

Research on the Search Scheme for Rare Items in Unstructured P2P Networks

Meng Xianfu, Wang Hao

School of Computer Science and Technology, Dalian University of Technology, Dalian 116024, China
xfmeng@dlut.edu.cn

Abstract - Search method is the key technique in unstructured P2P systems. The existing search methods are efficient for locating popular items, but not so for locating rare items. Therefore increasing the search efficiency for rare items would improve the network availability and user experience. In this paper, we propose a novel search scheme for rare items. First, we present a data structure of TDBF for counting the files' local popularities by using Bloom Filter data compressing technique. Second, a replication strategy and a dynamic search strategy for rare items are proposed, which could make fully use of the heterogeneity of the P2P networks and also take the load balance into account. We conduct comprehensive simulations to evaluate the algorithm. The simulation results show that our approach could significantly improve the search quality of the rare items as well as reduce system traffic overhead.

Index Terms - P2P, rare item, replica, Bloom Filter

1. Introduction

The objective of a search scheme is to successfully locate resources while incurring low overhead and delay [1]. A number of search algorithms have been proposed for effective resource locating. Search methods for unstructured peer-to-peer networks can be mainly categorized in two ways: blind search and informed search [1-3]. In the blind search, nodes do not keep information about item locations. In the informed search, nodes try to obtain some knowledge about item locations and forward their messages according to the history records stored in their local caches.

A recent measurement of Gnutella networks [2] has shown that these Gnutella-like networks are effective for locating highly replicated items, these popular files in Gnutella not only return in large quantities, but also get good response times. However, for rare items, Gnutella is not so efficient. 41% of all queries receive 10 or fewer results, and 18% of queries receive no results. In addition, these results have poor response times. For queries that return 10 or fewer results, 50 seconds passed on average before receiving the first results [2].

As the observation above, there are a considerable number of queries for rare items, and there is considerable opportunity to improve the recall and response time of the queries for rare items.

To deal with the problems above, we propose a search scheme for rare items, and the main work of this paper is the following.

A. A Bloom Filter based local popularity management structure is proposed;

B. We present a placement strategy for the replicas of rare resources;

C. Finally we propose a dynamic search scheme according to user behavior.

2. Related Work

Expanding ring [4] tries to not involve too many nodes if acceptable result can be retrieved, because, in many cases, it is not necessary to cover the majority of the network to retrieve relevant resources, especially for those popular resources. The main shortcomings of expanding ring are the generation of large amount of duplicate messages and much response time for the rare items.

Light flood [5] is based on the idea of FloodNet, which is a tree-like overlay built on peer-to-peer networks. The overlay can be constructed with local information only. However, Light flood will greatly increase the diameter of the whole network because many existing neighbors between nodes are ignored.

A hybrid structure, QRank [6], was proposed by combining an unstructured peer-to-peer approach with a structured DHT-based index, which only publishes the rare items into the DHT-based network. Searches are performed either by flooding the unstructured network, or by looking up in the DHTs, according to the popularity of desired items. The main disadvantage of the hybrid structure is that the ultra DHT subnet changes the unstructured network topologies.

3. Management of File Local Popularity

If a node could be aware of the file popularity in local area (within its two or three hops), the node would make a better decision on how to propagate the request messages. To do so, we design the data management structure of TDBF, the short for time and distance decayed bloom filter. TDBF is actually a synopsis of local popular files and compressed in a Bloom Filter [8] form. When a node creates its TDBF, it exchanges its TDBF with neighbor peers, and merges the TDBFs gotten from its neighbors with its local TDBF. To reflect the dynamic changes in the network, peers should swap these messages periodically. By means of Bloom Filter technique and attenuating the TDBF bits through time and distance, TDBF can adapt to the dynamic changing of file popularity, and control the message exchanging in local area other than in the whole network. Therefore, the nodes could get the local file popularity at a low cost. Next, we give the TDBF formally.

Let $BF(x)$ be the Bloom Filter vector for a local resource x , $BF(A)$ represent the Bloom Filter vector for the local resources in node A , and $S(A)$ represent the set of local resources.

Definition 1: File access frequency vector is defined as

$S = \{(r_1, t_1, fr_1), (r_2, t_2, fr_2), \dots, (r_N, t_N, fr_N)\}$. Assume that each node stores N files, and each file is stored in a three tuple form, (r_i, t_i, fr_i) , $1 \leq i \leq N$, where r_i is the name of i th item, t_i is the time stamp, and fr_i is the file access frequency. Each node also stores the query with query hits which it forwarded before.

