ISSN 2229-5518

Improving Query Search Performance in Unstructured P2P Network Using Proclamation Based Search (PS) Algorithm

Ar.Arunachalam, Dr.V.Khanaa

Abstract— The challenge in unstructured p2p networks is designing an efficient Search Algorithm. Some typical search algorithms are random walk and flooding. Flooding in general covers many nodes but the drawback is it generates huge amount of query messages. Random walk generates only fewer amounts of query messages but takes long time to search. In this paper we propose a Proclamation Search (PS) algorithm for avoiding multiple hops in forwarding the query in unstructured Peer to peer network. Proclamation is publishing the synopsis of the contents a peer tends to share and properly distributed and cached by other peers. In this system, the nodes anticipatorily publish their contents, and selectively store interesting contents received from other peers. When there is a search request, a node can find the destination nodes by looking up its local publishing repository, and thus obtain one-hop search with moderate search cost. We analyze the performance of PS algorithm compared to other search algorithms based on search efficiency.

Index Terms— Peer-to-peer, proclamation, PS, search, synopsis, Un-structured P2P

1. INTRODUCTION

Peer-to-peer (P2P) networks have emerged as a new Internet computing paradigm over the past few years. The most prevalent P2P application today is file sharing. The popularity of the P2P network has been triggered by the need of file-sharing over the network, which made the study of performance of such networks an interesting area of research. Searching in unstructured P2P networks is considerably more challenging because of the lack of global routing and dynamic topology. Many search algorithms have been developed in the past. In unstructured P2P systems, most of existing query-based search algorithms shares a common approach: when there is a search request, a peer sends out a query to other peers in the overlay network. When a query travels multiple hops, it may take a long time for the query to be answered. Additionally, the search process incurs multiple query related messages across the network. As a result, more amounts of bandwidth and computing power are consumed at all the involved nodes on the routing path. Excessive usage of queries usually leads to long search latency and high system load. One possible solution is processing a search request in one hop and thus reduces the search cost. Some typical search algorithms are flooding and random walk. Flooding in general covers many nodes but the drawback is it generates huge amount of query messages and because of this the search cost is more. It produces considerable query messages even when the resource distribution is scarce. The search is especially inefficient when the target is far from the query source. On the other hand, random walk (RW) is a conservative search algorithm. By Random Walk(RW) the query source just sends one query message (walker) to one of its neighbors. If this neighbor does not own the queried resource, it keeps on sending the walker to one of its neighbors, except for the one the query message comes from, and thus, the search cost is reduced. The main drawback of RW is the long search time. In this research paper, we present a Proclamation Based Search (PS) Algorithm for unstructured P2P systems which aims to achieve a *one-hop search* with moderate *search cost*. Proclamation is nothing but publishing the synopsis of contents a peer tends to share and cached by other interested peers in the system. Rather than waiting for the peers to send out query messages, it is more appropriate for the system to prepare the interested content index *before* the searches are initiated. To do this, each node in the system publishes its shared contents by delivering compact information with high-level semantics, and selectively stores interesting contents received from other peers. When there is a search request, the node simply looks up its local publishing repository. Comprehensive simulation results show that PS algorithm is efficient in terms of search efficiency compared to other search algorithms.

2. RELATED WORK

Flooding and Random Walk (RW) are two typical examples of search algorithms in P2P network by which query messages are sent to neighbors without any knowledge about the possible locations of the queried resources or any preference for the directions to send. These algorithms rely on queries for content location. The search performance has been continuously improved by new alternatives. To further improve the search efficiency, researchers have proposed several one-hop lookup algorithms. Based on Chord, Gupta *et al.* [14] proposed a one-hop lookup scheme for P2P overlays, in which each node maintains accurate routing tables with complete membership information. For the purpose of disseminating information about membership changes, the system requires relatively

powerful and stable nodes to act as slice and unit leaders. Content publishing is widely used in pub-sub systems where publishers generate events that are consumed by subscribers. Unlike P2P systems, Publish-subscribe systems bear distinct system architecture in that they rely on a dedicated network of routing brokers working exclusively for event propagation. While the search performance was reported promising, the system load tends to be high due to the global gossiping. This could limit the system scalability. Our proposed PS algorithm improves search efficiency.

