IAC: Interest-Aware Caching for Unstructured P2P*

Xucheng Luo, Zhiguang Qin, Ji Geng, and Jiaqing Luo
School of Computer Science and Engineering
University of Electronic Science and Technology of China
Chengdu, 610054, P.R.China
{xucheng, qinzg, jgeng}@uestc.edu.cn

Abstract

The simplicity and robustness of unstructured P2P system make it a preferable architecture for constructing real large scale file sharing system. Most of the existing paradigms require more overhead. The trace data analysis shows that the workloads among peers are correlated. The bigger the intersection of workloads, the higher the probability that they share other files is. By exploiting such principle, we propose IAC, an interest-aware resource advertisement caching paradigm for unstructured p2p system. Each peer advertises its resource list. If a peer is interested in the resource advertisement received, it then caches the advertisement. Through local cache search, usually the peer gets more than 50% success rate. If local cache search is failed, the random walk-based search is used. The simulations show that as to the same hit rate, the maintenance and search overheads are low and the search delay is very low.

1. Introduction

P2P system is a kind of distributed system which is characterized by large scale, fault-tolerance, self-organization, and etc. P2P computing is one of the basic technologies of Knowledge Grid [1]. The development of P2P computing will nourish the Knowledge Grid. P2P systems can be classified into structured and unstructured P2P system. Unstructured P2P system has flexible query processing and can adapt to churns easily. Resource locating is a core function of P2P sharing system. Because there is no hint about the storage of resource shared, peers in unstructured P2P system locate resource by query flooding. Although flooding algorithm can propagate query to almost all nodes rapidly, the redundant messages may incur congestion in overlay network. In this paper, we concern the performance improvement of Gnutella-based unstructured P2P system.

To reduce the message redundancy of flooding query, many alternatives have been proposed, such as expanded ring, k-random walks [2], and the combination of random walk and shadow flooding [3]. P2PLSNs[4] is another way to improve the performance of resource locating which uses semantic links overlay. Most of the existing search algorithms in Gnutella-like networks are issuing query to all peers. The answer to a query can only serve the requestor. If the sharing information is advertised to all peers, this message can serve all peers. Because the memory used for caching is limited, it is impossible for peers to cache all the Resource Advertisements (RAs) received. As to a specific peer, not all the RAs are useful for it. It is important for a peer to keep the useful RAs in its cache. The trace data of real P2P system shows that the workloads of peers are correlated. The bigger the intersection of workloads, the higher the probability that they share other files is. This important character has been revealed in [5] and [6]. To exploit this principle, a paradigm called interest-based-shortcuts was established in [5]. However, the description of interest patterns in it is not accurate. We propose an efficient paradigm to exploit the interest patterns among peers and the advantage of the advertising mechanism, which is called Interest-Aware Caching (IAC). In our paradigm, interest-based potential is counted and used for caching. Peers need only cache RAs interesting to them. By local cache search, the success rate of resource locating is high and search delay approximates zero. If local cache search is failed, a random walk-based query is used to complete the resource locating. The simulation shows that the message overhead of IAC is no more than that of other paradigms.

* This work is supported by National Natural Science Foundation of China under grant numbers 60473090 and 60573129.

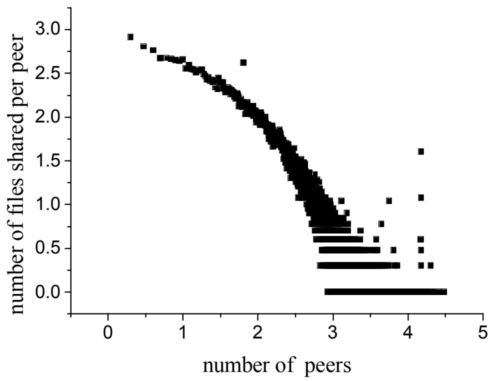

Figure 1. Distribution of the number of shared files (log-log scale)

