P2P Web Search: Make It Light, Make It Fly

Matthias Bender, Tom Crecelius, Sebastian Michel, Josiane Xavier Parreira
Max-Planck-Institut für Informatik
Saarbrücken, Germany
{mbender, tcrecel, smichel, jparreir}@mpi-inf.mpg.de

ABSTRACT

We propose a live demonstration of MinervaLight, a P2P Web search engine. MinervaLight combines the (previously separate) focused crawler BINGO! (to harvest Web data), the local search engine TopX, and our P2P Web search system MINERVA under one common user interface. The crawler unattendedly downloads and indexes Web data, where the scope of the *focused* crawl can be tailored to the thematic interest profile of the user. The result of this process is a local search index, which is used by TopX to evaluate user queries.

In the background, MinervaLight continuously computes compact statistical synopses that describe a user's local search index and publishes that information to a conceptually global, but physically fully decentralized directory. MinervaLight offers a search interface where users can submit queries to MINERVA. Sophisticated query routing strategies are used to identify the most promising peers for each query based on the statistical synopses in the directory. The query is forwarded to those judiciously chosen peers and evaluated based on their local indexes. These results are sent back to the query initiator and merged into a single result list. We give a live demonstration of the fully functional system.

1. INTRODUCTION

Peer-to-Peer (P2P) systems have been a hot topic in various research communities over the last few years. Many prototype systems have been implemented, but hardly any of them has been deployed beyond the scope of medium-scale dedicated playgrounds (e.g., PlanetLab [19]). Not surprisingly, when the mass media reports on the campaign of P2P systems today, they typically still showcase early-day P2P file sharing systems, like Gnutella or Kazaa, which have gained the connotation of largely distributing illegal content.

In order for an innovative P2P prototype system with a research background to actually be deployed by non-expert users on a larger scale, we feel the following three conditions need to coincide:

- **Added value:** users must expect substantial improvements in a frequent task.
- **Easy deployment:** the software must not enforce any (hardware or software) requirements which are not fulfilled by off-the-shelf preinstalled computers. It must be easy to install and run the software for non-expert users.
- **Easy usage:** the software must integrate seemlessly into existing usage patterns, i.e., it should not require the users to significantly change their current habits.

One of the most frequent tasks performed by computer users today is searching the Web. The fact that "to google" has lately become a synonym for searching the Web is not only due to the fact that Google has been able to deliver superior search results, but largely also due to the fact that Google offers a hassle-free user interface that allows non-experts to easily submit their queries - which immediately relates to our above requirements.

As the characteristics of a P2P system (unlimited scalability, resilience to network failures and dynamics) offer enormous potential benefits for data management systems in general and Web search applications in particular, distributed information retrieval systems have been a hot research topic. While centralized search indexes are fundamentally limited in their coverage of the Web and in the freshness of the information they provide, using the combined power and knowledge of millions of users and their PCs makes it possible to provide search results which could not yet have been updated or indexed at all by a centralized service provider. Also beyond centralized Web search engines, a P2P Web search engine can benefit from the intellectual input (e.g., bookmarks, query logs, etc.) of a large user community [5, 13, 12], as every peer in the network is operated by a human user. Finally, but perhaps even most importantly, a P2P Web search engine can also facilitate pluralism in informing users about Internet content, which is crucial in order to preclude the formation of information-resource monopolies, the biased visibility of content from economically powerful sources, and politically motivated censorship.

2. RELATED WORK

Distributed data management systems (and P2P Web search in particular) have been a hot research topic that has brought forward many prototype systems [25, 9, 17, 8, 24, 11, 7, 20, 27]. We have previously demonstrated our P2P Web search engine prototype MINERVA [2, 16].

3rd Biennial Conference on Innovative Data Systems Research (CIDR) January 7-10, 2007, Asilomar, California, USA.

Every time we gave a demonstration, the audience was amazed and asked for the prototype software. But, MINERVA is a *research prototype* that was originally designed and implemented as an internal tool to evaluate the strategies we develop for query routing and result merging in the course of our research on P2P Web search. Its user interface with countless parameters was not simple enough for non-expert users; also the fact that focused crawling and indexing (BINGO!, [6]), local query execution (TopX [26]), and distributed searching required three separate applications with an incompatible look-and-feel discouraged the users. Additionally, each application relied on a full-fledge database system (with incompatible schemata) and required a number of (not always intuitive) parameter settings to work properly. Distributing this stack of software and getting it to run smoothly on various system platforms turned out to be a troublesome task.

