Content Based Searching Framework in Super Peer P2P Network

Muhammad Nazrul Islam*, Md. Ashiqul Islam*, Imam Jafar Shadaque*
and Md. Razib Hayat Khan**

ABSTRACT

Most existing Peer-to-Peer systems support only title based searches and are limited in functionality when compared to today's search engines. Moreover, searching is an important factor in p2p network for content retrieval. That's why without knowing the unique filename we can't retrieve the content of the file in title based search. In this paper, we designed the super peer p2p network that supports content-based search for relevant documents. First, we propose a general and extensible framework which is based on hierarchical summary structure for searching similar documents in p2p network. Second, based on the framework, we develop an effective document searching system, by effectively summarizing and maintaining all documents within the network with different factors. Finally, the experimental result is verified on a real p2p prototype and large-scale network is further simulated. The results show the effectiveness, efficiency and scalability of the proposed system.

Keywords: *Peer-to-peer, Content based search, Title-based search, Hierarchical summary, Indexing*

1 INTRODUCTION

Peer-to-Peer (P2P) computing has recently attracted a great deal of research attention. In a P2P system, a large number of nodes (e.g., PCs connected to the Internet) can potentially be pooled together to share their resources, information and services. Our system has several key features: unit level, peer level and super peer level. Summaries are first represented as vectors, which are further optimized by LSI [1] techniques and represented as high-dimensional points. In this paper, we address the problem of semantic-based content search in the context of document retrieval. Given a query, which may be a phrase, a statement or even a paragraph , we look for documents that are semantically close to the query. We propose a general and extensible framework for semantic -based

*Department of Computer Science & Engineering, Khulna University of Engineering & Technology, Khulna 920300, Bangladesh.
**Department of Computer and System Sciences, Royal Institute of Technology (KTH), Sweden.
E-Mail: {nazrul_bd80, swadkuet, raz_cit}@yahoo.com

content search in P2P network. The super-peer P2P architecture which is more efficient for contents look-up is employed as the underlying architecture. To facilitate semantic-based content search in such a setting, a novel indexing structure called Hierarchical Summary Indexing Structure, is proposed. With such an organization, all information within the network can be summarized with different granularity, and then efficiently indexed. Based on this framework, we develop our distributed document search system in P2P network.

2 RELATED WORKS

We will first review previous work on P2P architecture [3]. Provides an analysis of *hybrid* P2P architecture develops an analytical model and uses it to compare various hybrid P2P architectures. [2] extends [3]'s hybrid architecture to design *super*-peer network, which strikes a balance between the inherent efficiency of centralized search, and the autonomy, load balancing and robustness to attacks provided by distributed search.

Much research effort has focused on improving search efficiency by designing good P2P routing and discovery protocols. However, current systems support only simple queries. For example, Freenet[13], Gnutella[14] and Napster[15] only provide filename-based search facility, which means that the end user cannot retrieve content unless he knows a file's unique name. Queries are broadcast to neighbors which in turn disseminate the queriesto their neighbors and so on. Thus, these systems can lead to long response time. Chord[10] and CAN[16] are designed for point queries and focus only on the problem of query routing and object allocation. [17] and [18] support keyword queries with regular expressions. Hence so far the queries issued by clients are up to context of keyword's complexity and for keyword matching only. More recently, PlanetP [9] presents a distributed text-based content search algorithm in P2P communities. Each peer has a summary produced by VSM. A local inverted index is then built on this summary. However, to our knowledge, there has not been much work done to facilitate efficient *semantic-based content* search for document retrieval in P2P sharing systems. Issue on fair load distribution has also been addressed by [12].

Summary techniques are crucial in P2P systems. Due to limit on network bandwidth and peer storage, it is not practical to transmit the complete information of a peer to the other peers in the network. Moreover, a peer usually contains thousands of shared files or more. To decide which peer to route the query to needs a similarity comparison between the queries and peer's information. From the above discussion, it is clear that effective summarization of peer information is absolutely needed in P2P network. So far, the only known summarization technique for text documents in P2P systems is keywords representation. Existing P2P systems, such as [17], [18] etc, summarize the peers/documents by keyword vectors which

contain pairs of keyword and its weight. Given a query, which is also represented as a vector, the similarity between the query and the summary of peers/documents are then computed However, such techniques are limited to exact keyword matching only and cannot be applied for semantic-based content search. In this paper, we propose a hierarchical summary indexing structure for efficient semantic-based content search in super-peer P2P network, which can support complex semantic-based queries.

