2011 2nd International Conference on Networking and Information Technology
IPCSIT vol.17 (2011) © (2011) IACSIT Press, Singapore

Multi-keyword search over P2P based on Counting Bloom Filter

Wang Meng-Fan[1], Zhang Da-Fang[2], Tian Xiao-Mei[2], Bi xia-an[2] and Zeng Bin[3]

[1]College of Information Science and engineering Hunan University, China Mobile Group Hunan Company Limited, Changsha Hunan China

[2]College of Information Science and engineering Hunan University, Changsha Hunan, China

[3]China Mobile Group Hunan Company Limited, Hunan, China,

wangmengfanwangzhi@163.com,dfzhang@hnu.cn,13974880055@139.com

Abstract. There are two basic modes for the existing multi-keyword searching over P2P network based on bloom filter (BF), which are "and query" and "or query". Generally, BF encoding is used to handle multi-keyword searching to reduce traffic. Because of the unique characteristics of P2P network, there is a problem: in P2P networks, the joining and leaving of nodes is free, the documents stored on the nodes will be removed as the departure of nodes. However, bloom filter does not support delete operation, when documents removed from the network frequently, the system needs to re-construct bloom filter to accommodate the remove of documents constantly. It will lead an additional system overhead. To that end, a multi-keyword searching mechanism based on counting bloom filter (CBF) is proposed in this paper. This mechanism can adapt to system's churn and reduce traffic. To evaluate the performance of this design, we performed extensive simulations on PEERSIM, simulation results show that this mechanism outperforms current multi-keyword searching mechanisms based on bloom filter in both the recall rate and the query latency, when the nodes join and leave from the P2P networks frequently, thus improves the search efficiency greatly.

Keywords- P2p networks, Multi-keywords searching, DHT, Bloom Filter, Counting Bloom Filter, Inverted index

1. Introduction

P2P networks can also be called peer-to-peer networks or peer-to-peer computing. It can improve network efficiency, with good scalability and strong fault tolerance. Therefore, P2P has become a hot spot of current scientific research and product development. The resources sharing is one of the most important application in P2P network, so further study on P2P search mechanism has very important significance.

P2P search mechanism can be divided into two categories: unstructured P2P search mechanism and structured P2P search mechanism. Unstructured P2P networks, Gnutella[1] and KaZaA[2],support multi-keyword searching by flooding. Structured P2P networks, such as, CAN[3], Chord[4],Pastry[5] and Tapestry[6], can handle single-keyword exact-match lookups. However, in practice, multi-keyword searching is more popular. Current multi-keyword searching generally is based on single-keyword searching. First of all, multi-keyword are split into several keywords. Secondly, each keyword conducts single-keyword searching. Finally, the results of distributed intersection/union operations of each single-keyword search are regarded as the results of multi-keyword searching. This will leading to unacceptable traffic costs.

Current researches on multi-keyword searching focus on the decrease of number of messages, Bhattacharjee et al[7] proposed an orthogonal and complementary technique: using result-caching to avoid duplicating work and data movement. This method handle multi-keyword searching through inverted index of intersection operations, will generate overmuch network traffic. Reynolds et al [8] used bloom filters[9,10,11], caches, incremental results to reduce network traffic. By sending a bloom filter based on documents instead of

sending documents itself to compress network traffic. While caches takes advantage of the locality in the list of words searched by a user to reduce network traffic. Yang et al [12] proposed a keyword search scheme be called Proof; the key idea is storing a content summary for each web page in the inverted list, so that a query can be processed by only transmitting a small size of candidate results. so decreases network traffic. Rai et al [13] proposed a multi-keyword searching mechanism in structured P2P networks, which groups peers with similar searches together, the formation and maintenance of groups are in a self-organizing fashion, peers in the same group share their cached results. Chen et al [14] researched two basic multi-keyword searching query modes in structured P2P networks: “and query” and “or query”. They also proposed the optimal order strategies for both “and” and “or” queries.

