Topology Organization in Peer-to-Peer Platform for Genetic Algorithm Environment

Hong Yao and Linchen Yu

School of Computer Science and Technology
China University of Geosciences
Wuhan, 430074, China
yaohong@cug.edu.cn

Abstract. Genetic algorithms (GAs) have the inherent nature of parallel search. With the advantage of the computing power of PCs, GA computing environment can be shifted from a single machine to Internet. Topology, the organization of the peers, as well as their dynamic change and maintaining mechanisms, is important to organize an efficient and stable topological structure. A new topology is proposed in this paper to create a hybrid structure for large scale of peers. The whole structure is divided into two layers. The upper part is composed by super nodes, while the lower part is composed by the ordinary nodes. Testing shows that it is good for maintaining the platform stable and scalable.

Keywords: peer-to-peer network, topology, super node, ordinary node.

1 Introduction

Genetic algorithms (GAs) have been demonstrated to be an effective problem-solving tool. The evolutionary process has the inherent nature of parallel search. There is an approach of reducing computational workload is to develop a GA computing platform, which is running in many PCs over network. Paper [1] brings forward a layered architecture for this job by K.C. Tan and M.L. Wang, etc. However, how to organize the topology is not clearly defined.

For the advantages of non-centralization, scalability, robustness, load balancing, and etc, P2P [2] technology has become more and more popular. With the advancement of hardware and network, end nodes which are considered at the edge of Internet have better performance and bigger bandwidth. As end nodes wish and have ability to share resources, the traditional *Client/Server* (C/S) model is challenged by the *Peer-to-Peer* (P2P) model. End nodes are organized in application layer of Internet, which forms the overlay network.

1.1 Related Works

We can divide P2P overlay network into four modules according to their topology, which are centralized topology, decentralized unstructured topology, decentralized structured topology, and partially decentralized topology.

Z. Cai et al. (Eds.): ISICA 2009, CCIS 51, pp. 156–161, 2009.

The biggest advantage of centralized topology is simple maintaining and efficient discovery. Because resources discovery depends on centralized catalog system, the algorithm is effective and supporting complex inquiry. The most major problem is similar to the traditional client/server structure, which is easy to result in the server breakdown, situation of visiting hotspot.

The distributed unstructured model distributes the resources in each peer. There is no centralized server, and every peer in the network is equally. Its disadvantage is that in order to search some resources, the request packets may pass through the entire network or at least a very great scope.

Distributed Hash Table, called DHT [3] for short, is actually an enormous hash table maintained by a large number of peers in a wide area. Peers will dynamic join and leave DHT structure adaptively, and it has the advantages of scalability, robustness, uniform distribution of node ID, and self-organizing capacity. The biggest problem is that maintenance of DHT is complex, especially when peers join and leave frequently, it will greatly increase the cost of maintenance. Another problem is that DHT only supports precise search, but not complex enquiries based on content or semantic.

Hybrid topology [4], is also called partially decentralized topology, is a balance of centralized structure and distributed structure, utilizing both of their merit. The advantages of this model is that compared to distributed model, it will reduce inquiry packets from dissemination, and compared to centralized model, it will reduce the influence of failure in the centre node. The main disadvantage is that it is difficult to partition super peer and ordinary peer [5]. Besides, the two kinds of peers are not easy to manage.

1.2 Contribution of This Work

Even though the hybrid model is popular currently, it also faces the following drawback: 1. How to judge peers' service ability and partition SN and ON rationally; 2. How to manage peers on upper layer, locating near neighbors fast, and avoiding isolated peers; 3. How to reduce churning of peers on lower layer when peers leaving on upper layer; 4. How to reduce searching packets from flooding in the network, improving search efficiency.

This paper introduces a novel hybrid topology, which is as follows: 1. Ring based topology of super peers, providing fast search of neighbors and avoiding isolated peers; 2. Ordinary peers will maintain backup father nodes, reducing churning when some father nodes leave; 3. Based on peers' value of relative ability, partitioning SN and ON according to the size of peers' number, balancing the system's load as far as possible.

2 Design of Overlay

2.1 System Overview

The overall topology adopts the popular two-tier structure, as shown in Figure 1. The upper layer is composed by those peers with strong power, called super node (SN for short), while the lower layer is composed by the others, called ordinary node (ON for short). The hierarchical structure not only reduces the size of distributed peers on upper tier, but also disperses pressure of the central peers. It integrates the advantages of central and distributed structure.