3. PS ALGORITHM

We present the detailed design of the system in this section. First part explains about the design principle and Second part explains the synopsis representation and the last part explains the search algorithm.

3.1 Design Principle

In this paper, we create the idea of preparing indices beforehand for unstructured P2P systems. Instead of placing them in an ID-matching we develop a more aggressive scheme that pushes the content indices to their potential users, such that user requests can be resolved by simply looking up local indices. Borrowing ideas from the real life play an important role in our real life. People receive a lot of information from a variety of sources, such as newspaper, websites, TV, radio and posters. With different background and specific interests, people collect, keep or remember some of the information that may be useful to them. When having a request, they just find or recall the related sources, go directly and get the product or service. Clearly this is an easier way to obtain information rather than blindly going out to search. Following the same philosophy, we design this system for efficient content location in unstructured P2P systems. The system is designed based on four observations. First, query messages contribute to a large portion of network traffic in P2P system. Second, in content sharing P2P systems, the arrival rate of search requests tends to fluctuate. Third, although peers may come and go freely, contents shared on many nodes do not change very often. Fourth, it is known that interest clustering is common in peer to peer system. It is expected that many peers share common interests and the variety of each peer's interests is limited. Furthermore, most peers are unlikely to change their interests very often assuming a peer only corresponds to one user. This system is being designed based on the above mentioned assumptions. The key approach in this system is that each note will have an index table and this table has to be updated frequently so that the query can be answered in a single hop just by looking up the local lookup. While the search requests continuously increase and fluctuate, the contents are relatively stable as long as the system has warmed up. This motivates us to design a system in which peers proactively distribute and cache content indexes.

3.2 Synopsis representation

In this system a synopsis is comprised of four components:
Node Identification of the node N_i, content information denoted by C_i, a set of topics T_s covered by the node, and a version number V_n. Thus a synopsis s is denoted as an ordered pair (N_i, C_i, T_s, V_n).The identification of the node can be IP address along with a system name. With regard to content information, the system predefines two types: one is with complete index of a peer's contents, the other is incremental index changes since the last update. The version number is a 12-bit integer used for consistently merging index changes. Example of synopsis is shown below

TABLE I: SYNOPSIS REPRESENTATION

Node ID	Content Information	Set of topics	Version Number
I	Index of all the documents Shared	AI DBMS NETWORK	325473467583

The content information with complete index summarizes all the contents shared on a node by using a hash-based data structure representing a set to support membership queries, and has been widely used in P2P system designs. The membership test returns false positives with a predictable probability but never returns false negatives. Assume D_I is the set of documents shared on node I, and $K_I = \{kd \in d_t | d_t \in D_I\}$ is the set of keywords that appear in any document in D_I where kd is a keyword that appears in document d_t. The content filter of node I is initialized by hashing all the keys in K_I and setting the corresponding bits. To determine the topics of a synopsis, we predefine a universal set S of all possible topics in the system, and apply classifications to the contents. We assume each document d_t belongs to a topic $t(d_t) \in S$, and each node n has a set of interests $I(n) \subseteq U$. For example, in music file sharing network like kazaa, music files are classified into different topics such as Rock, Hip-Hop and R&B. A node may be interested in rock and hip-hop but indifferent in any other types. Therefore, the topics of a synopsis s (no matter which type this synopsis is) from node I is denoted as $T(s) = \{t(d_t) | d_t \in D_I\}$. A node a is interested in synopsis s if there is non- empty intersection between $T(s)$ and $I(A)$, where $I(A)$ is the set of node

A's interests. The document classification technique is matured in information retrieval field and out of scope of this paper.