To overcome these problems and to ease the deployment of the fruits of our research, we have started a sister project coined *MinervaLight*, which we deliberately design for ease of use. The most notable innovation from a user's perspective is the fact that MinervaLight now supports the complete data lifecycle of crawling and indexing Web data, creating and disseminating statistical synopses, and executing queries globally under one common user interface. Also, instead of relying on an external database system, it contains the open-source database engine Cloudscape/Derby[1] which is collectively used by all components and does not require any installation. Only the most important and intuitive system parameters of MinervaLight need to be configured by the user; for many other parameters, we have chosen appropriate default values. Thus, the barrier of using the software has dramatically been reduced. Nevertheless, MinervaLight can instantly benefit from all of MINERVA's research results, including overlap aware query routing [15], correlation-aware query routing [3], or result merging based on globally compatible scores[1].

MINERVA was originally layered on top of a home-brewed re-implementation of Chord [23], which worked fine in the controlled settings of our lab experiments. In real-world deployments of MINERVA, however, we often ran into system issues, e.g., caused by firewalls or strange IP configurations. Instead of reinventing the wheel, MinervaLight now relies on the mature FreePastry P2P overlay network [22], which additionally takes care of the directory replication necessary in order to deal with network dynamics and failures in a distributed system and can further improve the user-perceived latencies using caching techniques.

3. SYSTEM ARCHITECTURE

MinervaLight combines the following building blocks under one common graphical user interface. It encapsulates BINGO! [6], a focused Web crawler that mimics a human user browsing the Web by only indexing documents that are thematically related to a predefined set of user interests. BINGO! is a multi-language parser, i.e., it can detect the language of documents and restrict the crawl to documents of a language of choice. BINGO! learns the user interest profile by running a feature analysis over the bookmarks that it can import from the user's Web browser. Within the user's interest, BINGO! can further classify the documents it indexes into predefined and automatically trained categories. Alternatively, BINGO! can instantaneously start a high-performing, multi-threaded Web crawl from a set of interactively entered URLs. Crawling is continuously performed in the background, without manual user interaction. BINGO! automatically parses and indexes all applicable content types (currently text, html, and pdf) to build a local search index from these documents. It utilizes stemming and stopword elimination. The search index (in form of inverted index lists) is stored in the embedded Cloudscape/Derby database. Different score values are computed without any user interaction, to support ranked retrieval queries. In order to support more sophisticated document scoring models, BINGO! can compute link-based authority scores (PageRank, HITS) on its local Web graph. Non-expert users can easily inspect which pages have been indexed and browse the pages by category, using MinervaLight's intuitive user interface.

TopX [26] is a search engine for ranked retrieval of XML (and plain-text) data, developed at the Max-Planck Institute for Informatics. TopX supports a probabilistic-IR scoring model for full-text content conditions (including phrases, boolean expressions, negations, and proximity constraints) and tag-term combinations, path conditions for all XPath axes as exact or relaxable constraints, and ontology-based relaxation of terms and tag names as similarity conditions for ranked retrieval. For speeding up top-k queries, various techniques are employed: probabilistic models as efficient score predictors for a variant of the threshold algorithm, judicious scheduling of sequential accesses for scanning index lists and random accesses to compute full scores, incremental merging of index lists for on-demand, self-tuning query expansion, and a suite of specifically designed, precomputed indexes to evaluate structural path conditions.

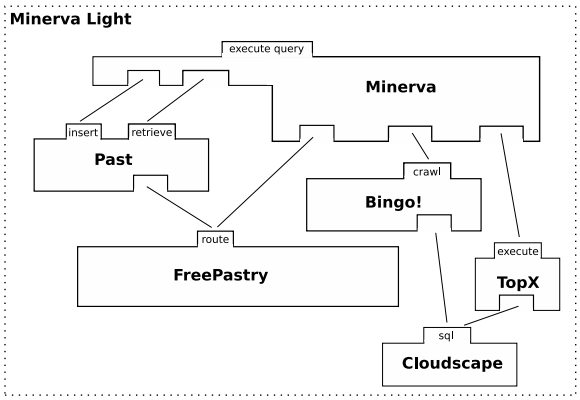

Figure 1: MinervaLight System Architecture

MinervaLight continuously monitors the local search index and computes compact statistical synopses that describe the quality of the index w.r.t. particular terms. These synopses contain statistical information about the local search index, such as the size of the index, the number of distinct terms in the index, the number of documents containing a particular term, and optionally elaborate estimators for score distributions, based on histograms or Poisson mixes. MinervaLight publishes that information into a fully distributed directory, effectively building a *term* → *peer* directory, mapping terms to the set of corresponding syn-

