the load transferred are not considered. Some systems are centralised and some of the systems are decentralized. The centralised systems can further be classified as full centralized and semi-centralized. In the full centralized systems one central node is responsible for storing the load conditions and selecting the destination for the load transfer. In semi-centralized the multiple central nodes are responsible for the storing the load conditions and selecting the destination for the load transfer for the local systems. Although the centralized system provides good load balancing but these system are immune to single point failures which leads to unbalanced system very easily.

The decentralized systems will not affect form the the single point failure problem but they suffer from load information collection overhead. Due to the decentralized nature it needs more time for convergence.

III. ANT COLONY OPTIMIZATION

An Ant is a simple creature of nature. Ants always work in a group, usually called as colony. Ants discover the shortest path to the food source from its nest. For this ants use indirect communication method called as stigmergy. Ants achieve stigmergic communication by using a chemical substance called as pheromone. While searching for the path to the food source ants lay down the pheromone on the path which is sensed by the other ants. Ants will make the decision depending upon the concentration of the pheromone on the path; the more the concentration high is the probability to select the path. Computer scientist transformed this model of collective intelligence into the computer algorithms. This emerging field is called as Ant Colony Optimization (ACO). In ACO analogous to the biological ants there is society of mobile agents. These mobile agents communicate depending upon the pheromone value which is nothing but the entries in the routing table made by the artificial ants. The ACO can be applied to various combinatorial problems like asymmetric travelling salesman problem, graph coloring problem, vehicle routing problem etc. To simulate the behaviour of an ant ACO uses different data structures like to simulate laying and sensing of pheromone simple procedures are used. For example while doing migration from one node to another node, an artificial ant emulates laying of pheromone by updating the routing table entries in a node.

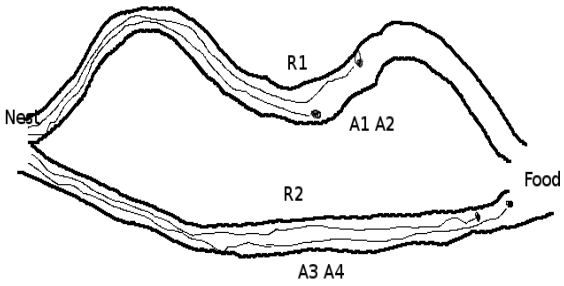

Fig. 2. Ant Colony Optimization.

Figure 2 is showing the typical working of ant colony optimization. In this network sis nodes are present out of which one node acts as nest and one node acts as food source. Between the nest and food source there are 4 more nodes present. There are 4 named as A1, A2, A3, A4 ants initially foraging for food-seeking. There are multiple paths present between nest and food source; out of which two paths namely R1, R2 are shown in the figure. A1, A2 randomly select the path R1 and A3, A4 selected randomly R2. While travelling through the path ants will deposit pheromone on the path. When A3 and A4 will reach the food source before A1 and A2 since R2 is having shorter length than R1. For their return way two paths are present R1 and R2, A3 and A4 will select R2 since the pheromone concentration on R2 is more than R1(A1 and A2 are still not reached food source). The further ants will sence the pheromone concentration on the two paths and will select the oath having more concentration. In this case it will be R2 since it has more probability over R1.

What are the advantages to use ACO?

1) Routing Information

when the Ant Colony Optimization technique is used the overhead of keeping the routing information is reduced. Unlike in other networks to maintain the routing table the needs the information from its neighbours to get the information of their neighbours. This overhead is not present in the ACO. In ACO each path is parallely identified and the routing table are also built randomly.

2) Routing Overhead

In the networks when the routing table gets updated by flooding of link state update messages. For this every node has to do the flooding after regular interval. This overhead is removed in ACO where only pheromone tables are updated parallel.

3) Adaptively and Stagnation

the network gets stagnant due to the flooded packets sent by all the nodes. In ACO only ant packets are sent to other nodes which are very small in size.

How ACO will do the load-balancing?

The basic idea behind the load balancing is to transfer the load of the heavy nodes and shift it the light nodes. In the DHT based system as explained in the section II above; the nodes are searched by different techniques. In ACO the searching of the light nodes will be done by ants. As the ants forage for food; to do the load balancing they will search the light nodes. While travelling the ant will reach to node they will check its load status and will simultaneously update the pheromone table which contains the load value of a node. When the ant will find the sufficient light node it will generate a backward ant and destroy itself. The backward ant will follow the path depending upon the pheromone concentration on it and will reach to the source node (the heavy node) and will destroy itself. The source node now has the information of the light node and the shortest path to reach to the node; and will transfer part of the load to the light node. In this way the load balancing is achieved. This has advantages over the DHT based systems

1. The ACO is free from the single point failure due to which the centralized DHT systems fails.
2. ACO has the advantage over the decentralized systems also; that the decentralized systems suffer from the load information collection overhead. While applying ACO there is no need of load information collection.

IV. FUTURE SCOPE

The ACO still is a part of the research. There are many areas where ant based solutions could work. The ACO also has some shortcomings like if a path is selected by all the ants then the path becomes stagnant, and the performance reduces gradually. So there was need to refine the ACO. The Multiple Ant Colony Optimization (MACO) is used to overcome the shortcomings of the ACO. The MACO is same like ACO with little advancement. In MACO there are more one colonies of ACO are working simultaneously. Each of the ACO will find its own optimal path and at the source node the comparison between the set of optimal paths will take place. Among the optimal paths the one which will give the shortest path will be selected. If the selected path becomes stagnant a\then the other path from the pool of optimal paths is selected and the performance will not be affected.

REFERENCES

[1] Ananth R., Karthik L., et.al. “Load Balancing in Structured P2P Systems,” Proc. Second Int’l Workshop Peer-to-Peer Systems (IPTPS ’03), Feb. 2003

[2] Brighten G., Karthik L., et.al., “Load Balancing in Dynamic Structured P2P Systems,” Proc. IEEE INFOCOM, 2004.

[3] K. M. Sim, W. H. Sun., “Ant Colony Optimization for Routing and Load-Balancing: Survey and New Directions,” IEEE Trans. On Systems, Man, and Cybernetics—Part A: Systems and Humans, Vol. 33, no. 5, Sept. 2003.

[4] Chyouhwa C. and Kun-Cheng T. “The server reassignment problem for load balancing in structured P2P systems,” IEEE Trans. Parallel and Distributed Systems, Vol. 19, Issue no. 2, Feb. 2008

[5] Ion S., Robert M., et.al, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” Proc. ACM

SIGCOMM '01, pp. 149-160, 2001.

[6] Jonathan L. and Margo S., "Distributed, Secure Load Balancing with Skew, Heterogeneity, and Churn," Proc. IEEE INFOCOM, 2005.

[7] P.Brighten G., and Ion S., "Heterogeneity and Load Balance in Distributed Hash Tables," Proc. IEEE INFOCOM, 2005.

[8] Ziyong X. and Laxmi B., "Effective load balancing in P2P systems," Proc. Sixth IEEE symposium Cluster computing and grid, Feb. 2006