International Journal of Power Control Signal and Computation(IJPCSC)
Vol3. No1. Jan-Mar 2012 ISSN: 0976-268X
www.ijcns.com

Query-driven Adaptive Term Set Search in large Peer-to-peer Textual Collections

V.Deepa[1], PG Scholar, Karpagam University,erdeepasri22@gmail.com
P.S. Balamurugan[2] , Head CSE, Karpagam University,balabeme@gmail.com

Abstract:

Most of the search mechanisms which include in Distributed Hash Table based Peer-to-peer system depends on multiple single keyword-based search operations. This increases the traffic cost and has a poor accuracy. Pre-computing the term-set-based index can reduce the cost but needs exponentially growing index size. Based on the observations made, queries are usually short and the users have limited interests. We propose a Novel Index Pruning method namely Query-driven TSS. Here, we index the term combinations which are frequently used in user queries and non-redundant with respect to the rest of the index. By publishing the most relevant term sets from documents on the peers, Query-driven TSS provides considerably a better performance. The search performance possesses a centralized solution and the index size is reduced from an exponential scale to the scale of O(nlog(n)). The results obtained prove the achievement of a scalable peer-to-peer text retrieval for very large document collections and deliver good retrieval performance.

1. INTRODUCTION

Peer-to-peer file sharing technologies have proved a great deal in improving the potential of network information sharing in the internet due to various reasons. One such reason is that information on the Internet exists in millions of Web sites and desktop storage media. Peer-to-Peer-based search has the ability to leave the shared, but distributed, data at their origin, rather than gathering and keeping them in a centralized repository. Moreover, there are a lot of significant performance, scalability and availability benefits by distributing the indexing and querying load over large networks of collaborating peers. The peer-to-peer search is more robust than centralized search, because the decease of a single server may make the entire search process futile. The peer-to-peer networks can provide more storage capacity with millions of nodes.

The existing Peer-to-peer systems often rely on the Distributed Hash Tables (DHT) to build distributed indices. A Distributed Hash Table is a class of distributed systems that provides lookup service similar to hash tables where the mapping from keys to values is performed by a large number of cooperative distributed nodes. By using Distributed Hash Tables, it is possible to build the index which maps an individual keyword to the global documents containing the keyword across the network. Using this single-keyword-based index, the list of entries for each keyword in a search query can be retrieved. Multi-keyword search is conducted by merging all the lists. One effective way of reducing the search cost is to pre-compute the index using term set indexing, which maps a term set to a set of documents that contain the multiple terms. Such an approach, however, is not widely adopted because it often incurs exponentially growing index size. Even if the length of the combination is fixed and only very few meta-data words are indexed, the total number of entries is far more than that of the standard single-term-based index. In this paper, we focus on the scalability problem of term set indexing in large-scale P2P search engines. By analyzing large traces collected from a major commercial search engine, we have the following observations: 1) queries are typically short and 2) users always have limited interests. Based on the observations, we propose a novel index pruning method for distributed term set index, called TSS. TSS utilizes the Term Frequency-Inverse Document Frequency (TFxIDF) model to rank the relevance between a term set and the document from which the term-set is extracted. It then selects to publish the top relevant term sets into the global index atop a novel distributed multidimensional hashing table. TSS utilizes a pushing synopsis-based gossip algorithm to collect the global statistical Inverse Document Frequency (IDF) information of terms in the P2P network. By hybridizing the unstructured protocol with the global index, the TSS term set index achieves multi-keyword searching and TFxIDF based ranking in large-scale P2P networks. We simulate the hybrid P2P network and evaluate the performance of our method on the TREC WT10G test collection and the query logs of a major commercial search engine [1]. Results show that TSS design significantly reduces the search cost as well as provides comparable search performance and ranking accuracy with state-of-the-art centralized search engines at the index cost of O(nlog(n)).