Another related area is high-dimensional indexing. In the literature, many high-dimensional indexing methods have been proposed. A survey can be found in [4]. However, existing methods are typically not efficient for more than 30-dimensions and are not scalable [5] due to the dimensionality curse phenomenon when the dimensionality reaches higher. VA-file [5] however, has been shown to be superior in nearly uniform datasets by *LP* distance functions. In this paper, we extend VA-file to support a different similarity metric for document similarity search.

3 A GENERAL FRAMEWORK FOR SUPER PEER P2P-BASED SEARCH

In this section, we present a novel Hierarchical Summary Indexing framework for P2P-based document search system. We shall first discuss the super-peer P2P architecture, and then look at how such a structure can facilitate the design of the proposed framework.

3.1 Super-peer P2P Network

A *super-peer* is a node in a peer-to-peer network that operates both as a server to a set of clients, and as an equal in a network of super-peers. A straightforward query processing mechanism in super peer network works as follows. A peer (client) submits its query to the super peer of its group. The super peer will then broadcast the query to other peers within the group. At the same time, the super peer will also broadcast the query to its neighboring super peers.

3.2 Hierarchical Summary Indexing Structure

Summarization is a necessary step for efficient searching, especially when the amount of information is very large. A summary is a very compact representation. In our framework, we introduce a new interesting concept, Hierarchical Summary Indexing Structure (Summary and Indexing), which is closely related to the super-peer P2P architecture we employed. Our scheme essentially summarizes information at different levels.

We have employed three levels of summarization in our framework. The lowest level, named as *unit level*, an information unit, such as a document is summarized. In the second level, named as *peer level*, all information owned by a peer is summarized. Finally, in the third level, named as *super level*, all information contained by a peer group is summarized. Fig.1 depicts such a

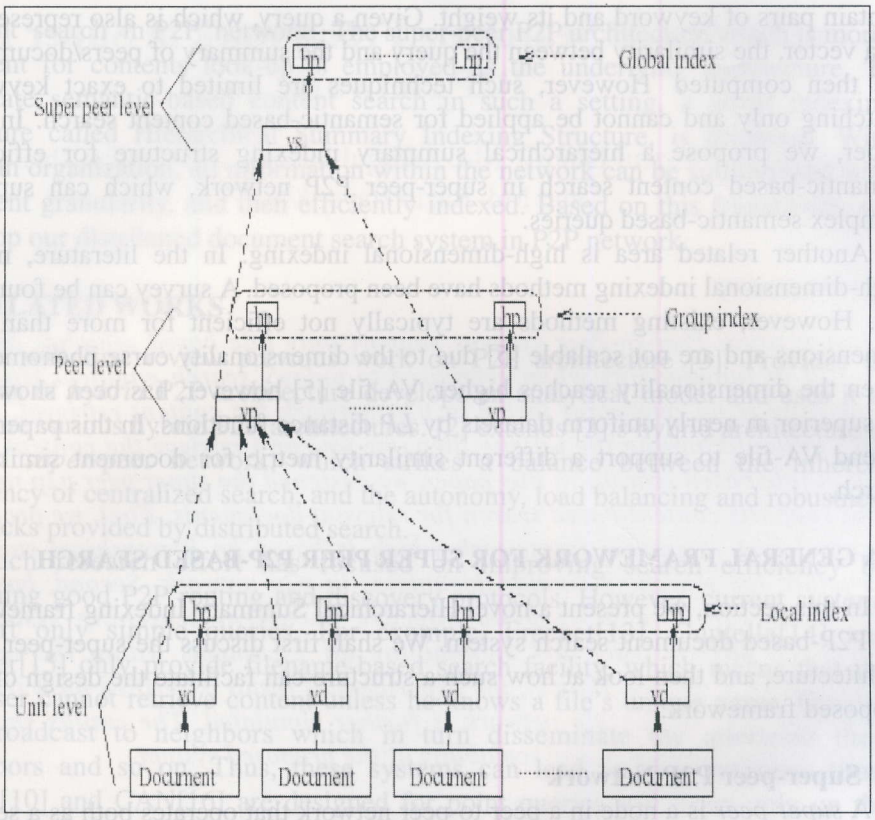

Figure 1: Hierarchical Summary Indexing Structure

structure for document summary. Fig.1 shows each level of summary has a corresponding index built on top of it. Fig.2 shows the hierarchical summary indexes in a peer group.

