Combined Query Technique For Optimization Of Bloom Filter

N.Bagavathi
Computer Science and Engineering, PRIST University, Trichy.
D/oR.Nagarajan,perumal kovil street,veppanthattai(po)(tk),perambalur(dt),tamilnad,India.
Bagavathi14@gmail.com.

Abstract-- **Peer-to-Peer multikeyword searching requires distributed intersection/union operations across wide area networks, raising a large amount of traffic cost. Existing schemes commonly utilize Bloom Filters (BFs) encoding to effectively reduce the traffic cost during the intersection/union operations. In this paper, we address the problem of optimizing the settings of a BF. We show, through mathematical proof, that the optimal setting of BF in terms of traffic cost is determined by the statistical information of the involved inverted lists, not the minimized false positive rate as claimed by previous studies. Through numerical analysis, we demonstrate how to obtain optimal settings. To better evaluate the performance of this design, we conduct comprehensive simulations on TREC WT10G test collection and query logs of a major commercial web search engine. Results show that our design significantly reduces the search traffic and latency of the existing approaches**.

keywords—**Bloom filter, DHT, multikeyword search, P2P.**

I.INTRODUCTION

With the emergence of peer-to-peer (P2P) file sharing applications, millions of users have used P2P systems to search desired data. A P2P network has also shown a great potential to become a popular network tool for sharing information on the web , where information resides on millions of sites in a distributed manner. P2Pbased systems have the ability to leave the shared, but distributed, data at their origins instead of collecting and maintaining them in a centralized repository.Existing P2P retrieval mechanisms provide a scalable distributed hash table (DHT) [3] that allows every individual keyword to be mapped to a set of documents/nodes across the network that contain the keyword. Using this singlekeyword-based index, a list of entries for each keyword in a query can be retrieved by using existing DHT lookups.For multikeyword search, the simple solution which merges the results of each keyword search incurs a lot of traffic. Given an example, considering a two-keyword query "peer-to-peer network," the query is decomposed into "peer-to-peer" and "network" and then the two keywords are searched separately with a consequent intersection operation. A potentially large amount of data traffic will be raised across the wide area network.

It is well known that Bloom Filter (BF) is an effective way to reduce such communication cost .A BF is a lossy but succinct and efficient data structure to represent a set S, which can efficiently process the membership query such as "is the element x in the set S." By transmitting the encoded sets instead of raw sets among peers, the communication cost can be effectively saved. Applying BF is not difficult, but how to get optimal results in terms of minimum communication cost is not trivial.In this work, we show mathematically that the optimal setting of BF in terms of communication cost is determined by the global statistical information of the involved items on both sides, not the minimized false positive rate as claimed by the previous studies . We further demonstrate how to get optimal settings through numerical analysis. Moreover,we find that the intersection order between sets is indeed important for multikeyword search; thus, we design optimal order strategies based on BF for both queries with "AND" and "OR" operators.We conduct comprehensive trace-driven simulations on Text Retrieval Conference (TREC) WT10G test collection and the query logs of a major commercial web search engine to evaluate the performance of this design. Results show that our design significantly reduces the search traffic and latency of the existing approach, respectively.

To summarize, we made the following main contributions in this work:

1. We show mathematically that the optimal setting of BF in terms of traffic cost is determined by the numbers of items involved on both sides when using a BF.
2. We derive an effective approach to achieve BF optimal settings through numerical analysis.

II .RELATED WORK

There are generally two types of decentralized P2P search engines:federated search engines over unstructured P2P networks and DHT-based distributed global inverted index on top of structured P2P networks.

In federated search engines, peers that maintain indexes of their local documents are organized in an ad hoc fashion.A simple search method is flooding. Each query is tagged with a maximum Time-To-Live (TTL) to limit the number of hops it travels. In order to reduce the search cost, many approaches have focused on the issue of query routing.

The proposed algorithms often need to perform the query search in two levels, the peer level and document level. First, a group of peers with potential answers are detected. Second, the query is transferred to the selected most relevant peers to evaluate the query against their local indexes and return the matched answers. Finally, the retrieved answers are merged to produce a single answer set for the user. In PlanetP , each peer replicates a global term-to-peer inverted index which contains a mapping "t ! p" if term t is in the local index of peer p. For each query, it ranks the peers using an IDF-like relevance model.

