A Hybrid Keyword Search across Peer-to-Peer Federated Databases

Jungkee Kim[1,2] and Geoffrey Fox[2]

[1] Department of Computer Science, Florida State University, Tallahassee FL 32306, U.S.A.,
jungkkim@cs.fsu.edu,
[2] Community Grids Laboratory, Indiana University, Bloomington IN 47404, U.S.A.
gcf@indiana.edu

Abstract. The need for Keyword search in databases is suggested both by Web integration with legacy database management system and by dynamic Web publication. However, it sacrifices the inherent meaning of database schema. Web search engines provide clues for resource location on the Web, but have the similar semantic problems. The Semantic Web suggests an ideal solution for the semantic problem on the Web. But due to the need for sophisticated domain definition and lack of unified definitions, many Web pages are not part of the Semantic Web. A hybrid search—combining semantic metadata and keywords—on P2P based federated databases provides meaningful and scalable search on an overlay network across the Internet. This paper describes the design of the combined search for unstructured data linked in metadata, information retrieval from the repository, and peer-to-peer based communication layer.

1 Introduction

Since the Internet was introduced as a communication and resource sharing environment, there have been many efforts to utilize its enormous and rapidly proliferating resources. *Keyword search* in databases [1, 15] is one response to the new Internet environment from the database society. Both Web integration with legacy database management systems, and dynamic Web publication through the embedded databases, strongly benefit from keyword search capability on the databases. This is because conventional queries on databases requires knowledge of the schema to extract target information. Additionally, semistructured schema like XML have been assimilated to the Web and databases recently. These make it more complicated to produce a proper inquiry. Though the keyword based search simplifies the search on the database, it loses the inherent meaning of the schema. Therefore, the keyword search usually does not return results based on semantic criteria.

A Web search engine is a typical example for use of the Internet. In 2004, Google Search [5]—one of most famous search engines—reaches more than 4 billion Web pages and provides very fast information retrieval through its collected indexes. However, the search results from the Web search engines are often highly irrelevant, because the crawled Web contents are only text indexed without any semantic schema. Some reputation models increase the probability to reach the appropriate information resource, but the search results still have inferior semantics comparing to the resource provider's explicit information links.

The *Semantic Web* [4] is an ideal extension of the Web. Data on the Semantic Web also includes information by which a machine can interpret the meaning. To represent the relations of the objects on the Web, the object terms should be defined under a specific domain description—an ontology. Domain experts usually design an ontology. The Semantic Web provides multiple relation links with directed labeled graphs and machines like Web crawlers can understand the relationship between different resources. The ordinary Web has hyperlink—a single relationship, and machine can not interpret further meaning like thesaurus. Currently, most Web pages include no such semantic content, and no unified definition of general semantic agreement exists.

On the other hand, the resurrection of the *Peer-to-Peer* (P2P) based networks makes it possible to provide customized overlay networks on top of the Internet. The desire for sharing resources such as music files quickly led to creation of a number of peer groups [23]. The idle CPU time of home computers can be exploited in a large computation and contribute to finding extraterrestrial intelligent life [30].

In this paper we will focus on mutual interest groups whose resources need to be shared and searched through an overlay network. The peer-to-peer connection of the search services provides low cost scalability. Associating additional schema to the resource is not as difficult as for the Semantic Web, but it will increase the information transparency. This model could be extended to give a simplified compromise between the Semantic Web and the Web search engine. Though the peer-to-peer overlay networks usually increase network traffic, modern network technology has already overcome the barrier of text only communications, and large amount of multimedia content is delivered through the current Internet. Many network research groups [31, 28, 29, 33] have focused on Distributed Hash Table (DHT) for distributed lookup in routing to reduce the unnecessary communications of P2P based network, but their mechanisms are based on the stabilized peer structure. We utilize an open peer-to-peer framework (JXTA [21]) that also considers the destabilized peer structure. The communication in our system depends on the default network control policy of the P2P framework. The final version of this paper will provide performance measurements on metadata and keyword query processing, and the P2P communication cost.