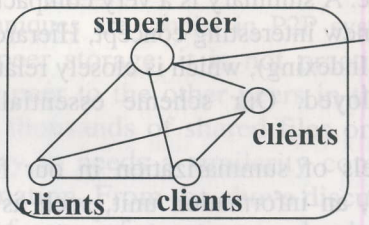

Figure 2: Summary indices in a peer group

4 SEMANTIC BASED CONTENT SEARCH SYSTEM

Suppose there are a large number of peers in the network, and each peer contains a large number of documents, what we want to achieve is to find the most relevant documents as quickly as possible.

4.1 Building Summary

For each level, our summarization process consists of two steps by techniques of Vector Space Model (VSM) [6] and Latent Semantic Indexing (LSI) [1] respectively. Briefly, in VSM, documents and queries are represented by vectors of weighted term frequencies. Latent Semantic Indexing (LSI) has been proposed to optimize the vector prepared by VSM. A technique known as Singular Value Decomposition (SVD) is used to reduce this concept space into a much lower dimensionality.

Algorithm 1: Building Hierarchical Summaries

1. For each peer
2. For each document
3. Generate its vector *vd* by VSM
4. Generated peer weighted term dictionary *vp*
5. For each document vector *vd*
6. Transform it into *D(vp)* dimensionality
7. Generate high dimensional point for *vd* by SVD
8. Pass *vp* to its super peer
9. For each super peer
10. Generated group weighted term dictionary *vs*
11. For each *vp*
12. Transform it int *D(vs)* dimensionality
13. Generate high dimensional point for *vp* by SVD
14. Pass *vs* to other super peers
15. Generated global weighted term dictionary *vn*
16. For each *vs*
17. Transform it into *D(vn)* dimensionality
18. Generate high dimensional point for *vs*

Figure 3: Algorithm for building hierarchical structure summaries.

Algorithm 1 indicates the main routine of building summary in the hierarchical structure as shown in Fig.3. We illustrate the algorithm in Example 1.

EXAMPLE 1 (AN EXAMPLE OF HIERARCHICAL SUMMARY BUILDING)

Table 1 provides a small P2P network with eight documents, di^n_m represents the i^{th} document which is in m^{th} peer Of n^{th} group. The process of summary building is depicted in Fig.4, where zthe weight of a term is represented by its frequency only. As we can see, vectors of documents vds within a peer form the vector of peer vp. Based on vp, each vd is transformed into D(vp)-dimensional vector which is then reduced into a 2-dimensional document summary by SVD. Take a look at the first peer which contains documents $d1_1^1$ and $d2_1^1$. Both documents are merged to form its vp of together with term weights, where D(vp) is 5. Based on vp, both documents are mapped into 5-dimensional vectors of (1,0,1,1,1) and (0,1,0,1,0) respectively, which are in turn reduced into a much lower 2-dimensional points by SVD. Same process is applied to generate a peer's and supper peer's summary.

Table 1: A table of documents.

Id	Document
$d1_1^1$	Monitoring XML data on the web
$d2_1^1$	Approximate XML joins
$d3_1^2$	Outlier detection for high dimensional data
$d4_1^2$	High dimensional indexing using sampling
$d5_2^1$	Document clustering with committees
$d6_2^1$	Document clustering with cluster refinement
$d7_2^2$	Title language model for information retrieval
$d8_2^2$	Document summarization in information retrieval