Definition 2: Query forward frequency vector is defined as $Q = \{(q_1, t_1, fq_1), (q_2, t_2, fq_2), \dots, (q_N, t_N, fq_N)\}$. In the three tuple (q_i, t_i, fq_i) , $1 \leq i \leq N$, q_i is the i th query, t_i is the time stamp, and fq_i is the query forward frequency.

Definition 3: Time decaying function $\phi(t)$ is a special non-increasing and non-negative function. It satisfies the following conditions:

- A. $\phi(0) = 1$;
- B. $\phi(t)$ is non-increasing;
- C. $0 \leq \phi(t) \leq 1$ for all $t \geq 0$.

Let f be the access frequency of an item, and $f^{(i)}$ be the access frequency of the item between time $\langle t_{i-1}, t_i \rangle$. Then $f = \sum_i f^{(i)}$. The access frequency of the item with the time decayed value can be given as follows:

$$f^* = \sum_{i=1}^n \phi(t_i) \cdot f^{(i)}$$

$$\phi(t) = \lambda^{t/T}, 0 \leq \lambda \leq 1, T > 0$$

where T and λ are parameters used to control the decayed level.

Next, we employ the file access frequency vector, query forward frequency vector and time decaying function defined above to detail the computation of TDBF. Given a set SQ , whose items are gotten from set S and Q which have greater access frequency than a threshold t . Each SQ item only stores one element of r or q , the first item of the above defines three tuple, and takes the form of $SQ = \{sq_1, sq_2, \dots, sq_n\}$. So the set SQ is a popular synopsis computed locally within a time interval. After we get the local popular synopsis set, we compress it to a Bloom Filter vector represented by $BF_{SQ}(A)$. The nodes in the network calculate this synopsis periodically and exchange it with its neighbors. When a node receives a new synopsis, it merges them to update its TDBF.

4. Heterogeneity of a P2P network

We exploit peer heterogeneity with respect to the file storage, node degree, node capacity and load balance. We can model the capability of a node in a vector form.

Node degree is the number of neighbors that a node connects to. A node who maintains a large amount of neighbors usually has a good capacity in terms of bandwidth and other performance. Also, a node who holds more shared files is more likely to have the requested files.

Definition 4: Assume that the max degree in the network is k_{\max} , in the case of power law network $k_{\max} = N^{1/\tau}$. If a node has the degree of $d \geq k_{\max} / 2$, then we call it the high degree node.

Theorem 1: The high degree node in the network is in the order of $O(N^{\frac{1}{\tau}})$.

In this paper, we will make use of the above described heterogeneity of the P2P network to design our replica placement strategy and the dynamic search scheme.

5. Rare Item Replica Placement

When a node joins the network, the node first communicates to its neighbors, and gets TDBF. Then the node scans the local resource list and calculates the similarity with TDBF of every item it stores according to the formula below.

$$Sim(x, T) = \frac{\sum_{i=1}^m (U[i] \times V[i])}{\sum_{i=1}^m U[i]}$$

where U represents the Bloom Filter bit array of resource U and V stands for the file's TDBF.

After the similarity of each item is calculated, the node replicates the item whose similarity is below a threshold (it is the rare item) to the proper high degree nodes as follows.

When a node wants to replicate a file to other nodes, it first creates a message with message ID= REPLICIA-PLACE. Then the message is propagated according to the high degree biased random walk protocols (i.e. the message is propagated on the path with higher degrees). When a high degree node receives a message with ID=REPLICIA-PLACE, it checks if it has already held the replica, and if not it checks whether it has the resources to store one. If the node is able to store a replica of the item, it informs the initiator.

6. Dynamic Search Scheme

In this section, we propose our dynamic search scheme based on TDBF. When a node initiates a query, it first converts the keyword into the Bloom Filter vector form, and calculates its similarity with TDBF. If the similarity is larger than threshold T , it is considered that the query is for the popular items, and if the similarity is smaller than threshold T , it is considered that the query is for the rare items. Next, we give the detail of our dynamic search scheme.

