ISSN:0975-9646

Search in Peer-to-Peer Networks based on Dynamic Topology Adaptation

Maruthi Ayyappan, Chirantan Sharma

Department of Electronics and Computer Engineering
Sreenidhi Institute of Science and Technology
Ghatkesar
Hyderabad, India

Abstract— **Peer-to-Peer (P2P) is one of the new computer networking paradigms, and currently finds applications in many diverse fields such as large-scale distributed computations and live media streaming. This paradigm is built around the concept of leveraging the resources of the participants to perform its functions in a manner which is robust and scales reasonably. One of the major difficulties, however, with P2P environments is the need for an efficient searching algorithm. Currently, there are algorithms such as Flooding and Random Walk (RW) algorithms, each having pros and cons of their own. While the Flooding algorithm searches by means of 'flooding' or broadcasting a search everywhere and covers most nodes in the system, it generates a large amount of search messages which increase the cost of the algorithm. On the other hand, the RW algorithm searches much more conservatively and incurs only a fixed cost for each hop, but takes much longer to finish the search. This paper proposes the Dynamic Search Algorithm (DS), a generalization of flooding and RW. It resembles flooding for short search and RW for long-run search.**

Keywords: **MOU Replacement Policy, P2P Networks, Dynamic Topology Adaptation.**

I. INTRODUCTION

The proposed system employs techniques to fashion the searching process in an unstructured file sharing system such that similar interests of certain peer nodes are taken into account, thereby increasing the efficacy and scalability of the system. P2P networking is basically a distributed architecture which divides or shares the workload between peers. Peers contribute a fraction of their resources such as network bandwidth, storage capacity, processing power etc. and make it directly available to other participants or peers. This is in contrast to the client-server model, where only the server provides all the resources and the clients consume them.

The 3 most common types of P2P environments are discussed below:

A. Napster type

The Napster type of P2P environment is 'hybrid' - a mix of the client-server and the P2P models. Each peer first connects to a server in the network and sends it information of the shared files it has. The server receives this information and maintains a 'shared-file list' for each peer. If suppose another peer sends a request to the server for a certain file, the server searches its database for the lists to determine which peers can provide the file. Consequently, the server generates a reply message containing the details of the peer such as host name, ID etc. and forwards it to the requesting peer, so that it may initiate a connection directly with the peer and obtain the file from it directly. Figure 1 shows the network structure of the Napster environment.

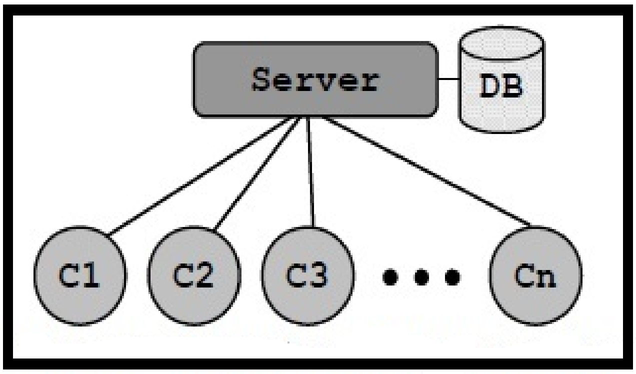

Fig. 1: The network structure of Napster environment

The main advantage of the Napster system is that since it stores only the information of the peers and not the files itself, it reduces a lot of burden on the system. However, each peer must send index information to the server at connection time and also each peer must be connected to the server each time it wants a file. Hence, a lot of traffic is still centred on the server. Also, all the peers cannot use the service when the server fails to work, thus pointing to a reliability problem with this approach.

B. Soribada type

The Soribada type, developed originally by a company in Korea as a P2P environment for sharing of MP3 audio files, is also a hybrid type. In contrast to the Napster system, however, the peers do not send index information about its shared files to the server at connection time. In fact, it does not maintain any database about shared files. Instead, where a peer requests a file, the server analyzes its request and sends request messages to all the peers in the network. The peer, upon receiving this message, checks whether it has the specified file in its directories and sends the appropriate information to the server. The server then sends the summarized information of each peer to the original peer, which may proceed to send a file request to the appropriate peer based on this information. Again, the drawback of this system is also that it is too server-centric, as the peers must connect to the server each time for file service. It also has the same reliability problem as with the Napster method, that all peers are denied services should the server fail.