The rest of this paper is organized as follows. In the next section we describe the hybrid search and its repository for the metadata and the unstructured data targeted keyword search. Section 3 describes the architecture of the P2P framework and the key modules—query processing, communication layer, and data integration hub. We cite relevant studies of searches based on P2P networks in Section 4. We summarize and conclude in Section 5.

2 Hybrid Search

Our development of hybrid search [18] was primarily motivated by the need to search large legacy body of educational documents with additional metadata [25]. A chunk of unstructured data typically is a file containing those data. The format of the file can be pure text, *Microsoft Word*, *Adobe Portable Document Format* (PDF), *PostScript*, or any other format of binary or text document for which a program is able to read text. For information retrieval, indexing on the text data is essential to search a keyword within an appropriate time. In a search service based on file systems, text filtering may be necessary to support the various data formats. Through the filter, we can obtain the unified format documents—the pure texts—and the document indexing program is uniformly applied to those text data. The market leading commercial database systems—IBM DB2, Microsoft SQL Server, and Oracle—have integrated text management in their systems and their query languages include text search syntax [20, 11, 8].

The metadata may be structured or semistructured to describe the information content of the data. We use XML, a semistructured data format, as a description language for the metadata. This is because XML provides a unified format among heterogeneous databases. Similar to the case of unstructured data repositories, metadata storage can be based on low-level file systems. To extract the information from XML instances in files, parsing and matching programs are needed. The extracted metadata information is combined with the query result of content search against the unstructured data. The relational database management systems are able to keep XML instances in their relational tables by the mapping paradigm. XML-enabled relational databases [6, 3, 26] are useful for the metadata storage, because the mapping and querying features are embedded in the database management systems. Native XML databases have excellent features to manipulate XML instances, but they are document-centric storages.

The initial objective was to achieve both content and metadata search against documents associated with separate metadata. Existing searches focused on keyword-only or XML-instance-only search. We address the hybrid search with a simple relationship as in figure 1. One entity is allocated for the metadata described in XML and the other entity is for the contents stored in external files. The keyword attribute for the metadata entity is a naming and directory for the each file and it is a file name for the contents entity.

In an early experiment to establish this approach is practical at a local peer level [18], we evaluated the performance of the XPath query against the metadata, we used 10,000 data-centric XML instances on a 1700 MHz Pentium 4 PC with 512 MB of memory, running a Microsoft Windows 2000 and a commercial DBMS. Later, we used the later version of the DBMS which improved the indexing over XML repository. Table 1 shows the performance results. The newer version provides the appropriate environment for our hybrid search. The XML query processing in older version was the bottleneck of the query response time.

Another experiment was made for the non-DBMS based repositories. We used 10,000 XML instances extracted from DBLP XML record [19] and another 10,000