As is known to all, bloom filter is a widely-used data structure that concisely represents a large set of contents for approximate membership queries. It offers good space efficiency at the cost of reduced accuracy, which has been applied in multi-keyword searching mechanism in the recent years. Although bloom filter has good space efficiency, an extremely important feature of the P2P networks cannot be ignored: the joining and leaving of the nodes in the P2P networks is free, so the documents stored on the node will be removed as the departure of the node. When the documents are removed from the network frequently, the system needs to re-construct bloom filter constantly to accommodate the deletion of documents, and this will lead to additional system overhead.

To address this issue, a multi-keyword searching scheme based on counting bloom filter is be proposed for structured P2P networks. Which uses counting bloom filter to store keyword indexes, CBF will be converted to BF before transferring between nodes. Thus, this scheme can adapt to document’s deletion from network and also can reduce traffic, thus improve search efficiency.

2. Related Work

2.1 Bloom Filter

A bloom filter [15] is a hash-based data structure that summarizes membership in a set that supports approximate membership queries. The principle of the bloom filter is hashing the elements in the collection to a bit vector by k hash functions, using the m-bit of the bit vector representation of the set of n elements, each element only need m/n bits on average, thus greatly saves storage space.

In standard bloom filter, all bits are shared by the elements in collection, so it only supports the insertion but does not support the deletion of elements. To support deletion of elements, counting bloom filter are proposed.

Here, supposing that collection A include two elements, X and Y, fig.1 shows the insertion of standard bloom filter, fig.2 depicts the insertion of counting bloom filter.

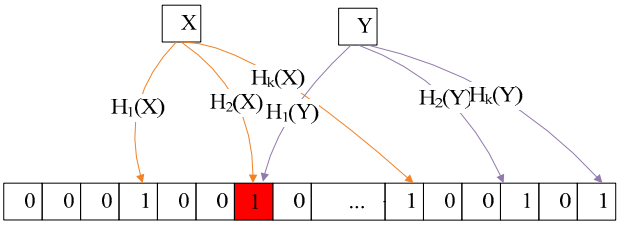

Figure 1. Insertion of standard bloom filter.

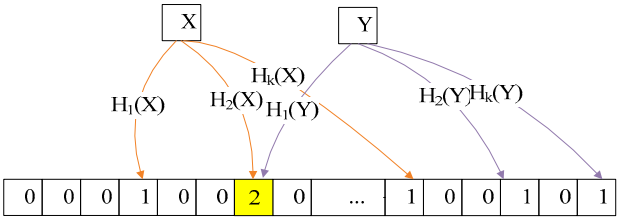

Figure 2. Insertion of counting bloom filter.

By comparison of fig.1 and fig.2, this fact can be seen that bloom filter cannot support the deletion of elements due to bit sharing, but counting bloom filter can. Fig.3 shows us the counting bloom filter how to remove “Y”.

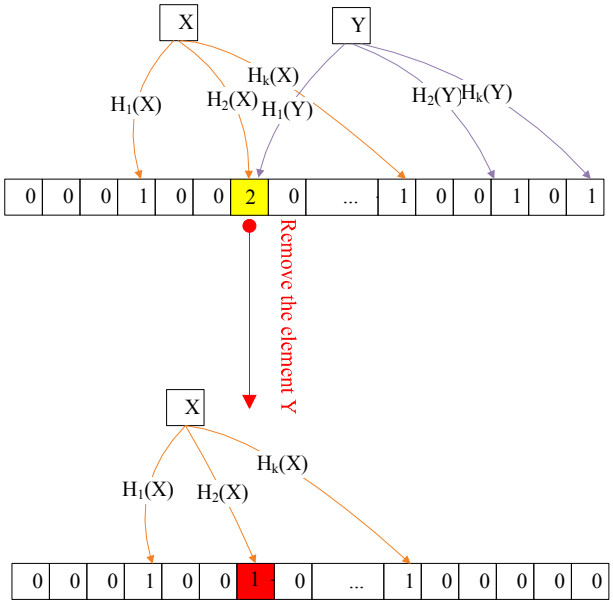

Figure 3. Deletion of counting bloom filter.