In the case of popular item locating, a node first initiates a flooding search with TTL=4. When the query hits, the forward node composes a query result message and tries to send the result back to the original query issuer. When the TTL decreases to 0, the node does not drop this message and sends an enquiry message to the query issuer to ask whether it has gotten enough results. If the issuer has gotten enough results, it

doesn't response to this message and if not it randomly chooses k nodes who send the enquiry messages to start a new random walk phase.

In the case of rare item locating, a query issuer directly starts a k-walkers high degree biased random walk. For the rare item replicas are placed on the high degree nodes, the search process is in the same way as that of the replica placement, and the rare items locating will succeed with high probability.

We divide the messages of our search scheme into two types: the popular item query messages and the rare item query messages. As shown in Fig. 1, a query message for popular items contains seven fields, in which *fTTL* represents the TTL value of flooding phase, *tag* is a flag used to distinguish flooding phase and random walk phase. When *tag=0*, search is in the flooding phase, otherwise search is in the random walk phase. *rTTL* represents the TTL value of random walk. Fig. 2 is the structure of a query message for rare items.

Qpop	qid	Key	path	fTTL	tag	rTTL
------	-----	-----	------	------	-----	------

Fig. 1 Query message of popular items

Qrare	qid	Key	path	TTL
-------	-----	-----	------	-----

Fig. 2 Query message of rare items

A node who received a message will determine how to forward it according to the structure of the received message.

7. Simulations and Performance Evaluations

A. Simulation Settings

The simulations examine the effectiveness of our strategy in terms of the number of the retrieved results, average recall rate, succeed rate and search efficiency in comparison with QRank [6] and Bubblestorm [7].

B. Performance comparison

Fig. 3 shows the average number of retrieved results in 50 simulation cycles. We see that our strategy could get more retrieved results than the other ones. Statistically, the average number of retrieved results is improved by 15 percent.

Fig. 4 indicates the relationship between item popularity and search success rate. From Fig. 4 we see our strategy has higher success rate than QRank and Bubblestorm. QRank applies the DHT method to index rare items for locating them, and suffers from extra cost. Our method does not maintain any specific structure or super node, and thus gets better success rate than QRank when searching rare items. For the rare item replication scheme, we can achieve better efficiency and keep the communication cost in a low level.

Fig. 5 plots the statistical average recall rate in each cycle. The statistical average recall rate is increased by 15% percent when the search scheme applied in this paper.

Fig. 6 illustrates the search efficiency of our strategy in comparison with Bubblestorm. From Fig. 6 we see that our search scheme sends many query messages in the first three hops and gets fast search efficiency growth, and the search

efficiency gets stable as the hop increases. This benefits from the two phases of our search scheme. In the first phase, small hops flooding makes the search efficiency grow rapidly, and in the second phase, the high degree biased random walk algorithm makes the search efficiency grow stably.