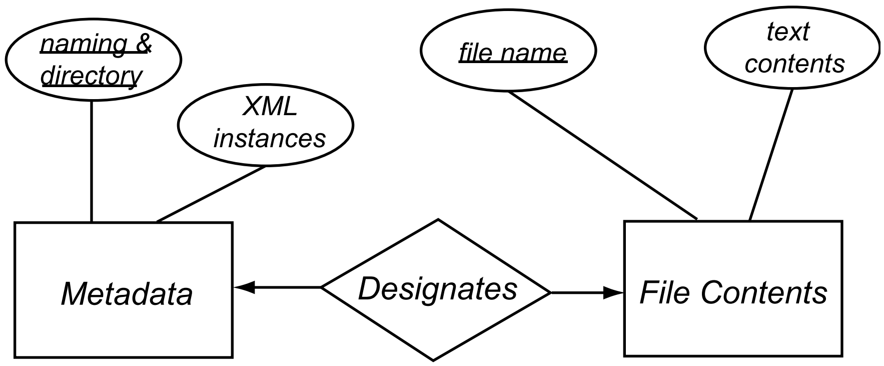

Fig. 1. E-R diagram of hybrid search

Table 1. Time for querying an XPath

Time	old version	new version	new version with context indexing
Sec.	74	27	0.06

abstract text files from the TREC collection, OHSUMED [13]. The performance was measured on an Athlon 1800 machine with 512 MB of memory, running a Linux 2.4 kernel and Java Hotspot VM 1.4.1. Apache Xindice 1.1b4 was used for the XML query and Apache Lucene 1.3 was used for the keyword query. The join operation was done by Java code with a hash table. Figure 2 and 3 shows the XML query performance from 10,000 to 100,000 XML instances. The exact match used an XPath for an author with a full name, while the approximate match required only a partial match for an author name using the CONTAINS function. We picked 1972 for the year element match, a year that is relatively rare in the data set. The indexed queries are usually fast, but the approximate match is as slow as the other non-indexed cases. The figure 3 shows the year query is slower than the exact match for the author. This is because the year query produces a large iteration result set. The combined query has an XPath AND operation for both the year and the author. Figure 4 shows the hybrid query performance measurement. For the large data set, the final result combined year matches in 192 XML instances with keyword matches in 4,562 documents. For the small data set there were just two XML matches and a single keyword match.

3 P2P Framework

Our hybrid peer-to-peer keyword search framework provides hybrid search mechanisms that combine semantic metadata as well as keyword search. A hybrid search is not as solid a semantic solution as in the Semantic Web, but it also does not require sophisticated definition for the object terms and it provides partially semantic search through metadata association to resource contents. We

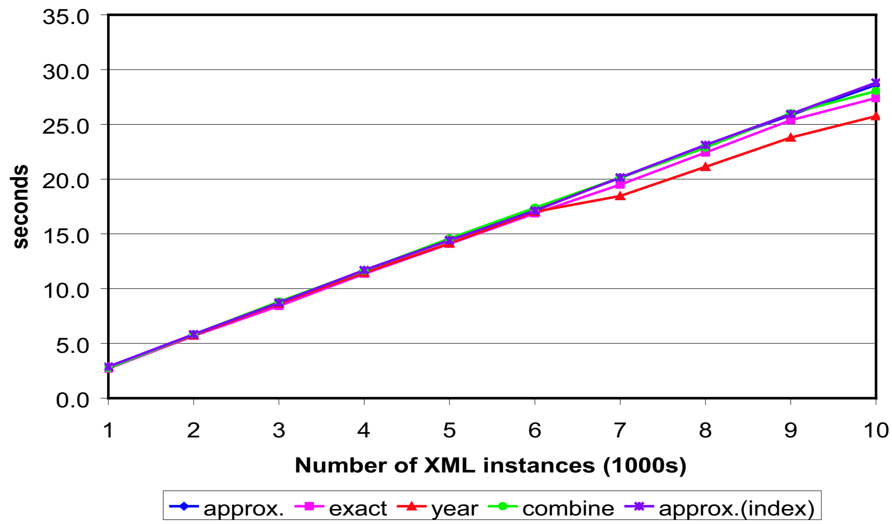

Fig. 2. Xindice query processing time with no index

aim to federate mutual interest group services scattered in the Internet for resource sharing and search. For communication between those services, we utilize a current network technology. The recent spotlight on the peer-to-peer network suggests that a P2P network is a possible solution for scalability on top of the Internet. Therefore, our hybrid peer-to-peer keyword search framework provides a meaningful and scalable search on an overlay network of search services. To organize the search framework, we are building the following modules:

- **Repositories for unstructured data and their metadata.** The search target data and their metadata are stored and managed in the search peer. The minimum requirement for the repository is that a search service peer should fetch target information data against a search inquiry within appropriate time. Indexing of the data is desirable for efficiency.
- **Query processing for the repository and a wrapper to convert queries.** The query processing module extracts the target result against the user inquiries. Most database management systems provide query processing as a primary module for their repository retrieval. However, an additional wrapper may be needed to convert the peer query to the internal retrieval commands.
- **User interface.** A user friendly graphical user interface or Web browser interface will give the user easy access the search system.
- **Communication layer for the peer-to-peer overlay network.** Each search service peer should communicate to share its data with other service peers. Core functionalities for the peer-to-peer communications are desirable.
- **Data integration hub on top of message-oriented middleware.** A data integration hub will reduce the query propagation traffic between peers.