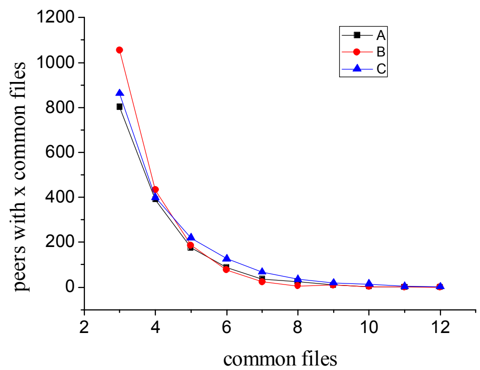

Figure 2. Amount of common files vs. the amount of peers with x common files for peer A, B, and C.

The remainder of the paper is organized as follows. In section 2, we analyze the trace data of a real P2P system. In section 3, the design of IAC is presented. The analysis of IAC system is described in section 4. To evaluate the performance of IAC, we do simulations in section 5. Section 6 is the conclusion.

2. Characterizing workload

In P2P file sharing system, the workload properties are very useful for design of efficient search scheme. [5] [6] have revealed that there is common interest between peers. This common interest is expressed by the files shared. To efficiently exploit such attribute, other properties should be uncovered. We concern the replica distribution, the number of files shared by each peer, and the relationship among the workloads of peers. These properties are obtained by the trace data analysis. The trace data analyzed in this paper is the same as that of [7]. The distribution of numbers of files

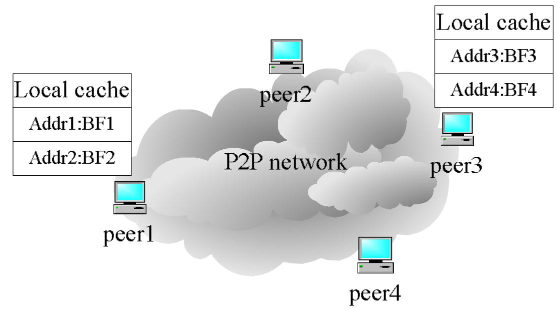

Figure3. Design of IAC system

shared by peers is illustrated in Figure1. This property reveals that a few files are extremely popular while a large number is replicated at several peers. To analyze the relationship among the workloads of peers, we choose three peers A, B, and C randomly. The trend is illustrated in Figure2. Majority of peers have few common files with some other peers. With the increase of number of the common files, the number of peers is decreased. The results of trace data analysis are: 1) there are common interests among peers. This attribute can be exploited to improve the search performance. 2) The bigger the intersection of workloads, the higher the probability that they share other files is. 3) If the popular files are used to measure the sharing interest, the result is not accurate.

3. Design of IAC

Most of the existing search algorithms of unstructured P2P system are issuing queries to all peers or some limited group of peers in the system. Usually, the answer to a query can only serve the requestor. We call this resource locating model as Query Advertising Model (QAM). In contrast, if we advertise the files list of a peer, this information can serve all peers received the advertisement. We call the second model as Resource Advertising Model (RAM). The message overhead of QAM and RAM is the same. Because a RA can serve multiple peers, the search cost in IAC is far lower than that of other paradigms. In IAC, peers construct the RA which includes the owner of the RA and the Bloom filter represent of the files set. Peers cache the RAs interesting to them. As to resource locating, peers firstly resort to Local Cache Search (LCS) and then degrade to the random walk-based search algorithm if the results of LCS are not good enough. The design of IAC is illustrated in Figure3. In section 3.1, we describe the resource presentation and propagating. The detailed design of RAs caching is described in section 3.2.

3.1. Resource advertising

Like the marketing behavior, peers with files to share propagate the resource list to other peers of the system. Efficient scheme is required to represent the resource list. Cost-efficient message propagating algorithm should be designed to disseminate the RAs.