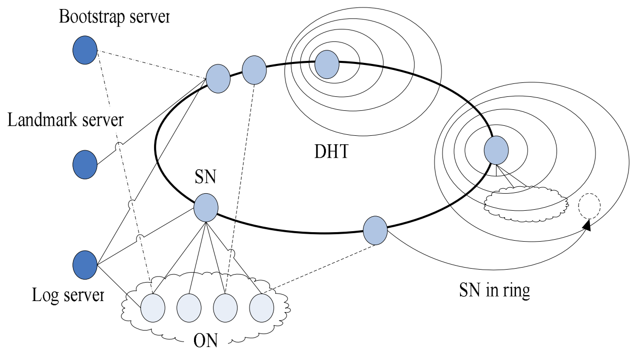

Fig. 1. Overlay Topology

Neighbor peers of super node are organized as a serial of rings based on delay. On the one hand, it is used for searching low delay neighbors, and on the other hand it avoids isolated node group from keeping sampling neighbor peers of delay in every interval.

Besides, there is also a distributed hash table (called DHT for short) for managing all super nodes. It is used for searching precise resources. It can find the specified resource within Log(n) hops in a DHT network of n nodes. This mechanism is a supplement of the searching strategy. It not only avoids the searching packets from flooding over the entire network, but also ensures scarce resources to be found with high probability.

The management of ONs is like centralized approach. Every ON should connect to a SN, and only keeps the relation with its father nodes. This structure will take low overhead of ONs, because they need neither maintain neighbor peers, nor manage resources, and most of this work is done by their father SNs.

2.2 Super Node Structure

Each SN will maintain a certain number of neighbor nodes, and each node is placed in a specified ring according to the delay, as is shown in Figure 2.

Peer A will randomly select some neighbors such as peer B from each ring periodically. Then A will send B a packet containing some other peers from its rings.

When peer B receives the packet from peer A, it will measure the delay with A and all the peers in the packet, and then add new peers in its rings, or update the old peers' delay.

This ring structure is different from former methods that only maintaining the near neighbor. It improves the coverage of each node. In the search process, it reduces the frequency of communication between peers effectively, reducing system load too. On neighbor's discovery mechanism, we use Gossip protocol [6].

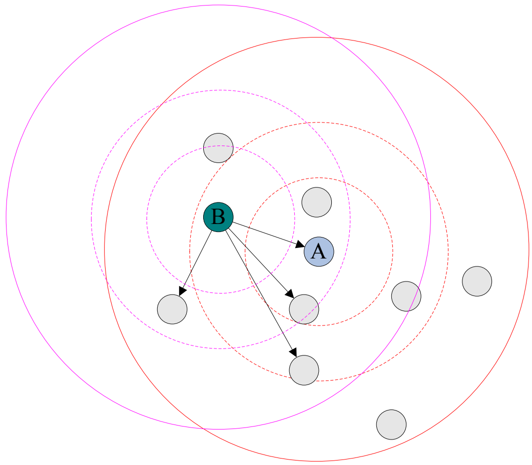

Fig. 2. Maintenance of ring structure

2.3 Ordinary Node Organization

ONs are on the lower layer of the system, because of their weak service capability, they only maintain their own resources, and don't transmit packets for other peers. This will not only reduce the radius of the entire network, increase search coverage, but also balance performance of all peers, improve efficiency of the whole system.

Every ON must connect to a SN, taking the SN as its father node, and the SN will process most of its request, including searching for neighboring nodes, publishing resources, searching resources, and so on. Therefore, the upper SN nodes are important to the lower ONs. If father node leaves, and the ON can't quickly find another father node to replace the former one, it will make the ON instability. So how to manage the ONs effectively is very important for the users on this layer.

3 Performance Evaluations

Test environment is shown in Table 1. The network environment is LAN. There is one Bootstrap server, one Landmark server, and 120 client peers.

Table 1. Test Environment

Node	OS	CPU	Memory	Bandwidth
Bootstrap server	Red Hat9.0	P4 2.4G Hz	1GB	100Mbps
Landmark server	Windows XP	Celeron(R) 1.70G Hz	256MB	100Mbps
Client peer	Windows XP	Athlon(tm) XP 2500+	512MB	100Mbps
…	…	…	…	…

3.1 Neighbor Distribution

Figure 3 shows SNs' number of neighbors in their rings and ONs' number of father nodes.