In practice, it is difficult for every node to maintain such global index information in a large-scale network.The search performance of a federated search engine can be further improved using superpeer-based P2P architectures , which consider the inherent heterogeneity of peers. In this kind of architecture, peers with more memory,processing power, and network connection capacity provide distributed directory services for efficient and effective resource location. Thus, the peers that are limited in these resources won't become bottlenecks in the network. In ,Shen et al. proposed a hierarchical summary indexing framework for efficient content search in superpeer P2P networks. In their scheme, the documents are summarized in three levels: document level, peer level, and super peer level. For each level, the summarization process consists of two steps. In the first step, they use VSM to represent the information of this level as vectors, where each component of the vector corresponds to the importance of a term. The vectors are then further reduced from a large-dimensional space to a much smaller one using SVD to facilitate indexing. For searching, they extend the existing highdimensional indexing technique—Vector Approximation file (VA-file) to perform efficient K Nearest Neighbor (KNN) searching for text documents.Another approach to improve the search performance of the federated search design takes advantage of the enhanced properties of its network topology. Zhang and colleagues identify the natural principle in Gnutella and other P2P networks called interest-based locality,which reveals that a P2P user often stores similar items.Peers with similar interest are linked together. By forwarding the queries through the interest-based shortcuts, their scheme reduces a significant amount of unnecessary flooding. In Bibster , peers advertise their expertise, which contains a set of topics that the peer is an expert in. Other peers may accept these advertisements or not, thus creating a semantic link to their neighbors. These semantic links form a semantic overlay for intelligent query routing. A language model-based method is designed by Lu and Callan to locally rank the neighboring peers. Queries are forwarded to the top-ranked neighbors who are most likely to have the answers. An SSW overlay network dynamically clusters peers with semantically similar data closer to each other and maps these clusters in a highdimensional semantic space into a one-dimensional smallworld network that has an attractive trade-off between search path length and maintenance costs.DHT-based searching engines are based on distributed indexes that partition a logically global inverted index in a physically distributed manner. Currently, there are two kinds of distributed index mechanisms: single-term-based inverted indexes and term-set-based indexes.Searching with a single-term-based distributed index can retrieve the list of documents/nodes for each keyword in a query. In , frequent terms of a document are selected to be published into the global index. When such a keyword is published, the list of other terms in the document is replicated with the identifier of the document in the posting list. Multikeyword search is performed by first locating the position of the DHT node which is responsible for a given keyword and then performing a local search in the posting list for other keywords. Finally, the list of documents that contain all the keywords is returned as the results. Little is known about the performance of the full text search using selected keyword publishing, because a few selected frequent terms may not be representative for a document and such replication strategy may incur unacceptable storage and communication cost. Another scheme performs a distributed intersection operation for multikeyword search. Based on the global single-term-based inverted index built on DHT, the multikeyword search looks up the sets for different keywords from multiple peers across the wide area network and returns the intersection. Although only a few nodes need to be contacted, each node has to send a potentially large amount of data across the wide area network. Zhang and Suel propose to use multiple rounds BFs to reduce the cost for large-scale textual collections. By transmitting the

Fs of sets instead of raw sets among peers,block by block, the BF-based schemes effectively reduced the communication cost. Reynolds and Vahdat claim that optimal BF settings can be achieved through minimizing the false positive of a BF. In this paper, we show that minimizing the false positive of a BF is far from optimal settings.Another strategy to reduce the communication cost is to precompute the index using term-set indexing techniques.Some preliminary experiments in have shown that the term-set-based indexing is promising for multikeyword search across the wide area network. However, the major drawback of term-set-based index is the exponential index size. To reduce the unacceptable index size, Podnar et al. proposed to index only highly discriminative keyword (HDK) combinations in a distributed global index.Although their method can reduce the total number of combinations in the global index, a large number of keys generated by HDK schemes are never or rarely used in queries, causing substantial consumption of both resources of bandwidth and storage. In , the HDK indexing schemes are extended by taking into account the popularity of term combinations appearing in user queries. The querydriven indexing (QDI) strategy uses query statistics to filter out superfluous keys. However, the scheme relies on a very large query log, which probably is unavailable for nodes in a P2P system. Bender et al. proposed to index keyword sets in queries issued by P2P users. In their design, a DHT node responsible for keyword x stores additional posting lists for term sets including x that are frequently searched together in previous query logs. Although such a term-set indexing scheme reduces the scale of indexes, it suffers from the problem of cold start because a distributed intersection operation is also required if a query has not been searched before.