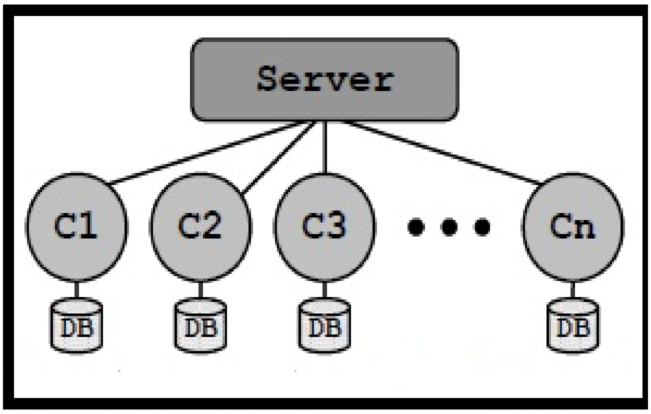

Fig. 2: The network structure of Soribada environment

C. Gnutella type

The Gnutella type is a pure P2P environment. The peers in this system satisfy the roles of both server and client. In this system, when a peer executes some process, it must get ID's of the peers already existing in the Gnutella network, and connect to them. Thus, the peers establish a mesh type network and share resources among themselves based on specific requests. The main drawback of this system is that since the broadcast method is used for peer-to-peer communication, file access speeds are greatly reduced, while the network traffic is heavy. Another problem is that a peer may get disconnected from the entire network if all its adjacent peers develop a failure. Figure 3 shows the network structure of Gnutella system.

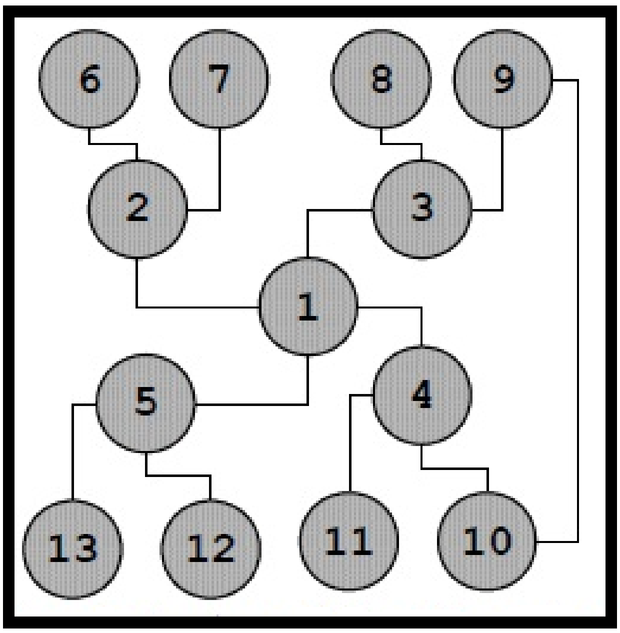

Fig. 3: The network structure of Gnutella environment

II. EXISTING METHOD

As discussed earlier, Flooding and Random Walk (RW) techniques are the two popular search protocols in P2P environments. In Flooding, the search request is broadcast to every peer in the network. So, in this process a large number of messages are generated. However, no management techniques need to be implemented for a broadcast mechanism, thus saving some cost. Also, broadcast method tends to find the destination peer much quicker than other techniques. However, since the overhead increases exponentially with each new message (because each of them is broadcast), network congestion surfaces as a major problem with this technique, which is otherwise well-suited for dynamic topologies.

The RW technique on the other hand, is a weakly-decentralized technique suitable for clustered overlay topologies. It searches conservatively and may be able to re-issue each search request multiple times. It generates only a fixed number of messages per hop, and thus incurs less overhead to the system. However, this causes the search times to be much longer than the flooding algorithm. The RW algorithm may also be used to add new peers to the system at fixed overhead.