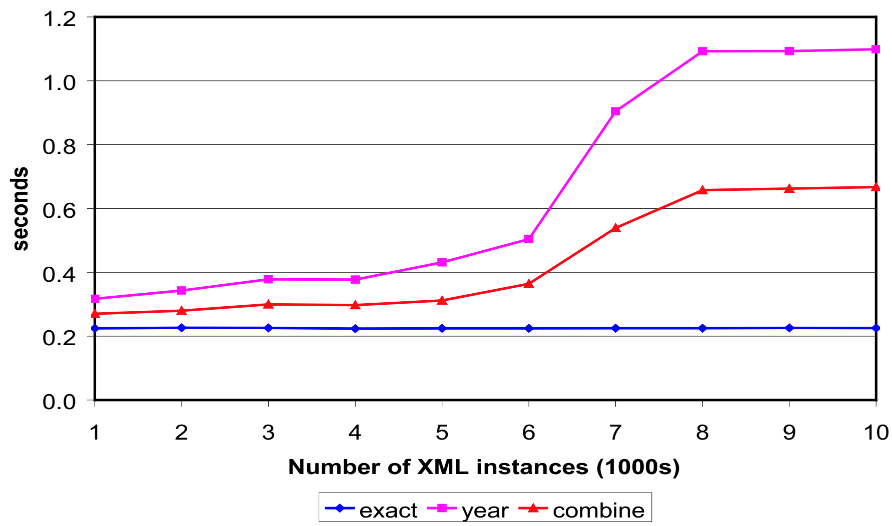

Fig. 3. Xindice query processing time with index

We utilize the functions of message-oriented middleware for integration of the search data.

– **Cache.** The information advertisement cache, the result data set cache, and even the cache in the repository will increase the performance.

The figure 5 shows the structure of the search peer framework.

We implemented hybrid search services at a local peer level with two different repositories—a commercial DBMS and non-DBMS repositories with Apache Xindice and Apache Lucene. Query processing depends on the DBMS and those Apache software packages. The join operation is implemented through the relationship table and nested subqueries in the DBMS. Java code with a hash table is used for joining two result sets in the non-DBMS based repositories. Those heterogeneous search services are connected on top of a message-oriented middleware as a distributed database, but the performance for P2P connection is not completely evaluated yet. The text-based user interface is running for the current prototypes.

3.1 Query processing

Query processing is a group of program activities that provides a search result against the search inquiry. The individual search service peer should provide query processing. The delivered query has to include the information about the target metadata and keywords of the data, but it does not necessarily have a particular SQL format. This will be discussed in the next paragraph. The query can be interpreted, and translated to the internal codes or to the SQL

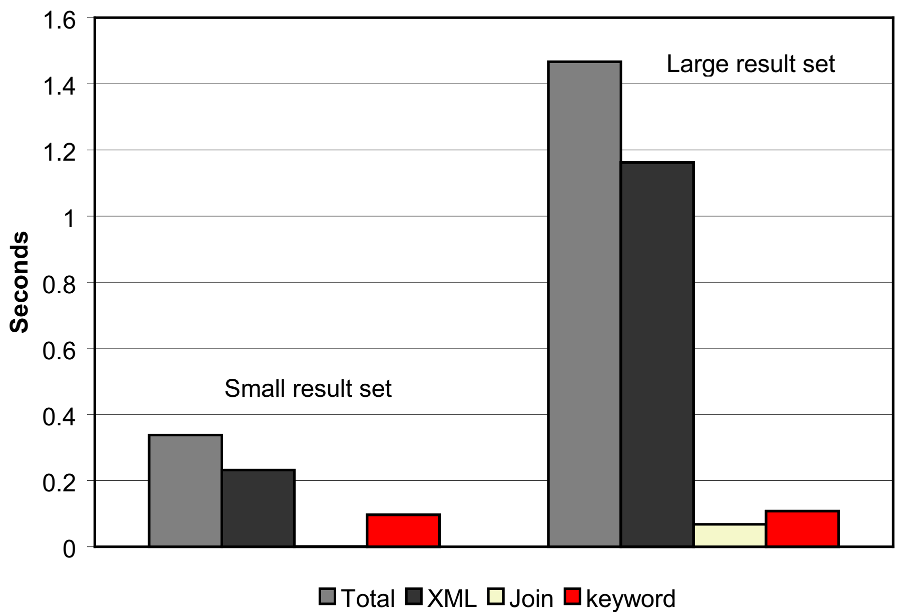

Fig. 4. Hybrid search query processing time

languages of the embedded database in the service peer. A wrapper may be used for converting the delivered query to the internal SQL queries of a database system.