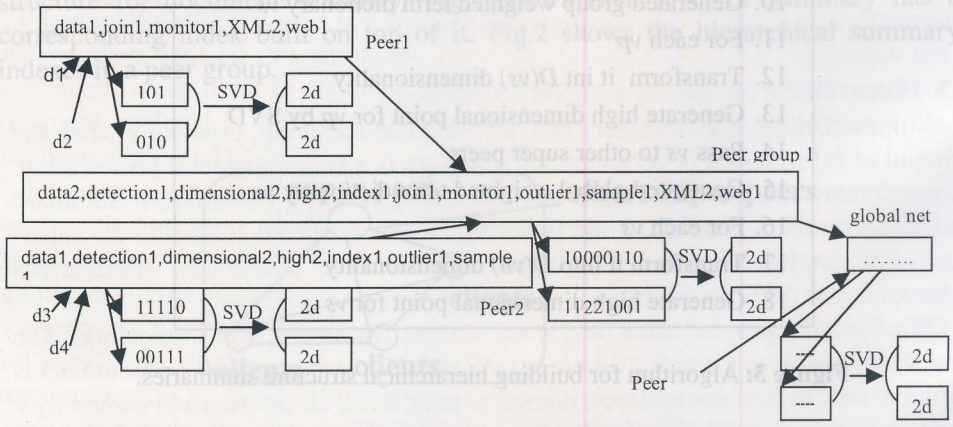

Figure 4: An example of hierarchical summary building.

4.2 Query Processing

Fig.5 depicts how a query is being processed in a P2P network. In the figure, dark arrow indicates the direction of a query being transmitted and blank arrow indicates the route of results being returned. When a peer issues a query Q, Q is first passed to its super peer, followed by the hierarchical indexing search in order of global index, group index and peer index, which is the reverse order of the summary construction.

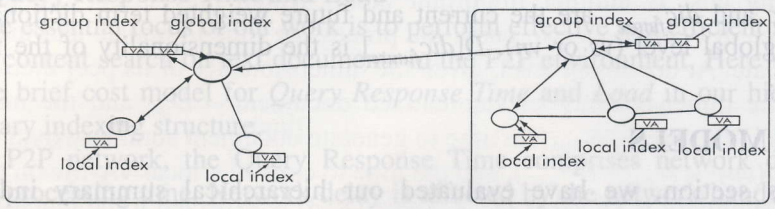

Figure 5: The routing of query processing initialized by dark peer

5 UPDATING ISSUES

One crucial difference between P2P and traditional information retrieval is that P2P network is dynamic in nature. A peer can join and leave the network at any time.

Algorithm 2: Peer insertion

1. Build peers local index
2. Pass peers vp to its super peer
3. If AIR $_{group} > \theta_{group}$
4. Rebuild and index group peers summary
5. Update super peers
6. Broadcast vs to other super peers
7. For each super peer
8. If AIR $_{global} > \theta_{global}$
9. Rebuild and index super peers summary
10. Else
11. Generate peers high - dimensional point
12. Insert the point into groups index

Figure 6: Algorithm for peer insertion

Hence the summarization and indexing techniques have to be able to handle dynamic operation efficiently. We propose the following peer insertion Algorithm 2 in our hierarchical indexing structure as shown in Fig.6

$$AIR(dic, dic_{future}) = \frac{\sum_{i=0}^{D[dic_{future}]} \left| dic[i] - dic_{future}[i] \right|}{\sum_{i=0}^{D[dic_{future}]} dic[i]}$$

Where dic and dic_{future} are the current and future weighted term dictionary at group or global level (*vs* or *vn*). $D[dic_{future}]$ is the dimensionality of the future dictionary.

6 COST MODELS

In this section, we have evaluated our hierarchical summary indexing structure in a super-peer network based on the following types of metrics: *Storage Overhead, Query Response Time*, *Load* and *Accuracy of Results*. Furthermore, the cost for indexing construction/updating is also estimated.

6.1 Storage Overhead

The storage overhead in our structure includes peer overhead and super peer overhead. For peer overhead, each peer contains its documents' summary; local index built on the document summary, local current and future dictionaries, together with the SVD's Singular Vectors. Hence the total peer Storage Overhead (SO) is:

$$SO_{peer} = D_{doc} * N_{doc} + VA_{local} + 2D(vp) + D_{doc} * D(vp)$$

where D_{doc} is the dimensionality of summarized high-dimensional points of documents, N_{doc} is the number of documents in the peer , VA_{local} is the size of local VA-file on points of documents, and $D(vp)$ is the dimensionality of peer's term dictionary (assume current and future dictionary have approximately same dimensionality).