3.3 Search Algorithm

In this system, the synopsis is moved towards interested nodes. So, later when there is a search request, the node can look up locally. Assume that node A request a file which is shared by node I

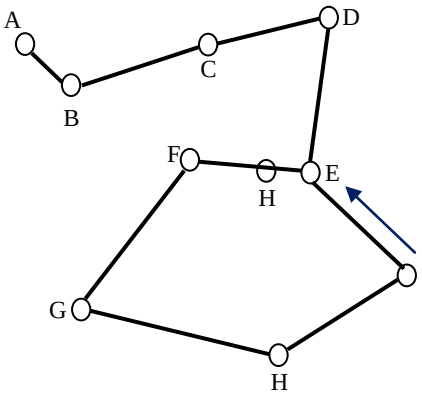

Fig.1.Node I publishes the synopsis

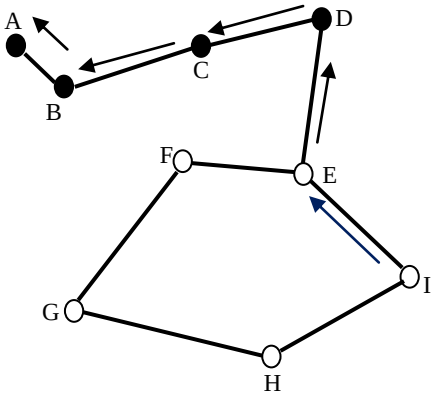

Fig 2.Delivery of synopsis to interested nodes (Shaded nodes)

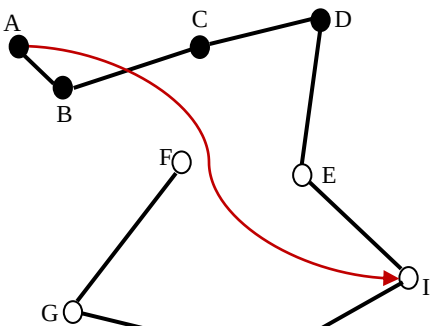

Fig3. The synopsis is delivered before the search request is done by node A and so this node can look up its local repository and can directly contact node I

TABLE II :PS ALGORITHM

```
PS_Search (request z)
//search algorithm running on node A with synopsis
cache $
{
M ← getSearchTerm(z)
        for each synopsis s ∈$
N ← getContentFilterFromSynop(s)
if match(M,N)=true
P ← getSynopSourceNode(s)
Send confirmation message to node I
If more response is needed
 For each neighbor x
A ← requestSynopFromNeighbors(x, h, I(A))
if A=0 return
for each synopsis s' ∈A
N' ← getContentFilterFromSynop(s')
if match(M,N') = true
P'← getAdSourceNode(s')
 send confirmation message to node P'
            $ ← $ ∪A
}
```

Given a request, node A firstly looks up its local repository, and tries to find matching information that contain the search terms. A match by a synopsis s_I (I is the source of this synopsis) indicates that node I has the requested file. Node A needs to send the request to node I for content confirmation, and after a positive match the search is completed with the cost of one hop communication .The above table shows the PS search algorithm. In $A \leftarrow requestSynopFromNeighbors(x, h, I_n(A))$, h is hop number and $I(A)$ is the set of node A's interests. With the help of synopsis cache lookup, most search requests are expected to be answered in one hop. However, if no match is found, or more responses are needed, then node A sends out request messages to its neighbors within a hop distance of h. These neighbors reply to A with their cached synopsis that contain topics overlapping with A's interests. The search is repeated by looking up the

replied synopsis for more possible hits. For most requests from nodes that maintain many synopsis, only one hop communication is needed for a search, delivering an optimal search performance. In the mean time, the search cost only includes content confirmation messages, and only the initiating and destination nodes are involved in the search process. By moving the synopsis towards their consumers beforehand, PS offers one-hop search performance with modest search cost.

4. EVALUATION

Simulation is developed to evaluate the performance of the algorithm compared with other un- structured search algorithms. For experimentation we use an overlay network with 15,000 peers. The logical topology used for experimentation is random. In random topology the nodes are randomly created. Let there be 60,000 physical nodes and we randomly select 15,000 P2P nodes and construct the topology. From a real world system like Kazaa file sharing system we build a trace. There are 8,50,000 files shared among 60,000 peers. The documents shared on these form a universal set. The documents in the universal set are classified into 10 categories. Example in Kazaa file sharing the different categories are Rock,Jazz,Hiphop. Assuming that peer asks for interesting synopsis, a trace has been created containing 40,000 search requests. The network is emulated by inserting 1200 nodes join and 1200 nodes depart. Time stamp for each query event is created. The trace is developed and fed into the system for testing.