[1]http://db.apache.org/derby/

opses published by peers from across the network. This directory is significantly smaller than naively distributing a full-fledge *term → document* index, which eventually makes P2P Web search feasible [10]. In order to further limit the size of the directory, each peer can determine whether it is a valuable source of information for a particular term, and only publish synopses for terms if it is considered a valuable resource for that term [4]. The publishing process can also be extended beyond individual terms to also account for popular key sets or phrases [14]. The directory implementation is based on Past [21], a freely available implementation of a distributed hash table (DHT). It uses FreePastry's *route* primitive to support the two hash table functionalities (*insert(key,value)* and *value←retrieve(key)*). MinervaLight passes *(term,synopsis)*-tuples to Past, which transparently stores it at the node in the network that is currently responsible for the key *term*. For this purpose, we have extended Past with bulk insertion functionality, in order to send batches of statistical synopses instead of sending them individually, greatly reducing the incurred network overhead. Each *directory node* maintains a list of all incoming synopses for a randomized subset of keys; this metadata is additionally replicated to ensure availability in the presence of network dynamics.

Figure 1 illustrates and summarizes MinervaLight's system architecture.

MinervaLight offers a simple search interface that allows a user to enter query terms, which (transparently to the user) starts the global query execution using Past as follows: for each term appearing in the query, MinervaLight executes *retrieve(term)* to retrieve all applicable synopses from the directory, which serve as the input to *query routing*, i.e., selecting a small subset of promising peers that are most likely to provide high-quality results for a particular query. MinervaLight uses FreePastry's *route* primitive to send the user query to these selected peers, which evaluate the query using their local TopX engines on top of their local indexes and return their top-matching URLs to the query initiator. MinervaLight appropriately combines the URLs from these autonomous sources (*result merging*) and displays the results to the user. After inspecting the results, the user can simply click on any result to open the document in the default Web browser.

In the spirit of social tagging communities, users can manually add arbitrary attribute-value-style annotations by means of a single mouse click. For example, users might rate documents with annotations like *rating=5*. Additional annotations can automatically be generated from the content data, such as *author=bender* or *conference=CIDR*. These annotations are also indexed and become part of the directory metadata, i.e., users can explicitly query for documents with *rating=5* and even combine this with regular query terms.

We have lately developed the JXP algorithm to efficiently compute PageRank scores in a distributed environment of autonomous peers with overlapping local indexes [18]. As PageRank has repeatedly been shown to improve the user-perceived result quality, the incorporation of JXP into MinervaLight is expected to increase the result quality beyond what has so far been achieved with other existing approaches solely based on statistical synopses or based on PageRank scores derived from the local partitions of the Web graph at each peer individually. Preliminary experimental results in the paper referenced above support this hypothesis.

4. DEMO DESCRIPTION

We give a live demonstration of our P2P Web search system MinervaLight, showcasing the complete lifecycle of P2P Web search. After the demo session, we invite all interested attendees to explore the features of MinervaLight in more detail, using their own notebooks.

4.1 Setup

Crawling, indexing, and local search could in principle be demonstrated by a stand-alone system. We could also show a movie recorded in our lab. However, we do not consider any of these alternatives a convincing demonstration of a P2P Web search prototype. Instead, in order to show the full beauty and convenience of MinervaLight, we will set up a live network of MinervaLight instances running on our notebooks PCs, creating a miniature P2P network. After the session, a wifi router will enable visitors to connect their notebooks to our network as well. We will distribute the software using a USB memory stick. In case there is no outbound internet connection available, one of our notebook computers will also run a local Web server with a dump of Wikipedia documents, so that users can nevertheless crawl a wide variety of documents "on-line".