Usually the value of D_{doc} is about 100, and $D(vp)$ is about thousands. $D_{doc} * N_{doc}$ represents the size of documents' summaries. Given that each dimension is represented by b bits, the size of VA is $32/b$ of summaries, assuming each dimension of summary is 4-byte long. Obviously, when N_{doc} is very large,

$$SO_{peer} \approx D_{doc} * N_{doc} + D_{doc} * D(vp)$$

Similarly, each super peer contains two sets of data: summaries, index of summaries, term dictionaries and Singular Vectors at group level and global level. Hence the total super peer Storage Overhead is:

$$SO_{super} = D_{peer} * N_{peer} + VA_{group} + 2D(vs) + D_{peer} * D(vs) + D_{super} * N_{super} + VA_{global} + 2D(vn) + D_{super} * D(vn),$$

where D_{peer} and D_{super} are the dimensionality of summarized high-dimensional points of peers and super peers, N_{peer} and N_{super} are the number of peers in the group and super peers in the network. VA_{group} and VA_{global} are the sizes of group VA-files on points of group peers and global VA-file on points of super peers. $D(vs)$ and $D(vn)$ are the dimensionality of group and global term dictionary respectively.

6.2 Query Response Time and Load

The essential focus of our work is to perform effective and efficient semantic-based content search on text documents in the P2P environment. Here we derive simple brief cost model for *Query Response Time* and *Load* in our hierarchical summary indexing structure.

In P2P network, the Query Response Time comprises network delay and peer's processing time. Network delay is affected by the network bandwidth and network *Load*. Here we define *Load* as the total amount of information the network must transmit. For simplicity, we measure the *Load* as the number of messages being processed in the network. Given the fixed network bandwidth and a time period, the number of messages is the major factor affecting the network traffic. A peer's processing time is the time to search the K most relevant peers and/or documents. Without summarization, the query has to compare with a peer's/document's complete information. And without indexing, the query has to compare with every single peer/document. Given the fixed CPU power, effective summarization and efficient indexing become two keys.

Our hierarchical indexing structure is designed to avoid network delay and speed up the processing time. In our structure, global index and group index are used to quickly determine which group and peer to be searched next. This avoids extensive broadcast cost. As for the peer's processing time, local index quickens the local document search. Effective summarization condenses large amount of information into small size and makes efficient indexing possible.In our hierarchical indexing structure, given a query, the times of a query being forwarded, is:

$$Times_{query} = 1 + K_{group} + K_{group} * K_{peer}$$

The client peer first forwards its query to its group's super peer. Its super peer then searches its global index and selects the K_{group} most relevant groups to which it forwards the query. In each selected group, the super peer searches its group index and forwards the query to the K_{peer} most relevant individual peers. At each level of index, KNN search is performed. Correspondingly, the total processing time is computed as:

$$Time = Time_{global} + K_{group} * Time_{group} + K_{group} * K_{peer} * Time_{local}$$

where, $Time_{global}$ refers to the processing time of KNN search in global VA-file. Clearly, the efficiency of indexing technique determines the performance. In the

VA-file, the total response time for KNN search mainly includes two parts: the time to scan the VA file and the time to compute the lower and upper bounds if the number of candidates is small. It has been shown that it outperforms sequential scan in high dimensional space. As for the accuracy of results achieved by our summary technique, they will be measured in the experiments section.

6.3 Cost of Updating

Updating cost is another important factor which may affect the overall performance. It consists of two parts: the processing time to update and the load for the update. At document level, given the dimensionality of its peer dictionary - D(vp), the dimensionality of summarized document – D_{doc} and the number documents in the peer – N_{doc}, the time to generate document summary by SVD is $O(N_{doc} * D(vp)^2 + N_{doc} * D^2_{doc})$, and the time to construct local VA-file on top of document summary is $O(N_{doc} * D_{doc})$. Since $D(vp)$ is expected to be much larger than D_{doc}, the total processing time to construct local index is approximately $O(N_{doc} * D(vp)^2)$. Similar formulas can be derived for peer level and super peer level.

Whenever a peer joins a group, its local index is built and its dictionary - *vp* is passed to its super peer. Hence the load is the peer's *vp*. In our peer insertion algorithm, if no re-building of group indexing occurs in the super peer, the joined peer is first mapped to D_{peer} dimensional summary point, and then its *VA* is appended to the VA-file, which takes constant time.