Before describing the format of a search query, we should address whether a unified structure for metadata is necessary. A well-defined unified structure of metadata (*schema*) will make the development of search services simple. Every search peer shares the same schema and the metadata search is only applied to the unified schema. However, this policy will cause some problems. When an additional element in schema—or other schema update—is needed, the cost may be very high. Every search service peer should reflect the new schema and this is hard to guarantee in the peer-to-peer environment, which usually has no central administration. On the other hand, if we do not have any unified schema and structures of the metadata are diverse, metadata search is not much different from the keyword only search. Without the common part of the schema, metadata do not have a role of classification over the documents that contain target keywords. Therefore, such metadata search does not narrow the scope of keyword search.

Our research focuses on particular interest groups that search their mutually shared data. A unified schema of the search group is desirable, but, to give the flexibility in the whole search system, it need not be fixed. Therefore, we designate a recommended definition of metadata, but allow the schema approx-

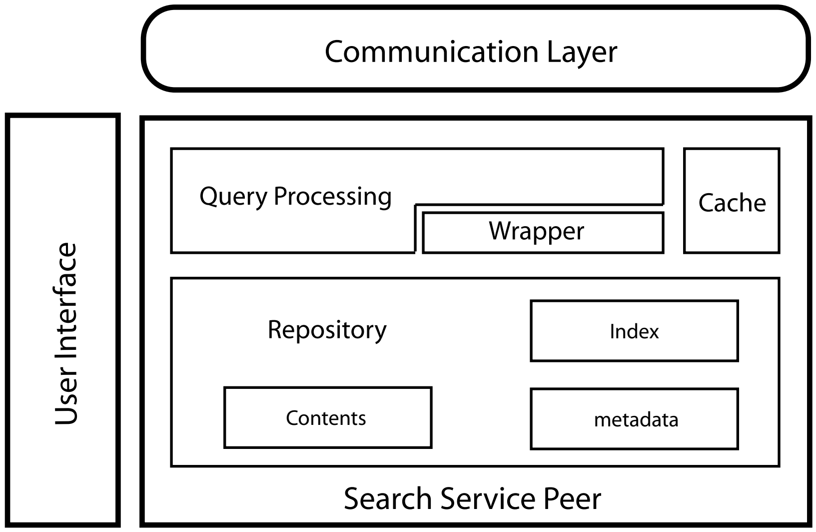

Fig. 5. Structure of the search peer framework

imation. Our approximation deals with some missing or excessive information for the metadata items, but not the theme capabilities. Research on approximate query processing in decision support systems [12, 16, 2] suports the values of partial results from querying on huge data. This research supports the idea that we do not have to insist on an exact match for the metadata query. That is because we focus on a more precise and semantic search than a Web-based keyword search. In addition, it is hard for the query to reach every edge node in the P2P overlay network within a reasonable response time.

XML is a promising description format for the search query as well as for the metadata, because it is a semistructured data model with tags. Any software application can interpret it by common XML program modules. A particular data sharing and search group can assign a recommended schema, and each search service peer should provide at least partial results against the user inquiry. The partial result means the service peers do not have to match all the information elements in their metadata against the user inquiry. Assume, for example, that a user query requests an author name and a year match for an article search. If a search peer has only author but no year information, it can return the query result only for author match. However, if the query matches the author but has a different year, the result should not be returned.

3.2 Communication layer

Information integration between two or more heterogeneous databases is a long term challenge for database system researchers. The heterogeneity of those data causes difficulties in interoperation between database systems. The recent spotlight on the emergence of XML reflects a demand from the information integration society, because XML provides a unified communication format for the information exchange between different machines. But, how to communicate between machines is another matter. To attain the proper search functionality over heterogeneous resources, the following communication features are necessary.