5. SIMULATION RESULTS

The search efficiency between other search algorithms are compared by measuring few performance metrics like search performance & cost in a random overlay topology. The search performance is measured in terms of success rate and response time and search cost is measured in terms of bandwidth consumption. Success rate is the percentage of search request that obtains the result and response time is the average of all successful search request. Compared to other search algorithms, PS algorithm obtains high success rate and low response time. Comparison of search cost is done by measuring the bandwidth consumption in Kilobytes.

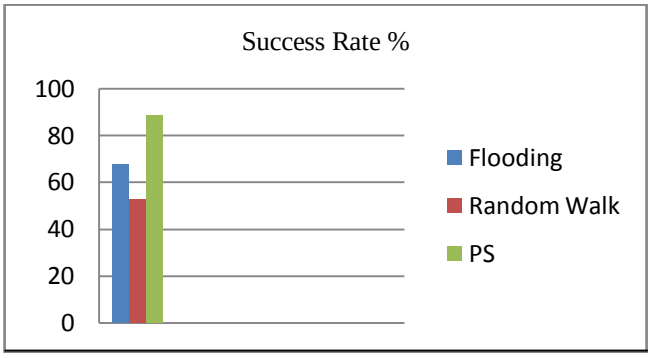

Fig.4.Search success rate for measuring search performance

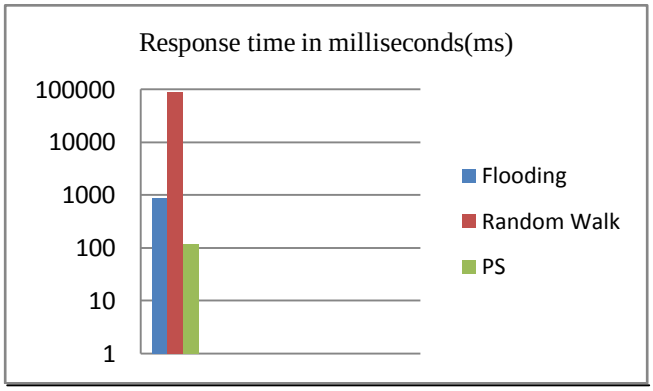

Fig 5.Response time for measuring search performance

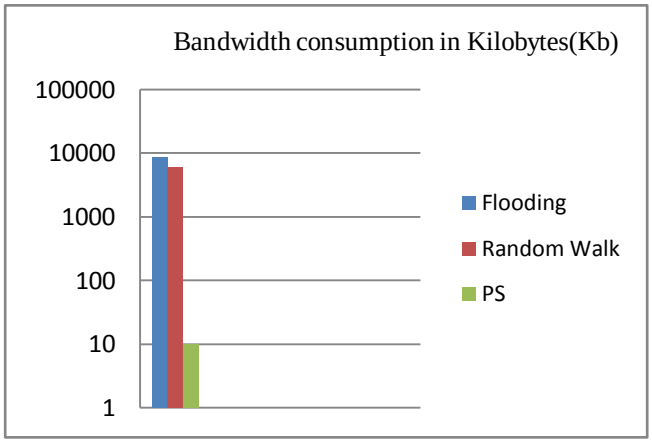

Fig.6 .Bandwidth consumption for measuring search cost

Metrics/Algorithm	Flooding	Random Walk	PS
Success rate	70%	50%	90%
Response time	900ms	90000ms	120ms
Bandwidth	8500Kb	6000Kb	10Kb

TABLE III: COMPARATIVE ANALYSIS

Table III illustrates that PS achieves high success rate, low response time and consumes less amount of bandwidth.

6. Conclusion

In this research paper we propose a new search algorithm for unstructured peer to peer network. Nodes in the system anticipatorily publish the content index to interested peers and each node caches a set of interesting contents. When there is a search request, the node first look up its local publishing repository and resolve the query by just one hop content confirmation. The experimental results show that PS improves the query search efficiency when compared to other search algorithms for unstructured P2P networks like flooding and random walk.