4.2 Crawling and Indexing

The user can import bookmark files containing URLs that are used as crawl seeds. The seed URLs are downloaded beforehand and used in order to build a classifier that can later-on be used to categorize crawled documents into different fields of interest. When this *training phase* has been completed, a simple click on the *Crawl*-button will start the Web crawl. If desired, users can set additional crawling options, such as limiting the crawling to a certain domain or restricting the crawl depth. The document download and the indexing of all applicable content (currently text, html, pdf) proceeds automatically, without any further user interaction required. The user can monitor the progress of the crawl by means of a continuously updated table that features a list of the most recently crawled URLs, including their page titles and categorizations (cf. Figure 2). The user can also start an illustration of the crawled Web graph and start the computation of authority-scores based on the links between the documents (PageRank, HITS). The user can also stop the Web crawl at any point in time.

4.3 Building and Maintaining Directory

MinervaLight starts to compute statistical synopses describing the local index harvested from the Web crawl and publishes that metadata in the distributed directory. For this purpose, a new overlay network can be instantiated, or an existing network can be joined by means of a bootstrap node. For illustrative purposes, MinervaLight also offers a way of inspecting the portion of metadata that has been received, i.e., the local "share" of the global directory.

4.4 Annotating Documents

Users can annotate documents with arbitrary attribute-value pairs, e.g., describing the quality of a document or providing bibliographic metadata, such as *conference=CIDR*. These annotations are also aggregated and published into the distributed directory. Currently, two query modes are supported: retrieving all documents with a certain annotation, and retrieving all annotations for a given document.