2. RELATED WORKS

Existing P2P search engines can generally be classified into two types: 1) federated engines using unstructured P2P networks and 2) distributed global inverted index in structured P2P networks. In the first type, peers which maintain indexes of their local documents are organized in an ad hoc fashion. A basic search method is flooding. To reduce the search cost of flooding-based schemes, many approaches focus on the issue of query routing. The proposed algorithms commonly search a query at two steps: the peer step and document step. First, a group of peers with potential answers to the query are detected. Second, the query is submitted to the identified most relevant peers to return answers from their local indexes. Lu and Callan [2] propose to use language model to locally rank the neighboring peers and forward the queries to the top-ranked neighbors which are most likely to have the answers. The enhanced properties of the network topology are extensively used to improve the performance of the federated search engines. Li et al. [3] propose the SSW scheme which dynamically clusters peers with semantically similar data closer to each other and maps these clusters in a high-dimensional semantic space into a one-dimensional small-world network that has an attractive trade-off between search path length and maintenance costs. By considering the inherent heterogeneity of peers, super peer-based P2P architectures can further improved the search performance of a federated search engines. 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 will not become bottlenecks in the network. In [4], Shen et al. proposed a hierarchical summary indexing framework in which the content is summarized in different levels: document level, peer level, and super-peer level.

Another effective approach to improve the search performance of a federated search engines is the replication strategy. By replicating items and queries properly across the network, such strategies can effectively improve the search successful rate while avoiding exhaustively flooding the unstructured P2P networks. Existing replication strategies in unstructured federated P2P search networks can be divided into two categories: the query-popularity-aware replication approach [5] and the query-popularity-independent replication strategy [6]. 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. Existing DHTs can naturally support mapping every individual term to a set of documents across the network that contain the term. Using this single-term-based index, a list of entries/documents for a given keyword in a query can be retrieved by using existing DHT lookups. In [7], 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. Multi-keyword search is performed by first locating the position of the DHT node which is responsible for one given keyword, and then, performing a local search in the posting list for the other left keywords in the query. Finally, the list of documents that contain all the keywords is returned as the results.

An effective multi-keyword searching scheme looks up the sets of documents for separate keywords in the query from multiple DHT nodes and returns the intersection. Although only a few nodes need to be contacted, each has to send a potentially large amount of data across the wide-area network, making the distributed intersection operations bandwidth costly. Another way to reduce the bandwidth cost is to pre-compute term set index that maps a set of terms to the list of global documents containing them. By avoiding the distributed intersection operations across the wide-area network, a term set index is potentially effective to reduce the communication cost [8].

3. PROPOSED SOLUTION

In this paper, we are proposing a Query-Driven TSS scheme for multi-term search in P2Ps in which an index pruning algorithm is introduced to reduce bandwidth consumption. Initially we are designing the framework for the Query-Driven TSS and then we present our enhanced approach of query-driven term set search. In this scheme we are processing the searching mechanism by means of the query-driven term set search. In this section, we first introduce the data set and query logs we use for the evaluation of the Query-Driven TSS performance. To evaluate the search bandwidth cost, we compare the average number of postings transferred per TSS query during the query search process with that by using the scheme of single-term-based index. During searching, the scheme using single-term-based index transfers the posting lists for the query keywords among DHT nodes to achieve the intersection, while the Query-Driven TSS looks up the multidimensional index and returns all the matched results. We further compare the communication cost of TSS with those of the Bloom filter techniques.

A. Constructing a term set based indexing

Initial step of this proposed approach is to construct a term set based indexing. Given a set of terms and a full list of global documents which contain the words in the term set, the Distributed Term-set-based Inverted Index (DTII) can map the term set to the list of documents. In contrast to the traditional Distributed Single-term-based Indexing (DSII), DTII takes much lower communication cost when performing a multi-term query. However, the scale of the naive DTII is considerably larger than DSII due to the fact that there are much more term sets than individual terms. Specifically, for a given document with n distinct terms, the set of all possible combined terms has $\sum_{i=1}^{n}\left(\frac{n}{i}\right) = 2^n$ (1) elements in the worst case. To address this issue, we propose a pruning method to reduce the index size of DTII as well as preserve the search performance.