The present techniques have various disadvantages, ranging from production of false negatives to high managements costs. There may also be a 'Hot-Spot' problem if many peers request the same file, which increases load. The algorithms are less precise and efficient as only exactly matched queries are supported.

III. PROPOSED METHOD

A. Searching P2P Networks Efficiently

The proposed system uses dynamic topology adaptation which maintains the semantic communities to improve search efficiency. The semantic communities may lead to spot-on problem when many users are requesting for the same file due to which the load increases. So the system uses a load balancing mechanism that directs the responsibility to answer a query to less loaded peer nodes. The system proposes a dynamic TTL (time-to-live) update scheme to further limit the network congestion without decreasing the query success rate.

- Provides precise results.
- Use of unstructured P2P reduces the network set-up cost.
- Never produce false negatives.
- Load of the database is well managed.
- Highly efficient.
- Semantic community creation yields successful results.

The main objective of this proposed system is to develop techniques to render the search process in unstructured network file sharing systems more efficient by taking advantage of similar interests of peer nodes (semantic communities) and implementing a direct flooding search protocol. Figure 4 shows the mechanisms proposed in this system.

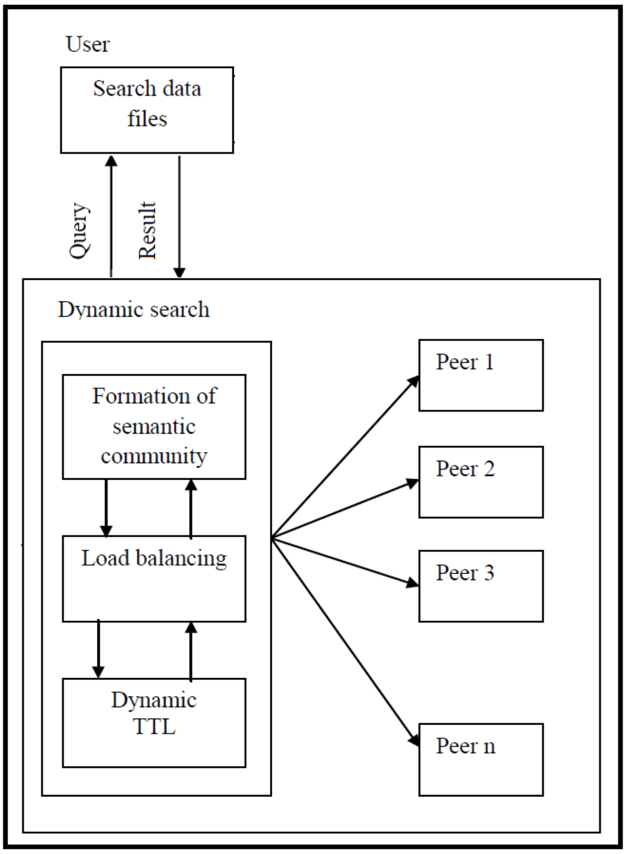

Figure 4: Mechanism in P2P Search Algorithm

B. *Dynamic Topology Adaptation*

Dynamic Topology Adaptation is enforced by directional Acquaintance links towards the peers. The basic principle consists of dynamically adapting the topology of P2P network so that peers sharing similar interests form well-defined semantic communities. Users are interested only in specific types of content; therefore forming semantic communities will increase the search efficiency and the success rate. Replacing a peer from the list depends on the responses to previous requests issued by each peer. In this paper, a replacement policy is evaluated.

The Most Often Used Policy (MOU) maintains ranking of the peers and select the peer with highest ranking as the acquaintance. A peer encompasses a highest ranking if it answers to several queries if it is well-connected.

After each successful query, every peer on the trail followed by the query has its rank increased by a decreasing value. More preference is given to newer answers by adding an age factor which decrements the ranking of older answers periodically. This gives preference to peers that stay longer in the system and are stable.

C. *Dynamic Search*

It uses TTL flooding scheme and exhibits improved efficiency. The semantic communities improve response time by increasing the chances that matching files are found inside the community within short distance of the requester. Therefore smaller TTL values are used for queries and thus reduce the network congestion without decreasing the success rate. The peers also maintain index of the files stored on their neighbouring peers. Thus the peer can explore other peers with similar interests at no communication cost. This also increases the success rate and reduces network congestion. The semantic communities has several disadvantages such as the network may get subdivided into sub networks and also peer searching for contents that are less frequently searched may give very low success rate.

