

Full-Text Retrieval in Unstructured P2P Networks using Bloom Cast Efficiently

S. V. Uthayasri¹ Mr. R. PremKumar²

¹M.E-Computer Science & Engineering ²Assistant Professor

^{1,2}Department of Computer Science

^{1,2} AMS Engineering College, Namakkal, India

Abstract— Efficient and effective full-text retrieval in unstructured peer-to-peer networks remains a challenge in the research community. First, it is difficult, if not impossible, for unstructured P2P systems to effectively locate items with guaranteed recall. Second, existing schemes to improve search success rate often rely on replicating a large number of item replicas across the wide area network, incurring a large amount of communication and storage costs. In this paper, we propose BloomCast, an efficient and effective full-text retrieval scheme, in unstructured P2P networks. By leveraging a hybrid P2P protocol, BloomCast replicates the items uniformly at random across the P2P networks, achieving a guaranteed recall at a communication cost of $O(N)$, where N is the size of the network. Furthermore, by casting Bloom Filters instead of the raw documents across the network, BloomCast significantly reduces the communication and storage costs for replication. Results show that BloomCast achieves an average query recall, which outperforms the existing WP algorithm by 18 percent, while BloomCast greatly reduces the search latency for query processing by 57 percent.

Keywords: Peer-to-peer systems, Bloom Filter, replication

I. INTRODUCTION

Due to the exact match problem of DHTs, such schemes provide poor full-text search capacity. In federated search engines over unstructured P2Ps, queries are processed based on flooding. Unstructured P2Ps are commonly believed to be the best candidate for supporting full-text retrieval because the query evaluation operations can be handled at the nodes that store the relevant documents. However, search recall is not guaranteed with acceptable communication cost using a flooding-based scheme.

Replication strategies are extensively utilized to improve search performance in unstructured P2Ps. The existing replication strategies can be divided into two categories. The first type is the query popularity aware strategies [3]. Such strategies assume that the access frequencies of the items are known and the number of replicas is determined by the query's popularity. Cohen and Shenker [3] claimed that the square-root replication strategy, where the number of the replicas is proportional to the square-root of the query popularity/rate, has the optimal search performance. In query popularity aware replication strategies, the items with high query rate are highly replicated for future query searching, thus the search performance for popular items is improved. However, the strategy is inefficient for solving insoluble queries, the queries for rare items [3]. Moreover, in practice, the query

frequency is difficult or even impossible to obtain in a distributed P2P system.

The second type of replication strategy is independent of the popularity of the query, such as the WP scheme [4]. By replicating data and query replicas randomly across a P2P network regardless of the query rate of the data, such kind of schemes improve search recall of queries no matter they are popular or not. In WP scheme, the term query replica is used to differentiate a query message transferred across the network without performing and a query that evaluated in a node. A query replica will be performed by the node holding it. In [4], the WP scheme utilizes random walk technique to deploy replicas. The problem of random walk-based scheme is that it is not fault-tolerant. Another problem of the existing replication strategies is that simply replicating document reference or selected metadata cannot successfully support full text retrieval. To support full text retrieval, the existing replication strategies need to replicate the full document across the network, raising possibly unacceptable communication and storage costs.

II. MOTIVATION OF BLOOM CAST

BloomCast replicates Bloom Filters (BF) [5] of a document. 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 element x in set S ." By replicating the encoded term sets using BFs instead of raw documents among peers, the communication/storage costs are greatly reduced, while the full-text multikeyword searching are supported. We show the effectiveness and efficiency of Bloom Cast through mathematical proof and comprehensive simulations based on NIST TREC WT10G data collection and query logs from a major commercial search engine. Results show that Bloom Cast can achieve guaranteed recall with largely reduced search latency, significantly outperforming existing schemes. Results also show that for multikeyword searching, Bloom Filter encoding can greatly reduce the communication cost for data replication. Support single keyword search by retrieving the list of documents for a given keyword. Because of the utilization of the exact hashing techniques, the DHT-based schemes, however, fail to support complex queries with multiple keywords. Tang and Dwarkadas [6] proposed a hybrid index scheme, where the frequent terms of a document are selected to be published on a 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 for other

keywords in the posting list. Finally only 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 [7] and such replication strategy may incur unacceptable storage and communication cost.

Reynolds and Vahdat [8] used Bloom Filters to encode the transferred lists while recursively intersecting the matching document set. Consider an example of a two-keyword (x,y) query search. The query is first routed to the DHT node which is responsible for keyword x. Then the DHT node identifies X, the list of identifiers of documents that contain x. It then generates a Bloom Filter for set X, denoted by $BF(x)$, and transmits $BF(x)$ to the DHT node responsible for keyword y, where the intersection of X and Y is estimated based on $BF(x)$. Due to the possible false positives of BFs, the result set may include elements that contain only keyword y but not x. To pick out the false positives, the scheme sends the estimated intersection, denoted by $Y \cap BF(x)$, back to the DHT node responsible for keyword x to calculate $X \cap (Y \cap BF(x))$, which is equivalent to $X \cap Y$. By transmitting the BFs of the sets instead of the raw sets among DHT nodes during the intersection with an inverse verification, the communication cost can be effectively reduced. However, the length of the set is roughly proportional to the size of the network (document collection). The Bloom Filter-based scheme achieves a substantial constant factor improvement; but it does not eliminate the linear growth in the communication cost.