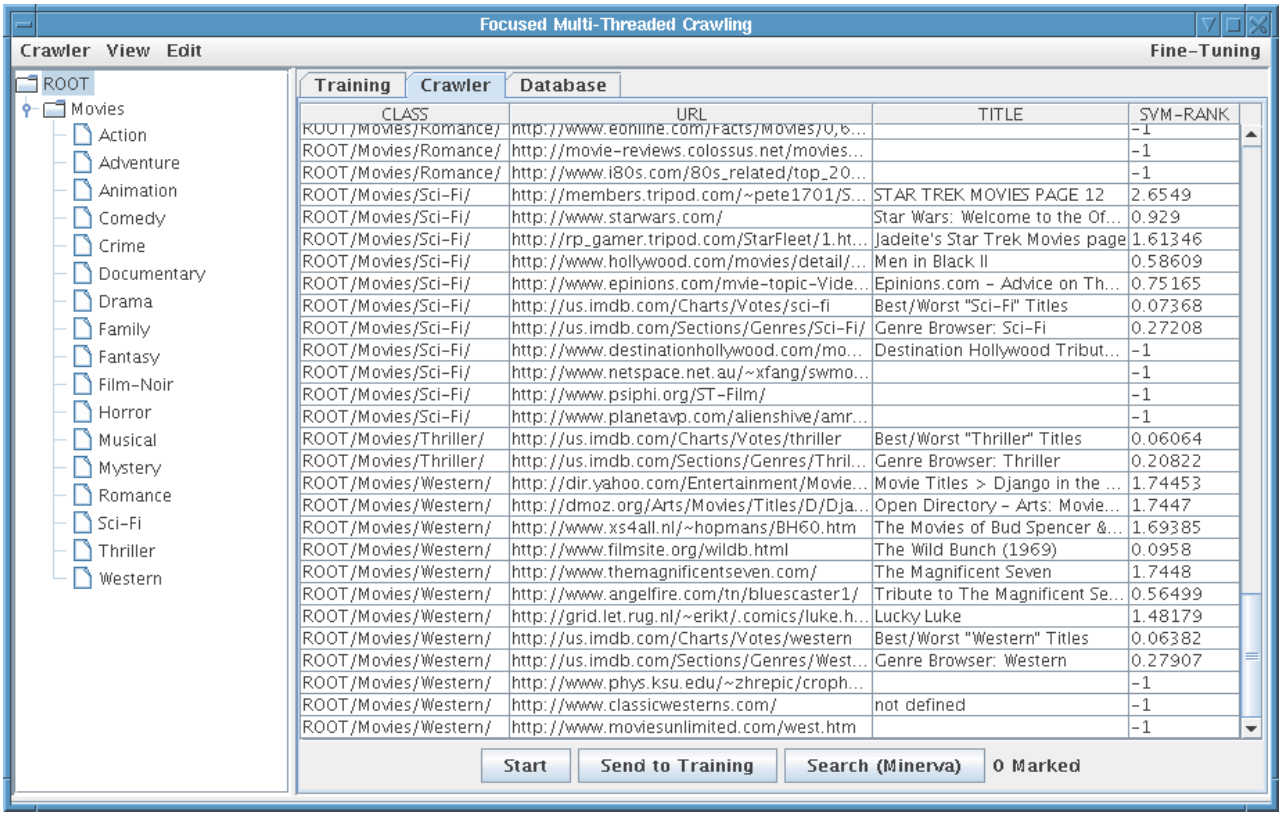

Figure 2: MinervaLight Crawling Interface

4.5 Querying the network

The user can enter keyword queries, e.g., *abraham*, into an intuitive search interface (cf. Figure 3). This triggers MinervaLight to retrieve all applicable metadata that is relevant for this query from the directory and to select the most promising peers for the query (*query routing*). Users can optionally use the metadata inspection functionality introduced above to verify the decision of the system. The query is forwarded to the selected peers and executed locally. Each peer indicates all such incoming queries, providing another way of intuitively following the progress of a query. After each of these peers has returned its local results to the query initiator, the results are displayed in form of a single merged result list that features the URL, the page title, and other information about the document. Users can inspect this list and open the documents in their Web browsers by means of a mouse click.

In order to query annotations, e.g., to retrieve documents that were annotated as *rating=5*, the user can simply enter this query *rating=5* into the same form field that has also been used for keyword queries. Again, MinervaLight will identify applicable documents based on synopses it retrieves from the distributed directory.

5. REFERENCES

[1] M. Bender, S. Michel, P. Triantafillou, and G. Weikum. Global document frequency estimation in peer-to-peer web search. In *WebDB*, 2006.

[2] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Minerva: Collaborative p2p search. In *VLDB*, 2005.

[3] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. P2p content search: Give the web back to the people. In *IPTPS*, 2006.

[4] M. Bender, S. Michel, and G. Weikum. P2p directories for distributed web search: From each according to his ability, to each according to his needs. In *WIRI*, 2006.

[5] M. Bender, S. Michel, C. Zimmer, and G. Weikum. Bookmark-driven query routing in peer-to-peer Web search. In *P2PIR*, 2004.

[6] Bookmark-Induced Gathering of Information with Adaptive Classification into Personalized Ontologies. http://www.mpi-sb.mpg.de/units/ag5/software/bingo/.

[7] A. Crespo and H. Garcia-Molina. Routing indices for peer-to-peer systems. In *ICDCS*, 2002.

[8] 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. In *HPDC*, 2003.

[9] R. Huebsch, B. N. Chun, J. M. Hellerstein, B. T. Loo, P. Maniatis, T. Roscoe, S. Shenker, I. Stoica, and A. R. Yumerefendi. The architecture of pier: an internet-scale query processor. In *CIDR*, 2005.

[10] J. Li, B. Loo, J. Hellerstein, M. Kaashoek, D. Karger, and R. Morris On the Feasibility of Peer-to-Peer Web Indexing and Search. In *IPTPS*, 2003.