B. Index Pruning

To further reduce the index size, we propose a pruning method based on the TFxIDF ranking model. This concept is motivated by another observation that when users search the Web, they often prefer a small number of results with top relevance rank values. For example, user behavior of search engines has been studied by iProspect [2], which shows that about 88 percent users change their searches after reviewing the first three pages.

C. Construction of Term Set Search Scheme

In this module we are presenting the term set search scheme for the problem. In TSS, a query is first inserted into the structured overlay. The Chord protocol routes the query to the responsible DHT node, which then ranks the list of documents for the query q using TFxIDF ranking scheme:

$$sim(q,d) = \frac{\sum_{t \in q}(1+\log(f_{d,t}))\times\log(1+\frac{N}{f_t})}{\sqrt{|q|\times|d|}} \quad (2)$$

where |q| is the length of the query and returns the top-k results. During search, the lookup requires average O(log(n)) messages, where n is the number of peers in the network. As we have seen that TSS can efficiently handle queries which have no more than l_{max} keywords. For the query which has more keywords, TSS first ranks the keywords according to their global frequency in an ascending order, and then, retrieves the set of URLs of the top-ranked documents which contain the first l_{max} terms. During retrieval, TSS piggybacks the ranking value with the other terms onto the retrieval message for a local search for the other terms. Only the documents with stable ranking values are returned after the local search. TSS multidimensional index provides a cost-effective solution for multi-keyword search, because it avoids the costly distributed intersection operations in wide-area networks performed by existing single-term index schemes. Note that a ranking scheme is critical for a P2P text retrieval system. Without ranking schemes, a complete list of results can possibly raise an unacceptably amount of communication cost roughly proportional to the size of the network, making the searching scheme un-scalable.

D. Construction of Query-driven Term Set Search Scheme

To compensate for the loss of information caused by the truncation, we extend the set of indexing features with carefully chosen term sets. Indexing term sets are selected based on the query statistics extracted from query logs, thus we index only such combinations that are a) frequently present in user queries and b) non-redundant w.r.t the rest of the index. The distributed index is compact and efficient as it constantly evolves adapting to the current query popularity distribution. Moreover, it is possible to control the tradeoff between the storage/bandwidth requirements and the quality of query answering by tuning the indexing parameters. Initially, the peers collaboratively build a distributed single-term index associating all single term keys from the global collection with their top-k global document references. Each peer maintains a part of the global index containing a set of keys assigned by the hashing mechanism used in the underlying Distributed Hash Table (DHT).

Algorithm

- For each node in the query lattice, request the posting list from the peer responsible for term combination.
- If the term combination is indexed then the associated posting list is sent back and the part of query lattice is excluded.
- Else is updated
- Produce the union of originating peer, re-ranks all records w.r.t the original query and presents the top- ranked results to the user.
- *On-demand indexing* mechanism is executed when a certain key is detected.
- The peer responsible for this key sends an indexing request to all the peers holding documents that contain the corresponding term combination and acquires a new posting list that stores a limited number of top-ranked document references.

- Now the key get *indexed* and can be used for future query processing.
- The list of peers to contact is maintained in a dedicated structure distributed in the network in order to avoid broadcasting of the indexing request.
- Finally remove the obsolete keys.

4. PERFORMANCE EVALUATION

In this module we are evaluating the proposed approach with the existing approaches with the parameter metrics such as the precision and recall. The results show that the proposed system exhibits a better performance than the existing approaches.