2.2 Lucene Index

Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

In this paper, inverted index established by lucene, as fig.4 shows, it is the procedure of construct inverted indexes for experiment data.

As fig. 5 shows, lucene index is composed by several segments, each segment is composed by several documents, each document is composed by several fields, each filed is composed by several terms, and term is the smallest index unit, it represents a string and its location, occurrences and other information in the document directly.

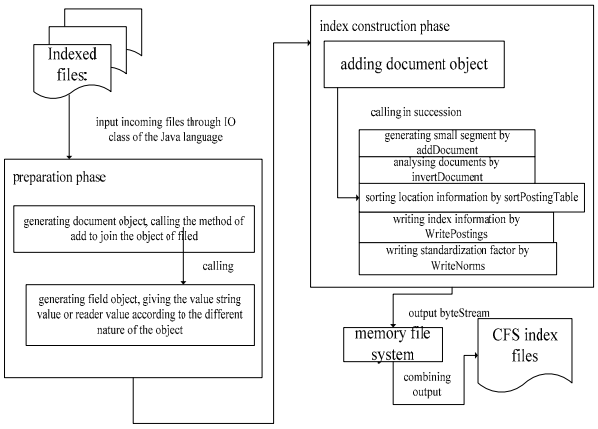

Figure 4. Flow chart of constructing inverted index.

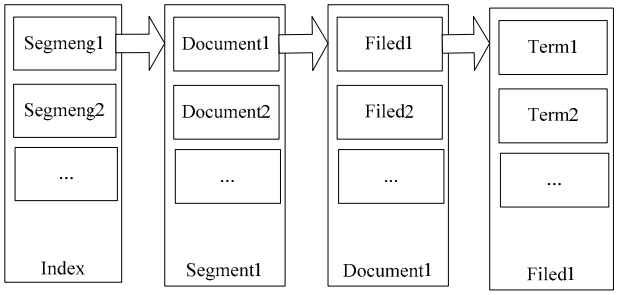

Figure 5. Logic structure of lucene index.

3. Muti-keyword Searching Based On Counting Bloom Filter

In P2P network, existing multi-keyword searching based on bloom filter have two basic models: "and query" and "or query".

3.1 And Query

A common solution for a multi-keyword search needs conducting a distributed intersection operation in a wide area network. fig.6 gives an example of a two-keyword (x, y) "and query" based on bloom filters.

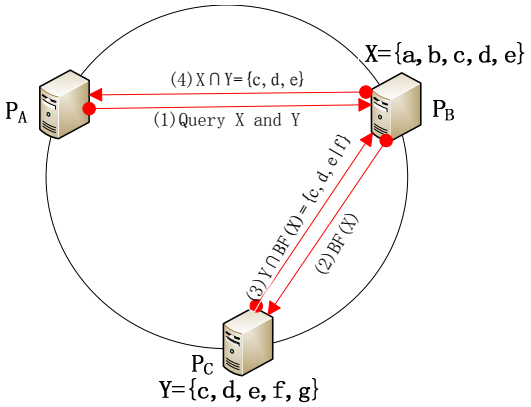

Figure 6. "And query" based on BF.

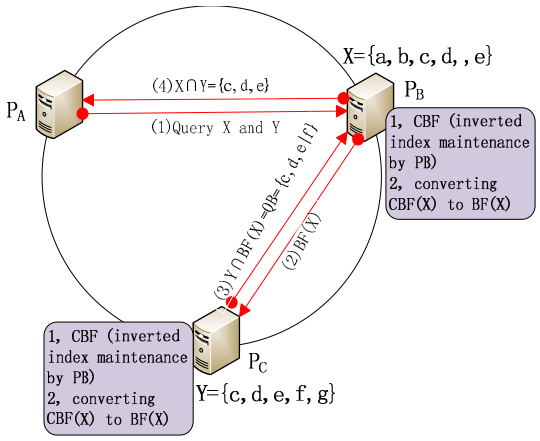

Figure 7. "And query" based on CBF.