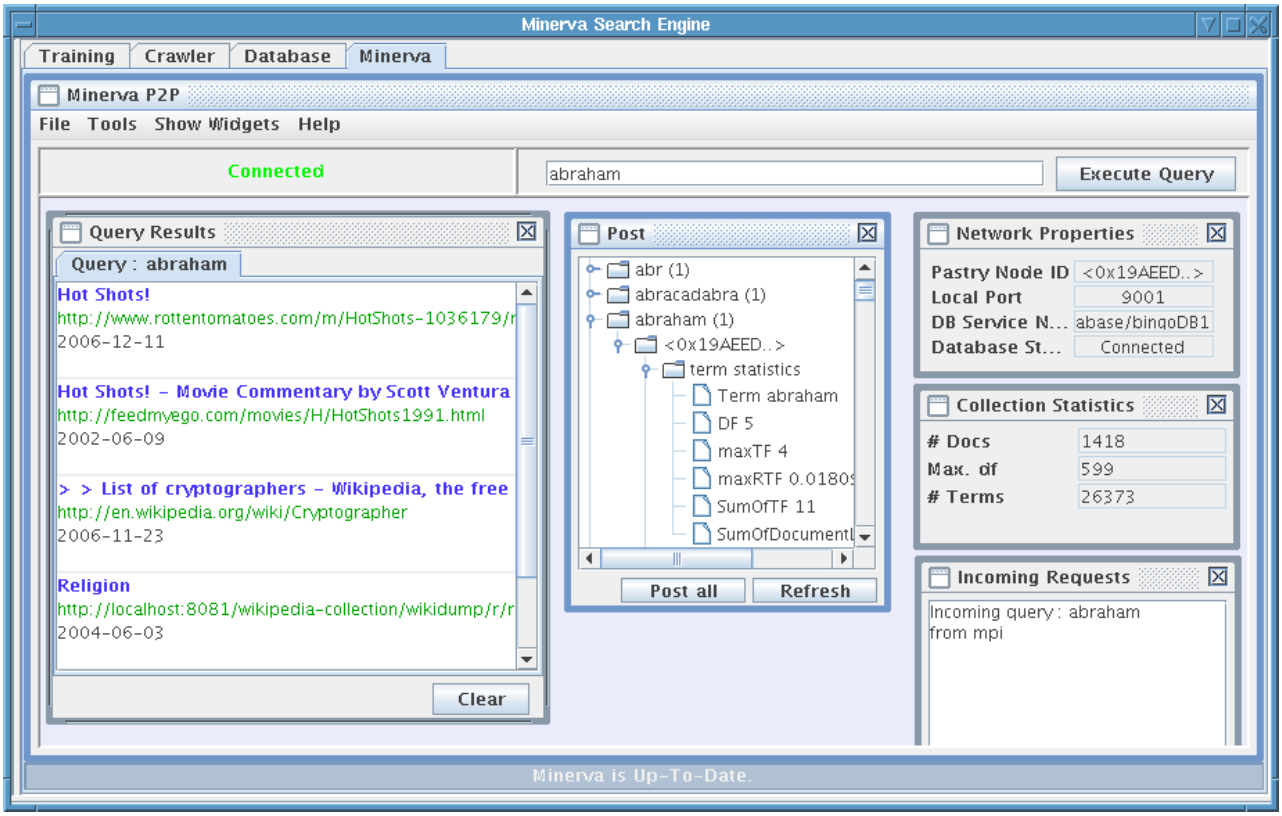

Figure 3: MinervaLight Search Interface

[11] J. Lu and J. Callan. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *ECIR*, 2005.

[12] J. Luxenburger and G. Weikum. Query-log based authority analysis for web information search. In *WISE*, 2004.

[13] J. Luxenburger and G. Weikum. Exploiting community behavior for enhanced link analysis and web search. In *WebDB 2006*, 2006.

[14] S. Michel, M. Bender, N. Ntarmos, P. Triantafillou, G. Weikum, and C. Zimmer. Discovering and exploiting keyword and attribute-value co-occurrences to improve P2P routing indices. In *CIKM)*, 2006.

[15] S. Michel, M. Bender, P. Triantafillou, and G. Weikum. Iqn routing: Integrating quality and novelty in p2p querying and ranking. In *EDBT*, 2006.

[16] S. Michel, M. Bender, P. Triantafillou, G. Weikum, and C. Zimmer. P2P web search with MINERVA: How do you want to search tomorrow? (demo). In *Middleware*, 2005.

[17] H. Nottelmann, G. Fischer, A. Titarenko, and A. Nurzenski. An integrated approach for searching and browsing in heterogeneous peer-to-peer networks. In *HDIR*, 2005.

[18] J. X. Parreira, D. Donato, S. Michel, and G. Weikum. Efficient and decentralized pagerank approximation in a peer-to-peer web search network. In *VLDB*, 2006.

[19] L. L. Peterson and T. Roscoe. The design principles of planetlab. *Operating Systems Review*, 40(1):11–16, 2006.

[20] P. Reynolds and A. Vahdat. Efficient peer-to-peer keyword searching. In *Middleware*, 2003.

[21] A. Rowstron and P. Druschel. Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In *Middleware*, 2001.

[22] A. I. T. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *Middleware*, 2001.

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

[24] T. Suel, C. Mathur, J. wen Wu, J. Zhang, A. Delis, M. Kharrazi, X. Long, and K. Shanmugasundaram. Odissea: A peer-to-peer architecture for scalable web search and information retrieval. In *WebDB*, 2003.

[25] C. Tang and S. Dwarkadas. Hybrid global-local indexing for efficient peer-to-peer information retrieval. In *NSDI*, 2004.

[26] M. Theobald, R. Schenkel, and G. Weikum. An efficient and versatile query engine for topX search. In *VLDB)*, 2005.

[27] Y. Wang, L. Galanis, and D. J. de Witt. Galanx: An efficient peer-to-peer search engine system. *Available at http://www.cs.wisc.edu/ yuanwang.*