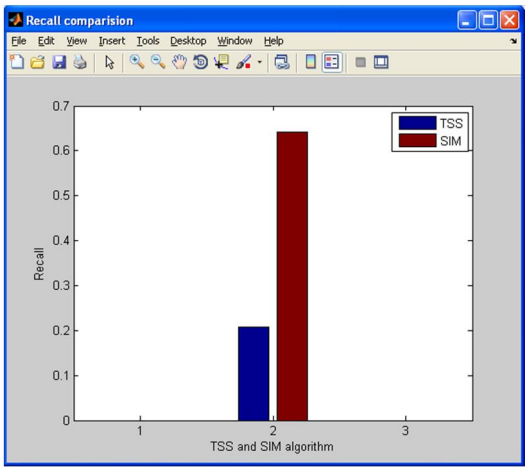

Fig. 1 Recall Comparison (TSS vs SIM)

Fig.1 describes the performance variation between the TSS approach and the SIM approach. The less the recall is, the more is the performance of the searching technique. From the experimental results, it is clearly visualized that the TSS is more efficient compared to the SIM approach. Also, it is evident that the TSS approach is far better than the SIM approach.

The precision is another important metric using which the evaluation of searching techniques is made. Therefore, another experimentation to compare the performance of the TSS and SIM approaches based on the precision is made. The less is the precision, the more is the performance of the searching technique. From the results obtained, it could be found that the precision indices of the TSS are less compared to that of SIM. Hence, it is proved that the performance of TSS is considerably more compared to that of SIM.

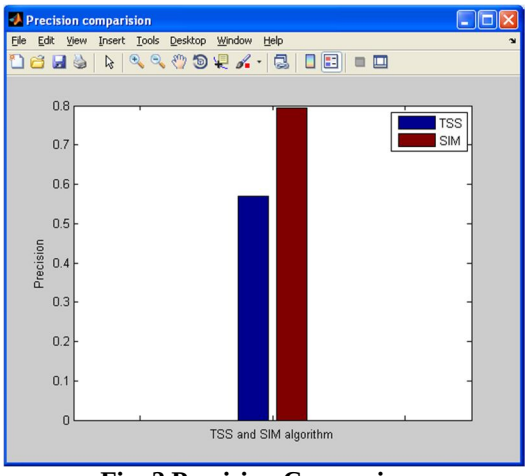

Fig. 2 Precision Comparison

5. CONCLUSION

In this paper, we propose the TSS scheme for multi-term search in Peer-to-peer in which an index pruning algorithm is introduced to reduce the bandwidth consumption. Moreover, we have shown results which prove that the TSS is more efficient than the SIM approach.

REFERENCES

[1] J.R. Wen, J.Y. Nie, and H.J. Zhang, "Query Clustering Using User Logs," ACM Trans. Information Systems, vol. 20, no. 1, pp. 59-81, 2002.
[2] J. Lu and J. Callan, "Content-Based Retrieval in Hybrid Peer-to- Peer Networks," Proc. Int'l Conf. Information and Knowledge Management (CIKM), 2003.
[3] M. Li, W.-C. Lee, A. Sivasubramaniam, and J. Zhao, "SSW: A Small World Based Overlay for Peer-to-Peer Search," IEEE Trans. Distributed and Parallel Systems, vol. 19, no. 6, pp. 735-749, June 2008.
[4] H.T. Shen, Y.F. Shu, and B. Yu, "Efficient Semantic-Based Content Search in P2P Network," IEEE Trans. Knowledge and Data Eng., vol. 16, no. 7, pp. 813-826, July 2004.
[5] E. Cohen and S. Shenker, "Replication Strategies in Unstructured Peer-to-peer Networks," Proc. ACM SIGCOMM, 2002.
[6] X. Luo, Z. Qin, J. Han, and H. Chen, "DHT-Assisted Probabilistic and Exhaustive Search in Unstructured P2P Networks," Proc. IEEE Int'l Symp. Parallel and Distributed Processing (IPDPS), 2008.
[7] C. Tang and S. Dwarkadas, "Hybrid Global-Local Indexing for Efficient Peer-to-Peer Information Retrieval," Proc. USENIX Symp. Networked Systems Design and Implementation (NSDI), 2004.
[8] O.D. Gnawali, "A Keyword-Set Search System for Peer-to-Peer Networks," Ma.