However, if group summary rebuilding is invoked, the total cost will include two more portions: super peer's processing time for group indexing building, and the broadcast load of the super peer's summary - *vs* to other super peers. The processing cost of group indexing building is approximately $O(N_{peer} * D(vs)^2)$ as explained above. The times of the super peer's summary being broadcast is proportional to the number of super peers which can be reached by this super peer. If a super peers' summary rebuilding is invoked, each reachable super peer performs the same process of summary indexing, which approximately takes $O(N^2_{super} * D(vn)^2)$ more processing time, by assuming all super peers can be reached. It can be expected that directly inserting peer into group index is much more frequent than rebuilding of group index, which in turn is more frequent than rebuilding of global index.

Obviously, in the process of indexing building/rebuilding, summary generation by SVD is the most time consuming. To reduce the cost of SVD, sampling techniques can be applied to achieve a better trade-off between time and accuracy. In the experiment section, we will see how the sampling technique can help to reduce the processing time while keeping a high accuracy. Furthermore, we will also see how stable the SVD technique can be in producing the accurate summary information as peers keep joining the network and when the index need to be rebuilt.

7 EXPERIMENTAL RESULT AND DISCUSSION

We have evaluated the proposed hierarchical summary and indexing scheme in a real P2P setting as well as via simulation. In this section, we report the results of the performance study.

7.1 Experimental Setup

Table 2 gives some experiment parameters and their default settings for both the real system and the simulator respectively.

Table 2: Parameters and settings.

Name	Default Value	Description
Network Type	Power-Law	Topology of the network, with out degree 3.2
Max User Wait Time	60s	Time for a user to wait an answer
Query Rate	8E-3	The expected number of queries per user per second
TTL	5	Time-To-Live of an message
Network Size		Number of peers in the network
Peer Group Size		Number of peers in each peer group
K_{group}		Number of super peers to return
K_{peer}		Number of peers for a super peer to return
K_{doc}		Number of documents for a peer to return

7.2 Retrieval Precision

In this experiment, we have examined the effectiveness of our summary technique. We first implement a relatively small real network to show that our proposals are very practical and applicable to P2P systems. Our real network has 30 nodes. We use 4 benchmark collections of documents which were used by Smart [7], together with their queries and human ranking. Table 3 presents the characteristics of the datasets.

Table 3: Characteristics of real datasets.

	MED	CISI	CACM	TIMES
Number of documents	1033	1460	3204	425
Number of queries	30	76	64	83
Number of terms occurring in more than one document	5831	5743	4867	10337

i) Effect of Dimensionality: The precision is measured by the ratio of the number of relevant documents over the number of returned documents. Fig.7 shows the changes of the average precision when the summary for the documents is reduced to different dimensions with SVD technique.

Next, we study the retrieval precision at the group level with the peer level summaries. Fig.8 illustrates the variation of the average precision as the number of dimension increases. Lastly, we repeat the experiment on the highest level of hierarchy to test if the correct groups that contain the relevant documents can be returned.

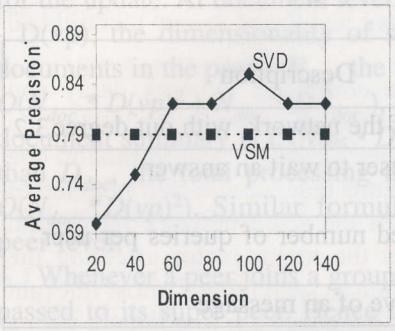

Figure 7: Document level summary precision.

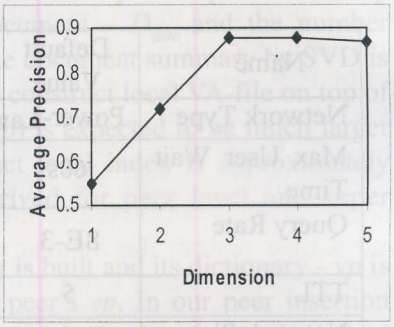

Figure 8: Peer level summary

At this level, its precision is measured by the ratio of the number of relevant peer groups over the number of returned peer groups. The result is shown in Fig.9. From Fig.8, Fig.9 and Fig.10, we can see that different dimensionality of summary may achieve different precision. The smallest value with highest precision is always chosen as the final dimensionality of summary at each level.