III. RELATED WORK

Full-text search is an important issue in distributed P2P information sharing systems. Without centralized index servers, nodes in a decentralized P2P system have to cooperate with each other to perform a full-text search. Existing P2P content search schemes can be divided into two types: DHT-based distributed global inverted index on top of structured P2P networks, and federated search engines over unstructured P2P networks.

A. Full-Text Search in Structured P2P Networks

DHT-based full-text searching engines utilize distributed global indexes, which partition a logically global inverted index in a physically distributed manner. Built on existing DHTs, single-term-based distributed index can effectively support single keyword search by retrieving the list of documents for a given keyword. Because of the utilization of the exact hashing techniques, the DHT-based schemes, however, fail to support complex queries with multiple keywords. Tang and Dwarkadas [6] proposed a hybrid index scheme, where the frequent terms of a document are selected to be published on a 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 for other keywords in the posting list. Finally only 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 [7] and such replication strategy may incur unacceptable storage and communications cost.

B. Search in Unstructured P2P Networks

It is commonly believed that unstructured P2Ps are promising to provide full-text content searching in large scale distributed environments. In this kind of search networks, peers which maintain indexes of their local documents are organized in an ad hoc fashion. Without a global index, unstructured P2P networks rely on flooding-based schemes to distribute queries to the network. Thus, the queries can be handled on peers containing relevant documents. Although unstructured P2P systems can naturally support full-text query evaluation, achieving efficient and effective search over unstructured P2Ps is challenging. First, the flooding-based protocols incur exponentially growing communication cost, restricting the scalability of the system. Second, the recall cannot be guaranteed unless a query is flooded exhaustively throughout the network.

C. Existing unstructured federated P2P search schemes

Often perform the query evaluation in two levels, the peer level and document level. The scheme first detects a group of peers with potential answers to the query, and then the query is submitted to the selected peers to evaluate the query against their local indexes and return the matched answers [9]. The search performance of unstructured federated P2P search engines can be further improved using super peer-based P2P architectures [10], which consider the inherent heterogeneity of peers [11]. Peers with more memory, processing power, and network connection capacity provide distributed directory services for resource location. Thus, the peers with limited resources won't become bottlenecks in the search network. Federated P2P search approaches can also take advantage of the enhanced properties of the network topology [12], [13] to improve search efficiency. Bloom Filter membership verification we design a query evaluation language to support full-text multikeyword search.

To solve the above problems, in this paper we propose BloomCast, a novel replication strategy to support efficient and effective full-text retrieval. Different from the WP scheme, BloomCast leverages a lightweight DHT for random node sampling. We mathematically optimal number of replicas is bounded by $O(\sqrt{N})$, where N is the network size. By further replicating the optimal number of Bloom Filters instead of the raw documents, BloomCast can achieve guaranteed recall rate while significantly reducing the communication cost for replicating. Based on the Bloom Filter membership verification we design a query evaluation language to support full-text multikeyword search.

IV. METHODOLOGIES

Evaluate the performance of BloomCast design using trace-driven simulations. In this section, we describe the simulation setup. First, we introduce the Gnutella traces we collected. We then describe the data used for evaluation

including the WT10G data collection from NIST and the query logs. Finally, we present the metrics used for performance evaluation. In order to well represent real world systems, we consider both the underlying physical topology and the P2P overlay.

The physical topology should represent the real topology with Internet characteristics. Previous studies [18] have shown that a large scale Internet physical topology follows the small world and power law properties. The topology of a small-world network has the properties of sparseness, short global separation, and high-local clustering of nodes while power law denotes the property of the node degree distribution. The study from Tangmunarunkit et al. [18] found that the topologies generated using the AS model have the properties of the small world and power law. BRITE [19] is a topology generation tool that provides the option of generating topologies based on the AS model Using BRITE, we generate a physical topology with 100,000 nodes.

Using the physical topology generated by BRITE, we can simulate the underlying Internet with rich configuration information, including bandwidth configuration, latency, and so forth. We have developed a crawler in Java based on the LimeWire open source client to collect topology information of Gnutella network [20]. We then use the traces to simulate a real P2P network. Using BRITE, we configure the upload bandwidth of a peer according to the measurement study on MSN from Microsoft [21] in 2007. The study has shown that 97.2 percent MSN video users have upstream bandwidth higher than 128 Kbps (16 KBps).

This corresponds to a DSL1 line quality. In the experiment, we set the upload bandwidth of a peer to 128 Kbps (16 KBps) and set the download bandwidth to 768 Kbps (96 KBps). On one hand, this conservative configuration about peer bandwidth capacity indeed pushes the system performance examination close to the system limits. On the other hand, in practice a real-world peer-assisted text retrieval system may not want to fully exploit the available bandwidth of a high capacity peer, as doing so might deter their participation.