III. System Design

In this section, we first give a brief overview of our hybrid P2P network design for P2P multikeyword search, and focus on how to optimize the communication cost of DHTbased multikeyword search using an optimal BF. We then describe the optimization strategies for "AND" queries and"OR" queries. In propose an optimized intersection-order strategy for multikeyword queries. It present the pushing synopsis gossip algorithm for collecting global statistical information.

A.Solution Outline

In this design, a hybrid P2P network [21] is a combination of 1) an unstructured P2P network which can use a gossiping algorithm to gather global statistical information, and 2) a BF enabled overlay based on DHT global inverted indexes.

Each peer participates in an unstructured network and acts as a structured DHT node as well.

B. Minimizing Communication Cost for Multikeyword Search

Before we discuss the mechanism for reducing the communication cost for multikeyword search, we introduce the following concepts:

Observations on user behaviors. We recently analyzed four months query logs of a major commercial web search engine. The query length distribution ,from which we can observe that 56 percent of the queries consist of at least two terms. This indicates that multikeyword search is quite common in web content searching.

Bloom filter. We review the basic of BF, following the framework of references [4]. A BF is essentially a bit vector bitvec m with m bits, initially all set to 0, that facilitates membership test to a finite set S ¼ fx1; x2; . . . ; xng of n elements from a universe U. It uses a set of k uniform and independent hash functions fh1; h2; . . . ; hkg to map the universe U to the bit address space [1 _m]. For each element x belonging to S, the bits hi(x) are set to 1 for

Table 1
Notations Used in the Algorithm

Notation	Description
n	Number of elements inserted into a BF
m	Size of the bit vector used as a BF
k	Number of hash functions used for a BF
f	False positive rate of a BF
f_{min}	Minimized false positive rate of a BF
X	The set of the IDs of the documents that containing keyword x
$BF(X)$	The BF for set X
$Y \cap BF(X)$	The estimated intersection of sets X and Y based on BF(X) and Y
r	Number of bits each item in the posting list takes

1 _ i _ k. To check whether an item y is in S or not, we check whether all hi(y) are set to 1. If not, y clearly is not a member of S. If all hi(y) are set to 1, we assume that y is in S.

Before we introduce our algorithm for multikeyword search, we list the notations used in our algorithm in Table 1.

C.AND Query

A common solution for a multikeyword search needs conducting a distributed intersection operation in a wide area network. It gives an example of a two-keyword (x, y) search. The query is first routed to the DHT node which is responsible for keyword x. Then, X, the set of identifiers of documents that contain keyword x, is transmitted to the node which is

responsible for keyword y for a consequent intersection operation to achieve X \ Y ,where Y is the set of document identifiers whose corresponding documents contain keyword y. The final results are returned to the client.

D. OR 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 multikeyword query to the users. It presents an example of the straightforward strategy for a two-keyword "OR" query. At the beginning, the query is separately sent to the DHT nodes responsible for keywords x and y, respectively. Then, the DHT nodes separately send back the complete list for each keyword. At last, the results of both keywords are merged at the client. Thus, the total communication cost is ðjXj þ jY jÞr.

E. Intersection-Order Optimization Strategy

For a query with more than two keywords, it is intuitive that there is much benefit if we first perform distributed intersection operations for the pairs of keywords that have smaller size of intersection. However, it is difficult to estimate the size of intersection incurred by two keywords before we get the exact intersection. In our design, we use BF to estimate the size of intersection between two sets for any given two keywords.

F. Gathering Global Keyword Popularity

Within the structure of a hybrid P2P network, we use a variant of the push-synopsis gossip algorithm first proposed in to gather global keyword popularity in the web. The robust algorithm enables every peer to quickly collect the global statistical term frequency in the P2P web.

IV.PERFORMANCE EVALUATION

In this section, we first introduce the metrics that we use in the evaluation. Then, we analyze how to achieve BF optimal settings through numerical method with Matlab. Based on the analysis results, we conduct comprehensive simulation to compare our design with the work proposed .