Fig.7 presents an example of "and query" based on counting bloom filters. The process is as follows:

1) P_A launching an "and query", inquires X and Y, it founds the index of keywords X maintenance by P_B, P_A sends query to P_B;
2) The index table of keyword X maintained by P_B storing by counting bloom filters, P_B converts CBF(X) to BF(X);
3) P_B sends BF(X) to the node of P_C, which maintained the index of keyword Y;

4) P_C checks the keyword Y whether in the index of keyword X maintenance by P_B, gets the result of Y ∩BF(X) and returns it to P_B;
5) P_B returns to the intersection of X∩Y∩BF(X), which equals to the intersection of keyword X and keyword Y.

It is worth mentioning that, in the fourth step, there are certain false positives of bloom filter, so the results of Y∩BF(X) contains some results do not belong to X∩Y, as show in fig.7, Y∩BF(X)={c,d,e/f}, the "f" is not belong to the X∩Y. the role of the fifth step get rid of false positives, if the user can tolerate a small amount of error, then the fifth step can be omitted, and in the fourth step PC returns the result which include a small amount of errors to PA directly.

3.2OR QUERY

In some applications, we need "or" queries, which desire the results containing any keyword in the query. Such query is critical for queries whose keywords are rare in the system. A search engine may combine both the "and" and "or" results for a multi-keyword query to the users. Fig.8 gives an example of "or query" based on bloom filter.

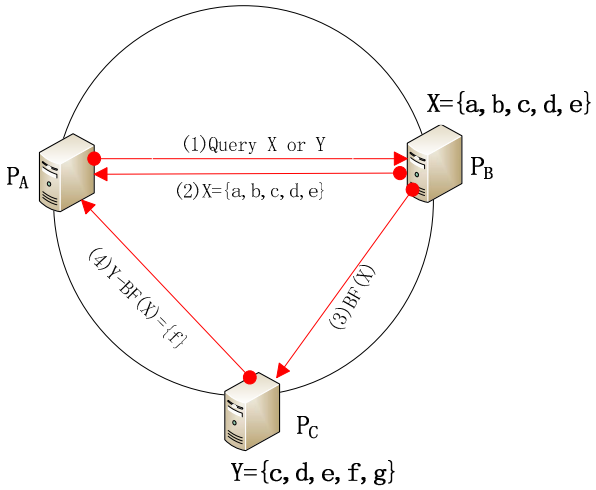

Figure 8. "Or query" based on BF.

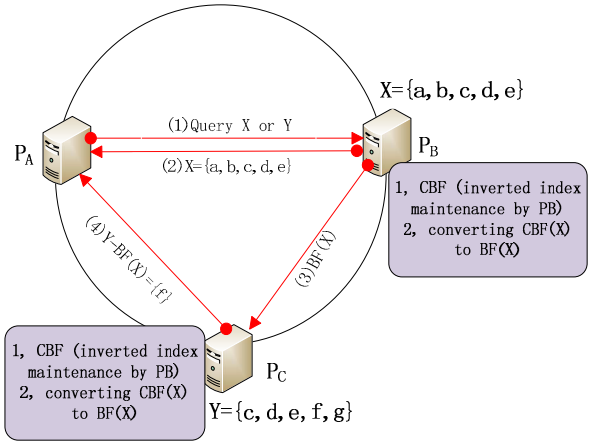

Figure 9. "Or query" based on CBF.

