An Efficient Distributed Search Method*

Haitao Chen, Zhenghu Gong, and Zunguo Huang

School of Computer Science, National University of Defense Technology,
Changsha, Hunan, China
nchrist@163.com

Abstract. The big challenge of constructing P2P applications is how to implement efficient distributed file searching in complex environment which implies huge-amount users and uncontrollable nodes. FriendSearch is introduced to improve the efficiency and scalability of distributed file searching. FriendSearch introduces a new hybrid architecture in which the storage and search of raw file is based on DHT network, but the storage and search of meta-data is based on unstructured P2P network. FriendSearch learns interest similarity between participating nodes and uses it to construct friend relations. The forwarding of queries is limited to friend nodes with similar interests. Simulation tests show that FriendSearch algorithm is both efficient and scalable.

1 Introduction

With the development of Internet applications in scope and depth, the role of ordinary nodes changes from just receiving content passively from servers to acting as a supplier of Internet content. P2P technology provides a new application pattern which enables the edge nodes participate the Internet application as both clients and servers at the same time. It offers powerful support for the construction of huge and complicated distributed network application. P2P file sharing applications which allow ordinary user to share files in local disk to others, have become one of the most popular Internet applications.

The big challenge of constructing P2P application is how to implement efficient distributed file searching in complex environment which implies decentralized and huge-amount users, uncontrollable nodes with unbalanced computing capacity and network connection. This paper presents a new distributed file search methods – FriendSearch. FriendSearch constructs friend relations between nodes based on interest similarity and limits query broadcast to nodes with similar interests.

2 Related Work

The current mainstream P2P file sharing applications can be divided into centralized search model, broadcast search model, and hierarchical search model. Napster[1] is a typical centralized search model. It can search effectively and reliably. But the

* This research is supported by the National Grand Fundamental Research 973 Program of China under Grant No.2003CB314802, also the National High-Tech Research and Development Plan of China under Grant No.2003AA142080.

L.T. Yang et al. (Eds.): HPCC 2005, LNCS 3726, pp. 265 – 270, 2005.

directory server is a single point of failure and performance bottleneck. Gnutella[2] is a typical broadcast search model. The main disadvantage of broadcast model is high bandwidth consumption which leads to bad scalability. Furthermore, the search results of this model is uncertain which means the documents existing somewhere in the network maybe can not be located. Kazaa[3] and JXTASearch[4] are typical hierarchical search models. The model enhances the stability and scalability through super-nodes. But the super-nodes are new performance bottle-neck. Also the communication between super-nodes depends on broadcast routing, which restricts scalability.

The problems of existing distributed file search systems include high bandwidth consumption, poor search pattern, bad ranking of results and so on. According to architecture, these projects can be partitioned into search in structured P2P network and search in unstructured P2P network.

Many researches concentrate on search in unstructured P2P network, such as Freenet[5], NeuroGrid[6], APPN[7], GS[8], Alpine[9] and so on. Freenet[5] can guarantee the anonymity of publisher, reader, and storage space supplier. The requests will be routed to the most possible positions. Its performance is almost as good as DHT, but it still has some uncertainty. NeuroGrid[6] abstracts the knowledge of documents distribution from the search results and makes routing decision based on keywords distribution of other nodes. The disadvantages of NeuroGrid are: the information updating speed is slow; the result of query is uncertain; the size of route table is interrelated with the number of files and the number of nodes in the network. APPN[7] optimizes search process through the construction of associated rule. One kind of simple associated rule is possession rule which means owning some special files. The spending of establish rule and the choice strategy of rules are still problems which APPN faces. GS[8] establishes shortcuts among nodes based on principle of interest locality. The method is simple and effective, but it lacks inspection for large scale network. Alpine [9] manages the sharing information by groups, and each member of a group will evaluate other members' trust degree according to the satisfaction with their services. For this method the establishing of group depends on the user contact out of the P2P network, also the scalability of group is limited.

Structured P2P network can finish search process in several limited hops, which provide a good method for deterministic search. Some researches devote to realize fuzzy searching in DHT network. PSearch[10] makes use of LSI (Latent Semantic Indexing) to construct semantic space of files and maps the file vector space to CAN space. PSearch only supports keywords query and is a promising method for text search in distributed environment. Semplesh[11] presents a method of mapping the RDF Triples to DHT in which every item of RDF Triples will be mapped to DHT network once. There are problems for Semplesh such as low search efficiency, heavy workload for popular item and lacking support for substring matching.