- Query passing.
- Result set delivery.
- Target data retrieval.

The user query should reach the search service through a communication layer. Under the traditional client/server communication model, centralized control would be unavoidable. The requester connects to all the other databases to collect the information (*federated database*), a central database (*data warehouse*) collects all the dispersed data, or a *mediator* responds to the query by retrieving the inquiry data from several databases. All of those models need the information on host locations and literacy with the query processing language of each database.

After the query is passed to a search service, the result set against the inquiry should be generated and returned back to the requester. The client/server communication is obvious for any response from the server, because the client usually stays connected until the communication closes. Pipe based communication—a half-duplex connection—may need additional mechanisms to pass the query response back to the querier.

The result set has the metadata of the inquiry data, and the user may only need selective data from the query results. Therefore, the user should choose the target data to be retrieved from the data providers. The communication layer has to provide the target data delivery to the querier. The integration model based on client/server has centralized control, and all the resource locations and the data transfer are managed by the central system.

The peer-to-peer network paradigm provides an overlay network on top of the Internet, which enables inexpensive scalability and partial match lookup. The protocols of those P2P models meet the above requirements in various manners. JXTA [21] is one of those protocols and the first effort towards a general P2P framework. JXTA provides the pipe communication, peer and resource discovery, and a peer group mechanism. Through its resolver service, a query can be sent, responded to, and even propagated between peers.

3.3 Data integration hub

Every information provider may not want to develop its own search services. Moreover, existence of many peers that have little data will unnecessarily increase the message traffic and reduce the chance to reach the target information. To prevent network flooding, *Gnutella-like* P2P networks usually restrict the number of hops to propagate a message. The supernode architecture in P2P networks integrates the information from partial nodes and dramatically increases the network performance. So, we can improve our architecture by partial

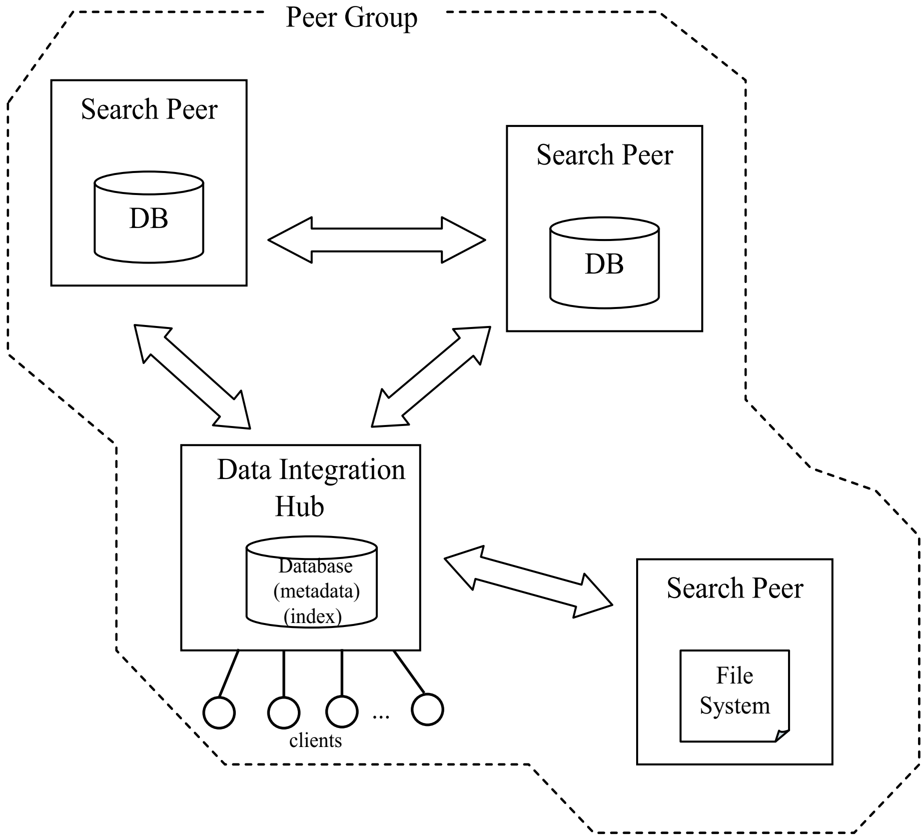

Fig. 6. Peer group architecture of distributed hybrid search

integration using a message-oriented middleware—in our case the NaradaBrokering system [27]. The data integration hub acts as a peer in the whole P2P networks. The figure 6 shows the group communication between search peers.