As illustrated in fig. 9, "or query" based on counting bloom filters. The process is as follows:

1) P_A launching an "or query", inquires X and Y, it founds the index of keywords X maintained by P_B, P_A sends query to P_B;
2) P_B sends the keywords stored on the locality to P_A;

3) Meanwhile, the index table of keyword X maintained by P_B storing by counting bloom filter, P_B converts CBF(X) to BF(X);
4) P_B sends BF(X) to the node of P_C, which maintains the index of keyword Y;
5) P_C gets the results Y-BF(X) by minus operation and returns it to P_A.
6) P_A merges the results return by P_B and P_C as the final answers to the query of "X or Y".

It is worth mentioning that, in the fifth step, there are certain false positives of bloom filter, so the minus operation of bloom filter will lead a little omission of the elements but has little effect on the results.

4. Experiment and Analysis

4.1 Experiment Settings

In order to express the dynamic network environment, in this paper, we defines churn as equation (1) shows, which "x" represent the number of joins or leaves of nodes, and "y" represent the total number of nodes in the network.

$$churn = \frac{x}{y} \tag{1}$$

1.Underlying P2P structure: In the PEERSIM system, we simulate chord to support multi-keyword search based on the inverted index. The network initializes 5000 nodes in all, in order to create a dynamic network environment, we supposes churn in the range: 0.2 to 0.7, the number of nodes in network keeps between 3000 and 7000.

2.Dataset: we adopts DBLP data set, the related information of a paper such as the author, the references and so on are regarded as a document, after being handled, the file DBLP.XML is divided into 369879 documents; then, these documents are stored on each node randomly, meanwhile, creates inverted index for them (inverted index established by lucene), finally, these indexes are mapped to the bloom filter or counting bloom filter. Table.1 summarizes the statistics for the test dataset.

TABLE I. STATISTICS OF THE DBLP DATA SET

Parameters	Value
Number of documents	369879
Number of collections	5000
DBLP topics	30000
Average number of document of a collection	75
Average size of documents	1KB

4.2 Recall Rate And Query Latency

We assumes "or query" composed by two keywords, fig. 10 plots the comparison of recall rate of based on bloom filter and counting bloom filter. As we can see from the graph, the recall rate based on counting bloom filter is higher than which based on bloom filter, what's more, with the increase of the churn of P2P network, this superiority is more evident. "and query" has similar conclusions.

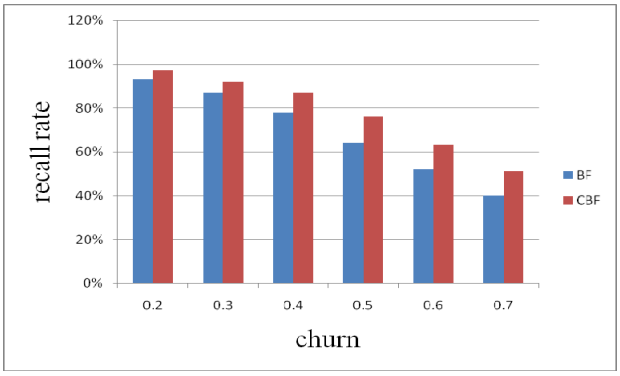

Figure 10. Recall rate of "or query" (BF VS CBF).

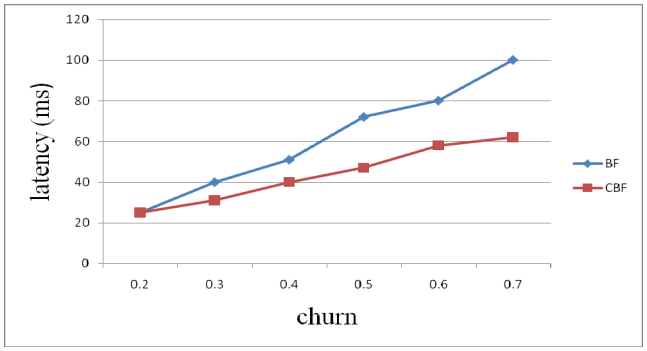

Figure 11. Latency of "or query"(BF VS CBF)

Fig.11 shows the comparison of query latency of based on bloom filter and based on counting bloom filter. As the fig.11 shows, the curve of counting bloom filter lies above the curve of bloom filter, and with the increase in churn, the curve of counting bloom filter more and more deviated from the curve of bloom filter, this shows that the latency based on counting bloom filter is lower than which based on bloom filter.