Bloom Filter is a space-efficient probabilistic data structure which is used to represent a set and support the membership query. A detailed survey of network applications of Bloom filter is given in [8]. In IAC, the Bloom filter data structure is used to represent the set of files to reduce the overheads of advertising. To exploit the clustering of workloads, the size of the intersection of Bloom filter should be evaluated. Letting z_1 and z_2 be the number of zeros in the two Bloom filters respectively and z_{12} be the number of zeros in the inner product of the two Bloom filters. The size of the intersection approximates $\frac{\ln(z_1+z_2-z_{12})-\ln z_1 z_2+\ln m}{k(\ln m-\ln(m-1))}$.

Unstructured P2P system is a large scale dynamic system. Peers join and leave the system autonomously. Efficient message propagating algorithm is needed to disseminate the RAs. We concern the message cost, fault-tolerance, and speed of the message propagating algorithm. Although flooding is robust and rapid, the message cost is very high. The probabilistic broadcast algorithms are better alternatives. In this paper, the selection of message propagating algorithm does not impact the performance analysis. We treat the message overhead of advertising for a peer the same as that of a normal query.

3.2. Resource advertisements caching

Each peer of IAC allocates a cache to store the interesting RAs. The interest of each peer is described by the files shared by the peer. If a peer has no files to share, the files obtained in the past can be used to describe the interest. When a peer received a RA, it compares the Bloom filter in the RA with its own. If the overlap of the two sets is bigger than the threshold, the probability that they share other files is high. The difference of the two sets determines the potential utility of the RA. It is not practical for a peer to caching excessive RAs. Peers of the IAC set the size of cache according to their capacity. The bigger the size of the cache, the higher success rate of search is. Peers rank the RAs in the cache. If the cache is full, the lowest ranking RA is discarded. The caching algorithm is described in algorithm 1.

Algorithm 1 RACaching(RA)

BF_c //bloom filter of files set of the current peer

BF_p //Bloom filter of files set of the RA

$n_p = |BF_p|$

$x = |BF_c \cap BF_p|$

$y = |BF_p \setminus (BF_c \cap BF_p)| = n_p - x$

If ($x \ge t_i$ && $y \ge t_p$) then

Insert_Cache(RA)

The two key parameters of the above algorithm are t_i and t_p . t_i is the threshold of the interest matching. If t_i is very small, such as 1 or 2, the amount of suitable RAs would be very large. On the other hand, if the t_i is large, such as 10 or bigger, the amount of suitable RAs is decreased. However, the search success rate is decreased rapidly. We divide the peers according to the number of common files into subsets and test the search success rate in different subset. The results are illustrated in Figure4. t_i can be determined by users according to the capacity of the node. It is advisable to set t_i as 4. This value can guarantee more than 50% success rate and a moderate number of RAs cached. t_p is the potential of the RA to the current peer. If t_p equals zero, it is obvious that this RA has no use for the current peer. The bigger the parameter t_p , the higher probability the search success rate is. IAC ranks the RAs in the cache by firstly t_i and then t_p . The lowest ranking RA is discarded for the incoming of high ranking RA when the cache is full.