4 Related Works

There are a number of search studies based on P2P networks. ODISSEA (Open DIStributed Search Engine Architecture) [32] is an example of content search on P2P networks with distributed global index. Its distributing index of object names depends on a Distributed Hash Table (DHT) structure, Pastry [29]. The study of Galanis and others [9] focused on query delivery to the right nodes through distributed directories. It used XPath as a query language, with the XML element names as indexes distributed through the Chord [31] DHT-based protocol. But they are keyword only searches or metadata only searches, while our hybrid search narrows the search categories with combination of both.

Some researchers [17, 7] noticed the utility of gossiping in peer-to-peer networks, as the mechanism for a replicated directory to locate the information. PlanetP [7] uses a gossiping algorithm to share the index and provides pure content search based on a peer-to-peer system. They claim that their gossiping paradigm is appropriate up to thousands of peers on a P2P overlay network. No metadata is assimilated in their work.

PeerDB [24] is a P2P distributed data sharing system that enables content-based search without shared schema. The query propagation depends on an agent in each peer. There is no specific policy, but Gnutella-like query propagation with a TTL value. The authors experimented on a cluster of computers.

Nakauchi and others [22] enlarged the category of keyword search on P2P networks with query expansion. The query expansion is based on a thesaurus keyword relational database with a distributed mechanism for updating. Hristidis and others [14] addressed the previous problems of keyword search in databases by exploiting a context rank for joining relevant tuples. This should be an effective strategy for keyword search on existing relational databases with reducing the lost part of the inherent meaning from database schema. However, our study focuses on keyword search over existing documents with additional metadata attachment to confine the search category.

XRANK [10] focused on search over contents embedded in the XML documents with semantic tags. This study emphasizes document-centric XML, and both semantic metadata and contents are included in the XML. To apply this research for the legacy system like a bunch of text documents, we may need to reorganize the old data to be included in the XML instances. In contrast to our work, this study does not consider the extension of scalability such as the peer-to-peer communication.

5 Conclusion

Searching for needed information on the Internet is difficult, because the information is dispersed, and it is hard to integrate the heterogeneous resources we find there. Our hybrid search on P2P networks addresses the semantic problem inherent in the keyword search while remaining a simpler solution than the Semantic Web. Our research also provides low cost scalability over heterogeneous resources through customized overlay networks. This framework may have a practical bridging role for the information search, on the road towards the ideal of information represented by the Semantic Web [4]. The benefit of our system is the information sharing for mutual interest groups over the Internet, where the information is distributed for the scalability. However, information quality and reputation is another matter to be solved by agents.