The specific reason is that in the circumstances of large churn, while "or query" based on bloom filter needs to re-constructed bloom filter to adapt the changes of documents caused by churn frequently, "or query" based on counting bloom filter only needs deletion operation or insertion operation of counting bloom filter, so, it can adapt to the changes of documents caused by churn. Re-construction is time-consuming, thus, as churn increases, the gap between the latency will increase.

5. Conclusion

In our paper, a multi-keyword searching mechanism based on counting bloom filter for structured P2P networks is proposed. Comprehensive simulations based on PEERSIM are conducted, results show that our design raise the recall rate of existing approach by 5%~10%, and significantly reduces the query latency of existing approach by 10ms~40ms, when the nodes join and leave from the P2P networks frequently. Therefore, the mechanism we proposed is a kind of fast and storage efficient of muti-keyword searching mechanism.

6. Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grant No. 61173167 and the key R&D project 2011_LH_17 of China Mobile Group Hunan Company Limited.

7. References

[1] M. Ripeanu, Peer-to-Peer Architecture Case Study: Gnutella Network Technical Report TR-2001-26, Univ. of Chicago, Dept. of Computer Science, July 2001.

[2] [KaZaA] http://www.kazaa.com.

[3] S.Ratnasamy, P.Francis, M.Handley, et al. A Scalable ContentAddressable Network. Proc. ACM SIGCOMM 2001, Aug. 2001.

[4] I. Stoica, R. Morris, D. Karger, , et al. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. Proc. ACM SIGCOMM 2001,Aug. 2001.

[5] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-pee systems. Proc. Middleware 2001, Nov. 2001.

[6] B.Zhao, J.Kubiatowicz, and AJoseph, et al . Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing. Technical Report, UCB/CSD-OI-1141, April 2000.

[7] B.Bhattacharjee, S.Chawathe, V.Gopalakrishnan, etal. Efficient Peer-To-Peer Searches Using Result-Caching. IPTPS 2003.

[8] P.Reynolds and A.Vahdat. Efficient peer-to-peer keyword searching.Technical Report 2002, Duke University, CS Department, February 2002.

[9] B.H. Bloom. Space/time trade-offs in hash coding with allowable errors.Communications.of.the.ACM,13(7):422-426, 1970.

[10] Fan, L., Cao, P., Almeida, J., and Broder, A.Z., Summary Cache:A Scalable Wide-area Web Cache Sharing Protocol,IEEE/ACM Transactions on Networking, 8 (3):282-293, 2000.

[11] Zhu, Y. and Jiang, H., False Rate Analysis of Bloom Filter Repoicas in Distributed Systems, in Proceedings of ICPP, 2006.

[12] K.Yang, and J. Ho. Proof: a DHT-based peer-to-peer search engine. WI 2006, Hong Kong, pp. 702-708.

[13] Mo Hai and Shuhang Guo. An Efficient Multi-Keyword Search Mechanism in Structured P2P Networks. Progress in Informatics and Computing (PIC), 2010 IEEE International Conference on.

[14] H. Chen, HJin, J.Wang, et al.Efficient multi-keyword search over P2P web. WWW 2008. / Alternate Track: WWW in China - Chinese Web Innovations April 21-25,2008

[15] Kun xie, Jigang Wen, Dafang Zhang, Gaogang Xie Bloom Filter Query Algorithm Journal of Software, Vol.20, No.1, January 2009, pp.96－108 ISSN 1000-9825, CODEN RUXUEW