Load Balancing

Flooding algorithms naturally direct much of the traffic towards highly connected peers. In this proposed system, a peer that has many neighbours can quickly become a hot-spot, because it receives more queries and perhaps sends more files to requesting peers. Although the issue of file transfers is not explicitly addressed in this project, it is a large source of overhead in P2P file-sharing networks and should not be overlooked. Therefore the following mechanism is used to better balance the file traffic. Before answering a query, a peer must check whether any neighbour also holds the same file. If so, it delegates the responsibility for answering the query to the peer among those serving the file that has the smallest in-degree (note that this peer could be “p”). Otherwise, “p” sends the file itself. The explanation behind this approach is that well-acquainted peers are likely to be heavily loaded i.e., receive a lot of requests and serve more files than peers with fewer neighbours. Further, there is a decent chance that the neighbours of a peer also have the same files. Therefore, force the less loaded peer to assume part of the load.

D. *Dynamic TTL*

Semantic communities reduce the query traffic. The query entering the semantic community to which the requested file belongs is answered by that peer with fewer hops. If the query does not match that community then it traverse more peers. Therefore the proposed system reduces the TTL value twice instead of 1 when the received query falls within the semantic community. This also reduces the network congestion without decreasing the success rate.

IV. Conclusions

In the system, Search in Unstructured Peer-to-Peer Networks based on dynamic topology adaptation is proposed. The dynamic topology adaptation is employed to boost search efficiency and potency. The main advantages of this proposed system is that it develops a technique that renders the search process in unstructured network making it more efficient and robust. The MOU replacement technique takes advantage of the peer nodes and implements a “directed flooding” search protocol. The system performs well in the unstructured peers to provide the specified results.

V. References

[1] D.Xu, S.Hambrush, B. Bhargava, On Peer-to-Peer Media Streaming, In Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS’02), Vienna, Austria, July 2012.

[2] Ahmed R, Boutaba R. Distributed pattern matching: a key to flexible and efficient P2P search. IEEE Journal on Selected Areas in Communications, 2007, 25(1): 73−83.

[3] Zhang R M, Hu C. Assisted peer-to-peer search with partial indexing. IEEE Transactions on Parallel and Distributed Systems, 2007, 18(8): 1146−1158.

[4] M. Kleis, E. K. Lua, and X. Zhou, “A Case for Lightweight Super peer Topologies.” KiVS Kurzbeiträge und Wksp Mar, 2005. pp. 185–88.

[5] Wang, Y. and H. Kobayashi, 2006. High performance pattern matching algorithm for network security. IJCSNS, 6: 83- 87.

[6] Zhao and V. Lo, "Result Verification and Trust-based Scheduling in Open Peer-to-Peer Cycle Sharing Systems," The Fifth IEEE International Conference on Peer-to-Peer Computing (P2P 2005), pp.31-38, Sept. 2005.

[7] P.Suresh Kumar, P.Sateesh Kumar, S.Ramachandram, “Recent Trust Models in Grid,” Journal of Theoretical and Applied Information Technology, vol.26 No.1, pp.64-68, April 15, 2011.

[8] Baruch, A., Christian, S.: Robust Random Number Generation for Peer-to-Peer Systems. Theory of Computer Science. 410(6-7), 453–466 (2009).

[9] Zhou, L., Zhang, L., McSherry, F., Immorlica, N., Costa, M., Chien, S.: A first look at peer to- peer worms: Threats and defenses. Castro, M., van Renesse, R. (eds.) IPTPS 2005. LNCS, vol. 3640, pp. 24–35. Springer, Heidelberg (2005).

[10] M. Castro, P. Druschel, A.-M. Kermarrec, A. Nandi, A. Rowstron, and A. Singh. Splitstream: high-bandwidth multicast in cooperative environments. In Proceedings of the nineteenth ACM symposium on Operating systems principles, pages 298.313. ACM Press, 2003