References

[1] R. Bolla, R. Gaeta, A. Magnetto, M. Sciuto, and M. Sereno, "A Measurement Study Supporting P2P File-Sharing Community Models," Computer Networks, vol. 53, no. 4, pp. 485 – 500, 2009.

[2] D. Stutzbach, R. Rejaie, and S. Sen, "Characterizing Unstructured Overlay Topologies in Modern P2P File-Sharing Systems," in Proc. of the ACM Sigcomm Internet Measurement Conference, 2005.

[3] S. Iyer, P. Rowstron, and P. Druschel, "Squirrel: a Decentralized Peer- to-Peer Web Cache," in Proc. of the ACM Symposium on Principles of Distributed Computing (PODC), 2002.

[4] Hailong Cai and Jun Wang. Foreseer: A novel, locality-aware peer- to-peer system architecture for keyword searches. In Proceedings of International Middleware Conference (Middleware 2004), pages 38–58, Toronto, Ontario, Canada, Oct. 2004.

[5] C. Gkantsidis, M. Mihail, and A. Saberi. Random walks in peer-to peer networks. In Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and communications Societies (INFOCOM '04), volume 1, pages 120–130, 2004.

[6] D. Milojicic, V. Kalogeraki, R. Lukose, K. Nagaraja, J. Pruyne, B. Richard, S. Rollins, and Z. Xu, Peer-to-Peer Computing Technical Report HPL-2002-57, HP, 2002.

[7] L.A. Adamic, R.M. Lukose, and B.A. Huberman, "Local Search in Unstructured Networks," Handbook of Graphs and Networks pp. 295-317, Wiley-VCH, 2003.

[8] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A Local Search Mechanism for Peer-to-Peer networks Proc.ACM CIKM Int'l Conf. Information and Knowledge Management(CIKM '02), pp. 300- 307, Nov. 2002.

[9] A. Crespo and H. Garcia-Molina, "Routing Indices for Peer-to-Peer Systems," Proc. 22nd Int'l Conf. Distributed Computing Systems(ICDCS '02), pp. 23-32, July 2002.

[10] Stefan Saroiu, P. Krishna Gummadi, and Steven D. Gribble. A measurement study of peer-to-peer file sharing systems. In Proceedings of Multimedia Computing and Networking (MMCN), San Jones, CA, Jan.2002

[11] Kazaa website http://kazaa.com

[12] Christos Gkantsidis, Milena Mihail, and Amin Saberi. Hybrid search schemes for unstructured peer-to-peer networks. In Proceedings of IEEE INFOCOM'05, Miami, FL, March 2005.

[13] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker. Search and replication in unstructured peer-to-peer networks. In Proc. Of ICS, 2002.

[14] Anjali Gupta, Barbara Liskov, and Rodrigo Rodrigues. One hop lookups for peer-to-peer overlays. In Proceedings of the 9th IEEE Workshop on Hot Topics in Operating Systems (HotOS IX), pages 7–12, Lihue, Hawaii, USA, May 2003.

[15] V. Cholvi, P. A. Felber, and E. W. Biersack. Ef_cient search in unstructured peer-to-peer networks. Technical Report RR- 03-090, Institute Eur_ecom, 2003.

[16]. Network Simulator 2 http://www.isi.edu/nsnam/ns/

Authors Profile

AR.ARUNACHALAM received his B.E degree in Computer Science & Engineering from Madras University in 2002 and received his M.Tech degree in Computer Science & Engineering from Bharath University in 2007. He is currently pursuing his Ph.D in Computer Science & Engineering at Bharath University,Chennai. He has 10 years of teaching experience and has guided many B.Tech and M.Tech projects.

Dr.V.KHANAA received his B.E and M.S degree in Computer Science & Engineering from BITS Pilani and has obtained the Ph.D degree from Bharath University He is currently working as Dean-Information in Bharath University,Chennai,Tamilnadu,India and his research interest are computer networks,digital image processing.