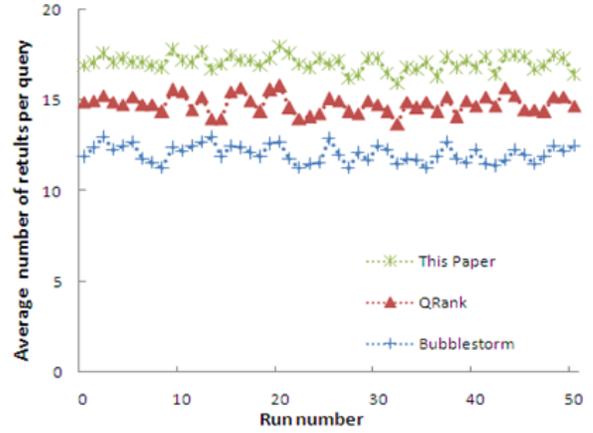


Fig. 3 Average number of results

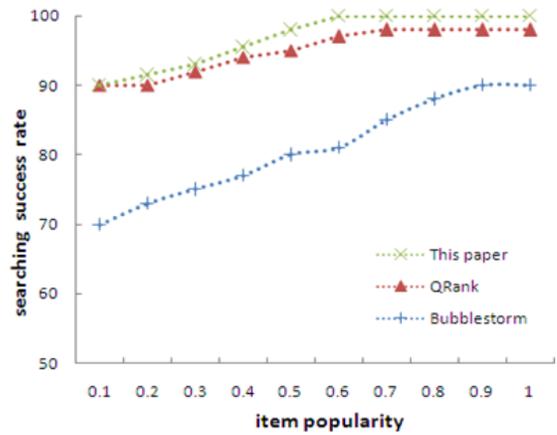


Fig. 4 Searching success rate

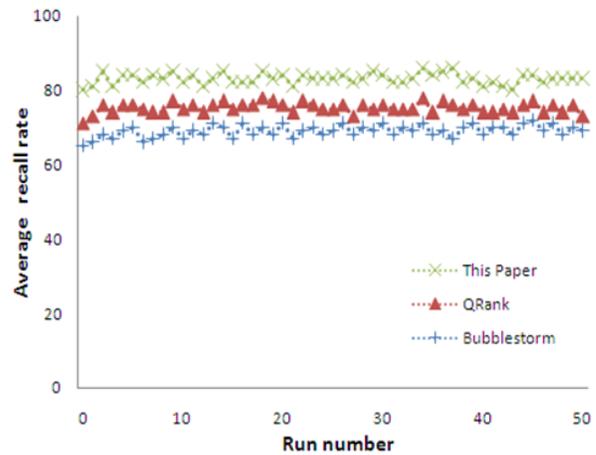


Fig. 5 Average recall rate

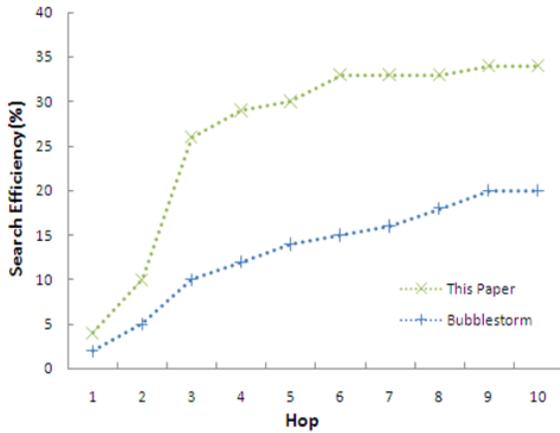


Fig. 6 Search efficiency

8. Conclusions

In this paper, we proposed a novel search algorithm for unstructured P2P networks. We first presented a data structure TDBF for calculating the local item popularity. Then, the rare item replica placement and the dynamic search scheme are proposed. By using these methods, the search efficiency of rare items is significantly improved, while keeps the search efficiency of popular items in a good level. Experimental

results showed that our strategy is effective and efficient in terms of retrieval success rate, retrieval cost and communication overhead.

References

- [1] Tsungnan Lin, Pochiang Lin, Hsinping Wang, Chiahung chen, Dynamic search algorithm in unstructured peer-to-peer networks. *IEEE Transactions on Parallel and Distributed systems*. 2009, 20(5):654-666.
- [2] Adamic L.A, Lukose R.M, Huberman B.A. Local search in unstructured networks. *Handbook of Graphs and Networks*. Wiley-VCH, Berlin 2003.
- [3] B.T.Loo, R.Huebsch, I.Stoica, J.M.Hellerstein, The case for a hybrid P2P search infrastructure. *Proceedings of the International Workshop on Peer-to-Peer Systems, IPTPS, 2004*.
- [4] Yang H. Garcia-Molina. Improving search in peer-to-peer networks. *Proceeding of the 22nd IEEE International Conference on Distributed Computing (IEEE ICDCS'02)*, 2002.
- [5] Song Jiang, Lei Guo, Xiaodong Zhang, Haodong Wang. LightFlood: Minimizing Redundant Messages and Maximizing Scope of Peer-to-Peer Search. *IEEE Transactions on Parallel and Distributed Systems*, 2008, 19(5): 601-614.
- [6] Hanhua Chen, Hai Jin, Yunhao Liu, Lionel M.Ni. Difficulty-Aware hybrid search in peer-to-peer networks. *IEEE Transactions on parallel and distributed systems*. 2009, 20(1): 71-79.
- [7] Terpstra WW, Kangasharju J, Leng C, Buchmann AP. Bubblestorm: Resilient, probabilistic and exhaustive peer-to-peer search. *Proceedings of the ACM SIGCOMM, 2007, New York*.
- [8] Zhu GM, Guo DK, Jin SY. P2P probabilistic routing algorithm based on data copying and Bloom Filter. *Journal of Software*, 2011, 22(4): 773-781.