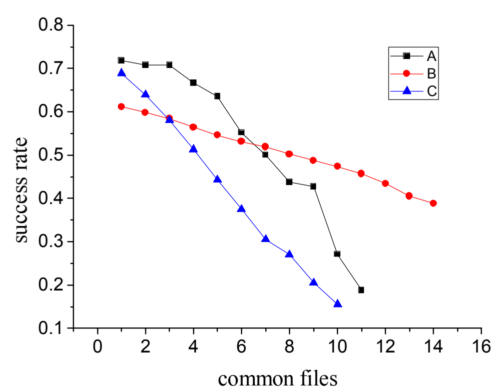

Figure 4. Success rate vs. number of common files

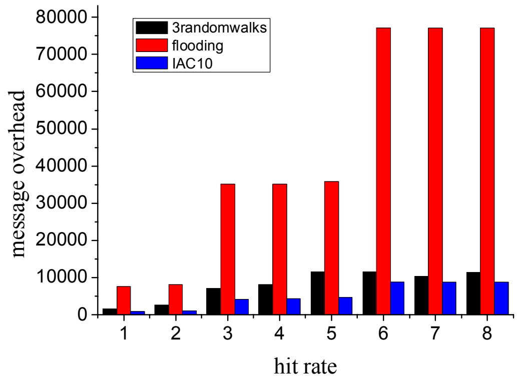

Figure 5. Search hit rate vs. message overhead

4. Analysis of IAC

The P2P search network is $G = (V, E)$. V is the set of peers and $|V| = N$. $E = V \times V$ is the set of edges between peers. d_v is the degree of peer v. $d = \frac{1}{|V|}\sum d_v$ is the average degree. As to blind search in P2P network, the query is propagated to all peers. The number of messages is M_p. If flooding is used, $M_p = \sum_{i=0}^{TTL} d^i$. In IAC, a RA needs to be propagated to all peers. The message overhead incurred in a RA propagating is M_p by using the same message propagating algorithm. If a RA includes f files and serves r requestors, the message overhead for a file locating is between M_p / rf and M_p / r. Because the local cache search has high success rate, the query delay usually approaches zero. If the search degrades to random walk-based scheme, the search delay is far below that of normal random walk because of the RAs caching.

5. Simulation of IAC

The message overhead and search hit rate are concerned in the simulation. In the simulation, the overlay networks are power law network of 50000 peers. The content shared by peers is extracted from the trace data. The record is selected randomly from the trace data. To reflect the fact, 50% free riders are deployed in the simulation. The search network constructed above is called the Original Search Network (OSN). In the simulations, the overheads of flooding query and k-random walks on OSN are obtained. The parameter r is set to 10. The results are illustrated in Figure5. IAC has lower overhead than that of the other two schemes.

6. Conclusion

Resource advertising propagates information about resource shared by peers. Advertising for a peer can serve multiple peers, while a query can only serve the requestor. The analysis of trace data shows that there is common interest between peers. Exploiting this principle, peers in IAC only cache RAs interesting for them. By this way, the search success rate is improved and the cache space is saved. The simulation shows that IAC has far lower message overhead and query delay than that of the other two paradigms.

7. References

[1] H. Zhuge, The Knowledge Grid. World Scientific Publishing Co., Singapore, 2004.

[2] L.v. Qin, P. Cao, E. Cohen, Li. Kai, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," Proceedings of the 16th international conference on Supercomputing, ACM, New York City, 2002, pp. 84-95.

[3] M.M. Christos Gkantsidis, Amin Saberi "Hybrid Search Schemes for Unstructured Peer-to-Peer Networks," INFOCOM05, IEEE, Miami, Florida, 2005, pp. 1526-1537.

[4] H.Zhuge, et al. "Query Routing in a Peer-to-Peer Semantic Link Network". Computational Intelligence 21(2), Blackwell Publishing, 2005, pp. 197-216.

[5] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient Content Location Using Interest-Based Locality in Peer-toPeer Systems," INFOCOM'03, IEEE, San Francisco, CA, USA, 2003.

[6] F.L. Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulie, "Clustering in Peer-to-Peer File Sharing Workloads," IPTPS'04, Springer, La Jolla, CA, USA, 2004, pp. 217-226.

[7] Daniel Stutzbach, Shanyu Zhao, and Reza Rejaie, "Characterizing Files in the Modern Gnutella Network: A Measurement Study," SPIE/ACM Multimedia Computing and Networking, ACM, San Jose, CA, Jan. 2006.

[8] A. Broder, and M. Mitzenmacher, "Network applications of bloom filters: A survey," Internet Mathematics vol. 1, no. 4, A K Peters, 2005, pp. 485-509.