References

1. S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A System for Keyword-Based Search over Relational Databases. In *Proceedings of the International Conference on Data Engineering(ICDE)*, February 2002.
2. B. Babcock, S. Chaudhuri, and G. Das. Dynamic sample selection for approximate query processing. In *Proceedings of ACM SIGMOD Conference*, June 2003.
3. S. Banerjee, V. Krishnamurthy, M. Krishnaprasad, and R. Murthy. Oracle8i—The XML Enabled Data Management System. In *Proceedings of the International Conference on Data Engineering(ICDE)*, February 2000.
4. T. Berners-Lee, J. Hendler, and O. Lassila. The Semantic Web. *Scientific American*, 2001.
5. S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of International World-Wide Web Conference*, April 1998.
6. J. Cheng and J. Xu. XML and DB2. In *Proceedings of the International Conference on Data Engineering(ICDE)*, February 2000.
7. F. Cuenca-Acuna, C. Peery, R. Martin, and T. Nguyen. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing. In *Proceedings of IEEE International Symposium on High Performance Distributed Computing(HPDC)*, June 2003.
8. P. Dixon. Basics of Oracle Text Retrieval. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, December 2001.
9. L. Galanis, Y. Wang, S. Jeffery, and D. DeWitt. Locating Data Sources in Large Distributed Systems. In *Proceedings of the International Conference on Very Large Data Bases(VLDB)*, September 2003.
10. L. Guo, F. Shao, C. Botev, and J. Shanmugasundaram. XRANK: Ranked Keyword Search over XML Documents. In *Proceedings of ACM SIGMOD Conference*, June 2003.
11. J. Hamilton and T. Nayak. Microsoft SQL Server Full-Text Search. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, December 2001.
12. J. Hellerstein, P. Haas, and H. Wang. Online Aggregation. In *Proceedings of ACM SIGMOD Conference*, May 1997.
13. W. Hersh, C. Buckley, T. Leone, and D. Hickam. OHSUMED: An interactive retrieval evaluation and new large test collection for research. In *Proceedings of the 17th Annual ACM SIGIR Conference*, 1994.
14. V. Hristidis, L. Gravano, and Y. Papakonstantinou. Efficient IR-Style Keyword Search over Relational Databases. In *Proceedings of the International Conference on Very Large Data Bases(VLDB)*, September 2003.
15. A. Hulgeri, G. Bhalotia, C. Nakhe, and S. Chakrabarti. Keyword Search in Databases. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, September 2001.
16. Y. Ioannidis and V. Poosala. Histogram-Based Approximation of Set-Valued Query-Answers. In *Proceedings of the International Conference on Very Large Data Bases(VLDB)*, September 1999.
17. H. Johansen and D. Johansen. Improving Object Search using Hints, Gossip, and Supernodes. In *Proceedings of IEEE Symposium on Reliable Distributed Systems (SRDS)*, October 2002.
18. J. Kim, O. Balsoy, M. Pierce, and G. Fox. Design of a Hybrid Search in the Online Knowledge Center. In *Proceedings of the IASTED Internation Conference on Information and Knowledge Sharing*, November 2002.

19. M. Ley. Computer Science Bibliography. W. W. W. http://www.informatik.uni-trier.de/ ley/db/.
20. A. Maier and D. Simmen. DB2 Optimization in Support of Full Text Search. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, December 2001.
21. Sun Microsystems. The JXTA project and Peer-to-Peer Technology. World Wide Web. http://www.jxta.org/.
22. K. Nakauchi, Y. Ishikawa, H. Morikawa, and T. Aoyama. Peer-to-Peer Keyword Search Using Keword Relationship. In *Proceedings of International Workshop on Global and Peer-to-Peer Computing (GP2PC)*, May 2003.
23. Napster. Napster. W.W.W. http://www.napster.com/.
24. W. Ng, B. Ooi, K. Tan, and A. Zhou. PeerDB: A P2P-based System for Distributed Data Sharing. In *Proceedings of the International Conference on Data Engineering(ICDE)*, March 2003.
25. NPAC. Northeeast parallel architectures center. World Wide Web, 2000. http://www.npac.syr.edu.
26. Oracle Corporation. *Oracle9i Application Developers Guide—XML*, June 2001.
27. S. Pallickara and G. C. Fox. NaradaBrokering: A Distributed Middleware Framework and Architecture for Enabling Durable Peer-to-Peer Grids. In *Proceedings of International Middleware Conference*, June 2003.
28. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A Scalable Content-Addressable Network. In *Proceedings of ACM SIGCOMM Conference*, August 2001.
29. A. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms(Middleware)*, November 2001.
30. SETI@Home. SETI@Home. W.W.W. http://setiathome.ssl.berkeley.edu/.
31. I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A Scalable Peertopeer Lookup Service for Internet Applications. In *Proceedings of ACM SIGCOMM Conference*, August 2001.
32. T. Suel, C. Mathur, J. 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. In *Proceedings of the Workshop on the Web and Databases*, June 2003.
33. B. Zhao, J. Kubiatowicz, and A. Joseph. Tapestry: An infrastructure for fault-tolerant wide-area location and routing. Technical report, U.C. Berkeley, April 2001.