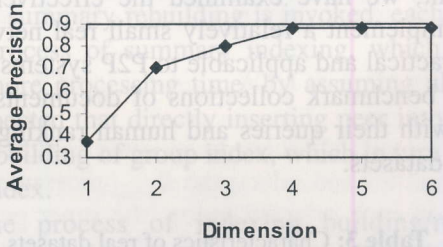

Figure 9: Super Peer Level Summary Precision.

ii) Precision of the Whole System: In the above subsection, we have seen how the dimensionality of summary affects the precision at each individual level. In this experiment, we integrate the three levels and test the overall precision of the whole system. The precision is measured by the ratio of the number of relevant documents over the number of returned documents after the

whole network has been searched. Obviously, the precision of the whole system is expected to be lower than the precision at documentation level since the precision is further reduced at higher levels.

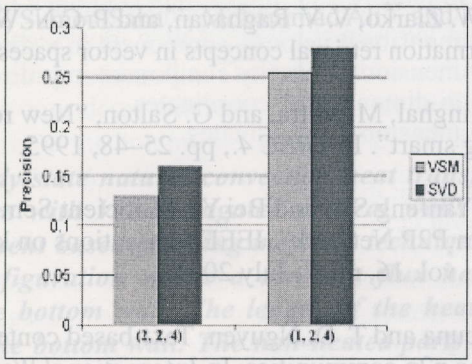

Figure 10: Overall Hierarchical System Summary Precision.

8 CONCLUSIONS

We have examined the issues to support content-based searching in a distributed peer-to-peer information sharing system. We have proposed the first general and extensible hierarchical framework for summary building and indexing in P2P network. Based on this framework, we have presented an effective two-step summarization technique to transform large size representations of documents, peers, and super peers into small high-dimensional points. A prototype and a simulated large scale network have been designed to evaluate the system performance. Our experiments showed that such a hierarchical summary indexing structures can be easily adopted and our prototype system achieves remarkable achievements.

REFERENCES

[1] C.H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala, "Latent semantic indexing: A probabilistic analysis", In *PODS*, 1998.

[2] B. Yang and H. Garcia-Molina, "Designing a super-peer network", In *ICDE*, 2003.

[3] B. Yang and H. Garcia-Molina, "Comparing hybrid peer-to-peer systems", In *VLDB'2001*, 2001.

[4] S. Berchtold and D. A. Keim, "Indexing high-dimensional spaces: Database support for next decade's applications", *ACM Computing Surveys*, 33(3): pp. 322–73, 2001.

[5] R. Weber, H. Schek, and S. Blott, "A quantitative analysis and performance study for similarity search methods in high dimensional spaces", In *VLDB*, pp. 194–205, 1998.

[6] S. K.M. Wong, W. Ziarko, V. V. Raghavan, and P. C.N. Wong, "On modeling of information retrieval concepts in vector spaces", In *TODS*, 1987.

[7] C. Buckley, A. Singhal, M. Mitra, and G. Salton, "New retrieval approaches using smart". In *TREC 4.*, pp. 25–48, 1995.

[8] Heng Tao Shen, Yanfeng Shu and Bei Yu, "Efficient Semantic-Based Content Search in P2P Network", IEEE transactions on knowledge and data engineering, vol. 16, no. 7, July 2004.

[9] F. M. Cuenca-Acuna and T. D. Nguyen. Text-based content search and retrieval in ad hoc p2p communities. In *International Workshop on Peer-to-Peer Computing*, 2002.

[10] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications.In *SIGCOMM*, 2001.

[11] Freenet. *http://freenet.sourceforge.com/.*

[12] P. Triantafillou, C. Xiruhaki, M. Koubarakis, and N. Ntarmos. Towards high performance peer-to-peer content and resource sharing systems. In *CIDR*, 2003.

[13] Freenet. *http://freenet.sourceforge.com/.*

[14] Gnutella. *http://gnutella.wego.com/.*

[15] Napster. *http://www.napster.com/.*

[16] S. Ratnasamy, P. Francis, M . Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *SIGCOMM*, 2001.

[17] B. Yang and H. Garcia-Molina. Improving efficiency of peer-to-peer search. In *28th Intl. Conf. on Distributed Computing Systems*, 2002.

[18] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *28th Intl. Conf. on Distributed Computing Systems*, 2002.