All P2P nodes in the trace are mapped into the under-lying physical topology. The communication cost between two logical neighbors is calculated based on the physical shortest path between the pair of nodes. The uptime of peers follows the distribution of Gnutella P2P systems reported in [22]. About 10 percent ultra-peers have an average uptime longer than 80 minutes; among them 5 percent nodes are selected as the DHT nodes for node number estimating and node sampling. The Chord protocol [2] is used to connect the DHT nodes.

There has been no standard data set established for evaluating the performance of P2P content search. We built one based on TREC WT10G collection, a large test set widely used for performance evaluation in information retrieval research area. To evaluate the performance of BloomCast, in the simulation we implement three baseline schemes:

The WP algorithm presented

For the WP algorithm, we set the parameter of c to one and set the parameter to two. The TTL in the flooding

algorithm is set to seven. When simulating the DHT based multikeyword search algorithm, we set the size of Bloom Filter by $m = \lceil \log_2(2 \cdot |A| \cdot |B|) \rceil$ to achieve the minimized the false positive, where $|A|$ and $|B|$ are the sizes of the posting list in both sides during the intersection, respectively. When simulating the DHT-based scheme, we set j , the average URL length, to 250 bits based on the research results conducted on Google search engine, which shows that the average URL length measured in character is 31.2 characters.

V. PROPOSED WORK

We propose a novel strategy, called BloomCast, to support efficient and effective full-text retrieval in this paper. We show mathematically that the recall can be guaranteed at a communication cost of $O(N)$, where N is the size of the network. Bloom Cast hybridizes a lightweight DHT with an unstructured P2P overlay to support random node sampling and network size estimation.

Furthermore, we propose an option of using Bloom Filter encoding instead of replicating the raw data. Using such an option, BloomCast replicates Bloom Filters (BF) of a document. It is clear that the Bloom Cast model works only when the two constraints are met:

- 1) The query replicas and document replicas are randomly and uniformly distributed across the P2P network; and
- 2) Every peer knows N , the size of the network. To support random node sampling and network size estimation, BloomCast combines a lightweight DHT into the unstructured P2P network. To further reduce the replication cost, BloomCast utilizes Bloom Filters to encode the full documents.

A. Enhanced work (our work on this paper):

We are going to create local repository (virtual storage) to store the path code details of searched data. So that we can collect the data which we searched long back with effective and efficient search time. We can reduce the Storage cost for replication. We achieved 91% of query recall capacity using hybrid p2p protocol and local repository. On the other hand bloom cast achieved 57% of query recall capacity.

VI. CONCLUSION

In this paper, we propose BloomCast, an efficient and effective full-text retrieval scheme, in unstructured P2P networks. BloomCast is effective because it guarantees the recall with high probability. It is efficient because the overall communication cost of full-text search is reduced below a formal bound. Furthermore, by replicating Bloom Filters instead of the raw documents across the network, BloomCast significantly reduces the communication cost for replication. We demonstrate the power of BloomCast design through both mathematical proof and comprehensive simulations based on the TREC WT10G data collection and query logs from a real world search engine. Results show that BloomCast outperforms existing schemes in terms of both search results quality and system efficiency.

REFERENCES

- [1] D. Li, J. Cao, X. Lu, and K. Chen, "Efficient Range

- Query Processing in Peer-to-Peer Systems,” IEEE Trans. Knowledge and Data Eng., vol. 21, no. 1, pp. 78-91, Jan. 2008.
- [2] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, “Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications,” Proc. ACM SIGCOMM ’01, pp. 149-160, 2001.
- [3] E. Cohen and S. Shenker, “Replication Strategies in Unstructured Peer-to-Peer Networks,” Proc. ACM SIGCOMM ’02, pp. 177-190, 2002.
- [4] R.A. Ferreira, M.K. Ramanathan, A. Awan, A. Grama, and S. Jagannathan, “Search with Probabilistic Guarantees in Unstructured Peer-to-Peer Networks,” Proc. IEEE Fifth Int’l Conf. Peer to Peer Computing (P2P ’05), pp. 165-172, 2005.
- [5] H. Song, S. Dharmapurikar, J. Turner, and J. Lockwood, “Fast Hash Table Lookup Using Extended Bloom Filter: An Aid to Network Processing,” Proc. ACM SIGCOMM, 2005.
- [6] C. Tang and S. Dwarkadas, “Hybrid Global-Local Indexing for Efficient Peer-to-Peer Information Retrieval,” Proc. First Conf. Symp. Networked Systems Design and Implementation (NSDI ’04), p. 16, 2004.
- [7] S. Robertson, “Understanding Inverse Document Frequency: On Theoretical Arguments for IDF,” J. Documentation, vol. 60, pp. 503-520, 2004.
- [8] P. Reynolds and A. Vahdat, “Efficient Peer-to-Peer Keyword Searching,” Proc. ACM/IFIP/USENIX 2003 Int’l Conf. Middleware (Middleware ’03), pp. 21-40, 2003.
- [9] F.M. Cuenca-Acuna, C. Peery, R.P. Martin, and T.D. Nguyen, “Planetp: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities,” Proc. 12th IEEE Int’l Symp. High Performance Distributed Computing (HPDC ’03), pp. 236-246, 2003.