3 Search Based on Friend Relations

3.1 Hybrid Architecture

This paper presents hybrid architecture- HA. HA includes file search layer and metadata layer, in which storage and search of raw documents are based on structure P2P while the storage and search of metadata are based on unstructured P2P network. This architecture combines the advantages of deterministic search in structured P2P network and the advantages of fuzzy search in the unstructured P2P network. Unstructured P2P network can express easily the complex relations among the file meta-data and meet the users' diverse query needs. DHT network can effective solve problems such as file moving, file replicating and download. It also supports for the discovery of file relations.

Each node of HA participates in two kinds of networks at the same time. The construction of unstructured network relies on the DHT network. The nodes publish raw files in DHT network. Then these raw files will establish relations according to their metadata. At last the nodes can establish friend relations according the relating of raw files. The relations of files and friend relations between nodes belong to the metadata layer. Based on the friend relations between nodes, most search requests can be restricted in a very limited scope.

3.2 Algorithm of Constructing Friend Relations

One key problem of FriendSearch is how to construct overlay network of friend relations based on file possession relations. The basic process of constructing friend relations includes three steps. 1) First each node collects all nodes that share same files as friend candidates in the bottom DHT network. 2) Then it ranks these friend candidates from high to low according the number of sharing same files. 3) At last it chooses the first k nodes as its friend nodes. If we view friend relations as directed edge, the graph that is make of nodes and friend relations is a directed graph with high clustering coefficient. The total overhead of this algorithm is linear with number of nodes. But the computation is distributed. Computing overhead of each node is only linear with its sharing files, which is obviously an acceptable overhead. At regular intervals, the algorithm has to run. Also the algorithm can run only when the search success rate is low enough. Simulation tests show that friend relations between nodes are very stable, so the algorithm needs not to run with high frequency.

3.3 Search Algorithm Based on Friend Relations

Search algorithm based on friend relations includes two steps. 1) First it makes use of two level friend relations to search. Original query nodes will broadcast query to all its direct friend nodes. If direct friend nodes fail, the query will be forwarded to the friend nodes of its direct friend nodes. Most requests can receive responses in first step. 2) Then if the friend search fails, it adopts efficient DHT-based flooding search as supplement. The search process uses cache to improve performance.

4 Simulation Test

For fully test of FriendSearch algorithm, we adopt many groups of test data and many test measures. We use simulation data and web log data to test the performance of pure multi-hops friend relations in section 4.1 and section 4.2.

We evaluate the performance of search algorithm using the follow targets. Success rate denotes the proportion of success rate of search. Search Consumption is the

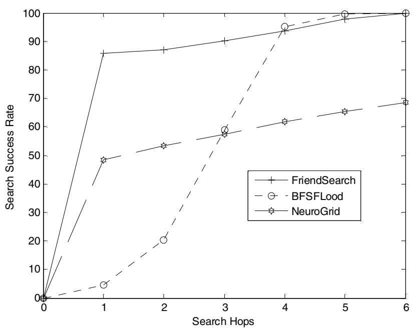

Fig. 1. Hops VS Search Success Rate

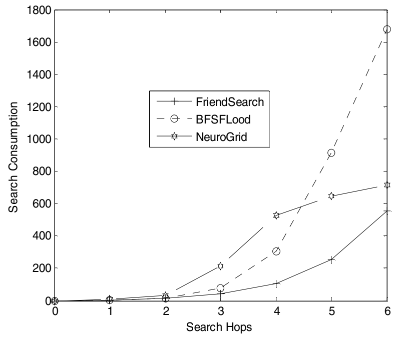

Fig. 2. Hops VS Search Consumption

number of peers in the system involved in query processing for each query. A smaller query scope increases system scalability. Query hop stands for the average delay for reply to come back.

4.1 Simulation Test Based on Simulating Data

NeuroGrid simulator [12] is a generic P2P simulator developed by Tokyo University of Japan. It has many configurable parameters and can implement simulation of P2P file sharing system with good expansibility. We implement FriendSearch search algorithm on the base of NeuroGrid simulator.

NeuroGrid simulator has several configuration parameters and can generate different test data. We found that test results of FriendSearch were stable. So here we just use typical results. Simulation tests compare FriendSearch with BFSFLood[13] and NeuroGrid[5]. BFSFlood is an improved version of flooding algorithm, which randomly chooses fixed number of neighbor nodes to forward query. Figure 1 shows that FriendSearch gains more than 80 percent search success rate at first hop, the search success rate enhance slowly with the searching hop increased. Figure 2 shows search consumption augment rapidly with the search hop increased, and FriendSearch keep the lowest rising speed.