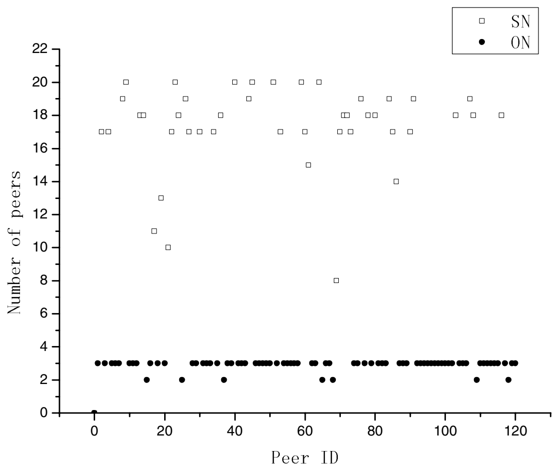

Fig. 3. Distribution of neighbor/father nodes

Because we limit the maximum number of SN's neighbors is 20, we can see from the figure that they are almost between 17 and 20, indicating the efficient and stable structure of neighbor peers in the system.

In addition, as can be seen from the figure, almost all the number of ONs' father nodes is 3, because we assign a primary father node and two backup father nodes to an ON. When an ON logins successfully, it will fetch information of other SNs from its primary father node. So the peers on the lower layer will not be jittering by the departure of peers on the upper layer.

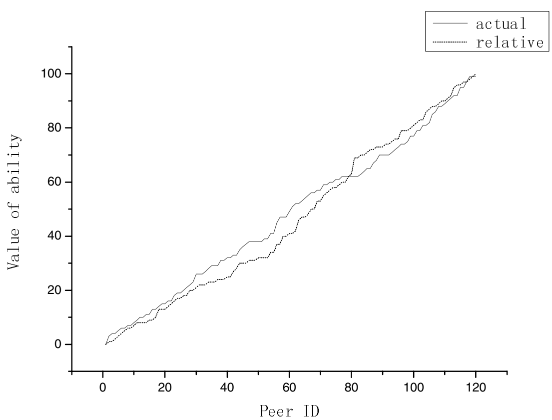

Fig. 4. Distribution of value of actual/relative ability

3.2 SN/ON Partition

The partition of SN and ON is according to their service ability. Our goal is that those peers with high value of ability act as SN, and those with low value of ability act as ON. Besides, the proportion of SN and ON should be controllable, avoiding imbalance for peers dynamic joining and leaving.

Figure 4 shows the distribution of peers' value of actual ability and value of relative ability namely random number.

4 Conclusion

This paper proposes a new P2P topology based on current existing mode for GA distributed computing environment. The whole structure is divided into two layers, and the upper part is composed by those peers with strong power, called super nodes, while the lower part is composed by the others, called ordinary nodes. Testing shows that it is good for maintaining the platform stable and scalable, as well as processing joining and leaving of the peers. Besides, it improves the system's load balancing.

However, there are still some thorough problems to be solved. For example, how to figure out the upper limit of neighbors on SN is a question that should be solved according to actual network environment. Moreover, the choice of Landmark servers in large-scale environment is also difficult to determine.

In the near future, we will shift some implementation of GA, such as TSP problem, to this overlay network.

References

[1] Tan, K.C., Wang, M.L., Peng, W.: A P2P Genetic Algorithm Environment for the Internet. Communications of the ACM 48(4), 113–116 (2005)

[2] Kant, K., Iyer, R., Tewari, V.: A Framework for classifying peer-to-peer technologies. In: Proceedings of the 2nd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID 2002), pp. 368–379. ACM Press, USA (2002)

[3] Jain, S., Mahajan, R., Wetherall, D.: A study of the performance potential of DHT-based overlays. In: Proceedings of the 4^{th} conference on USENIX Symposium on Internet Technologies and Systems, Seattle, WA. USENIX Association, USA (March 2003)

[4] Xiao, L., Zhuang, Z., Liu, Y.: Dynamic Layer Management in Superpeer Architectures. IEEE Transactions on Parallel and Distributed Systems 16(11), 1078–1091 (2005)

[5] Garbacki, P., Epema, D.H.J., van Steen, M.: Optimizing Peer Relationships in a Super-Peer Network. In: Proceedings of the 27^{th} International Conference on Distributed Computing Systems, pp. 31–42. IEEE Computer Society, USA (2007)

[6] Baset, S.A., Schuizrinne, H.G.: An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In: Proceedings of 25^{th} IEEE International Conference on Computer Communications, Barcelona, Spain, pp. 1–11. IEEE Computer Society, Los Alamitos (2006)