A.Metrics

In the evaluation, we mainly consider two metrics, traffic and search latency.

B. Traffic

P2P traffic has a significant impact on the underlying network. Heavy network traffic limits the scalability of P2P networks. We define the traffic as network resource used in the search process, which is mainly a function of the length of the links, the bandwidths, and other related expenses .

Specifically, in the P2P network, when a message is transferred over the application layer overlay network from a peer to another, the message actually traverses a path consisting of a set of underlying physical links. The cost caused by this single hop over the P2P overlay is calculated by adding up the cost of the underlying links: Tc ¼ M Pi Li=Bi, where M is the size of the message and Li and Bi, respectively, represent the length and the bandwidth of the ith link in the physical layer the message traverses. Generally, a query processing message travels multiple hops across the application layer overlay network.Thus, the traffic is the summed cost of all the hops.In the experiment, we use BRITE to simulate the underlying physical network across which the packets aretransferred. The parameters of the physical links simulated with BRITE include the euclidean length, the bandwidth,and other configurations of the physical links. The traffic cost values are eventually computed according to the sizes of the transferred data and the related parameters of the underlying physical links.

C. Optimal Setting of Bloom Filter

In this section, we show how to achieve the minimized communication cost using optimal settings of BFs. We analyze the communication cost quantified by with Matlab. We consider three typical situations. We set r to 250 bits based on the research results conducted on Google search engine, which show that the average URL length measured in character is 31.2 characters We adjust the parameters m and k and examine how the value of fðm; kÞ changes.We find that the intersection order is critical for minimizing the communication cost. It can observe that when jXj is larger than jY j, BF does not work for minimizing the communication cost at all. The optimal strategy is setting m to 0 that means transmitting no BF but fetching the original set Y to the DHT node that is responsible for keyword x.

In the simulation, we vary the value of k from 1 to 10. It shows that when the value of k varies from 1 to 5 with fixed value of m, the communication cost decreases; while the communication cost changes very slightly when k varies from 5 to 10. The minimized value appears when k ¼ 8. Thus in the experiments, we use a set of eight hash functions. On

the contrary, the value of fðm; kÞ is significantly influenced by the variable m. The minimal value of fðk;mÞ can be achieved when m is set as an optimal value.The results demonstrate that the optimal BF is determined by the popularities of keywords and the intersection order. Much benefit can be achieved if we transfer the BF for the set of a less popular keyword to the DHT node responsible for a popular keyword during the process of distributed intersection. Based on these observations, given jXj, jY j, the objective of our optimal BF-based intersection algorithm is to enable each node intelligently choose the optimal m and the intersection order to achieve the minimal communication cost.

V.Conclusions

In this paper, we show mathematically that the optimal setting of BF in terms of traffic cost is determined by the numbers of items involved on both sides. We derive an effective approach to achieve BF optimal settings through numerical analysis. We also proposed the optimal order strategies for both "AND" and "OR" queries. We conduct comprehensive simulations based on TREC WT10G test collection and the query logs of a commercial web search engine. Simulation results show that our design outperforms existing work. In the future work, we will try to examine the performance of more comprehensive solutions by using larger scale data collections.

REFERENCE

[1]T. Suel, C. Mathur, J. wen Wu, J. Zhang, A. Delis, M. Kharrazi, X.Long, and K. Shanmugasundaram, "Odissea: A Peer-to-Peer Architecture for Scalable Web Search and Information Retrieval,"Proc. Int'l Workshop Web and Databases (WebDB), 2003.

[2] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan,"Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM SIGCOMM, 2001.

[3] B.H. Bloom, "Space/Time Trade-Offs in Hash Coding with Allowable Errors," Comm. ACM, vol. 13, no. 7, pp. 422-426, 1971.

[4] P. Reynolds and A. Vahdat, "Efficient Peer-to-Peer Keyword Searching," Proc. Int'l Conf. Distributed Systems Platforms and Open Distributed Processing (Middleware), 2003.

[5] J. Zhang and T. Suel, "Efficient Query Evaluation on Large Textual Collections in a Peer-to-Peer Environment," Proc. IEEE Int'l Conf. Peer-to-Peer Computing (P2P), 2005.