We can draw these conclusions from the test results:

- The size of route table of FriendSearch algorithm is fixed and small. NeuroGrid algorithm can get good performance at the cost of more than one hundred of items in its route table, so the cost of maintenance is high.
- Search success rate of FriendSearch algorithm is very high. Especially it can get more than 80 percent success rate just at the first hop.

4.2 Simulation Test Based on Web Log

Web log data and P2P access data are similar at some ways and they both obey some same laws. Collecting the web log data is much easier than P2P data, so many researches [7] [8] adopt web log data as test data of P2P search research.

We adopt three groups of wildly-used web log data to test FriendSearch algorithm.

- Boston [14] is the web log data of Boston University which contains 558261 records, 538 nodes and 9431 files.
- Berkeley [14] is the web log data of Berkeley University which contains 1703836 records, 5222 nodes and 116642 files.
- Boeing [14] is the web log data of Boeing Corporation which contains 4421526 records, 28895 nodes and 254240 files.

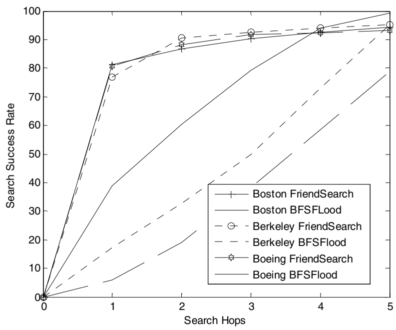

Fig. 3. Search Success Rate

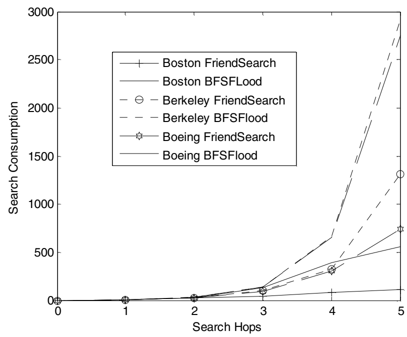

Fig. 4. Search Consumption

These three groups of web log data respectively represent access case of different scale networks. The test method is similarly with the method in [7]. The test results are shown in figure 3 and figure 4. We can draw these conclusions from the test results:

- Three groups of test results of FriendSearch algorithm are similar, which shows FriendSearch algorithm's stability.
- Simple friend relations construction can guarantee very high search success rate. Search success rates of FriendSearch algorithm in three groups of test data are very high, especially the searching success rate can exceeds 75 percent at the first hop.
- Search consumption of FriendSearch is low and the search efficiency of FriendSearch algorithm is high.

5 Conclusions

This paper researches on distributed file search in complex environment and presents a new method of search sharing files. FriendSearch constructs friend relations between nodes based on search interests and sharing files. The search requests firstly are forwarded to friend nodes. Only failed requests will continue to broadcast in DHT flooding pattern.

Simulation tests show that FriendSearch algorithm is efficient and stable. FriendSearch brings the performance to within an order of magnitude of improvement compared with classical algorithms such as BFSFlood[13], NeuroGrid[6] and so on. The future researches include: 1) more effective algorithm of constructing friend relations. For example it can take file rarity into count. 2) Adding semantic description on friend relations to improve the expansibility of system.

References

1. Napster. www.napster.com. 2005.
2. Gnutella. www.gnutella.com. 2005.
3. Kazaa. www.kazaa.com. 2005.
4. JxtaSearch. http://search.jxta.org/. 2005.
5. Clarke, I., Sandberg, O., Wiley, B. and Hong T. W. Freenet: A Distributed Anonymous Information Storage and Retrieval System. In Proc of the Workshop on Design Issues in Anonymity and Unobservability, Ed. Federrath H., Berkeley, CA, July 2000.
6. Joseph, S.R.H. NeuroGrid: Semantically Routing Queries in Peer-to-Peer Networks. International Workshop on Peer-to-Peer Computing, Pisa (2002).
7. Edith Cohen, Amos Fiat, Haim Kaplan. Associative Search in Peer to Peer Networks: Harnessing Latent Semantics. In Proc of INFOCOM 2003.
8. Kunwadee Sripanidkulchai, Bruce Maggs, Hui Zhang. Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems, In Proc of INFOCOM 2003.
9. Alpine. http://www.cubicmetercrystal.com/alpine/. 2005.
10. Chunqiang Tang, Zhichen Xu, Sandhya Dwarkada. Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Over-lay Networks. In Proc of SIGCOM 2003.
11. Semplesh. http://www.plesh.net/. 2005.
12. Sam Joseph, An Extendible Open Source P2P Simulator. P2P Journal. 2003.
13. V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A Local Search Mechanism for Peer-to-Peer Networks. In Proc of CIKM, 2002.
14. webtraces. http://www.web-caching.com /traces-logs.html.