MINERVA: Collaborative P2P Search

Matthias Bender[†], Sebastian Michel[†], Peter Triantafillou[‡], Gerhard Weikum[†], Christian Zimmer[†]

[†]Max-Planck-Institut für Informatik
66123 Saarbrücken, Germany
{mbender, smichel, weikum, czimmer}@mpi-sb.mpg.de

[‡] University of Patras
Rio, 26500, Greece
peter@ceid.upatras.gr

Abstract

This paper proposes the live demonstration of a prototype of MINERVA[1], a novel P2P Web search engine. The search engine is layered on top of a DHT-based overlay network that connects an a-priori unlimited number of peers, each of which maintains a personal local database and a local search facility. Each peer posts a small amount of metadata to a physically distributed directory that is used to efficiently select promising peers from across the peer population that can best locally execute a query. The proposed demonstration serves as a proof of concept for P2P Web search by deploying the project on standard notebook PCs and also invites everybody to join the network by instantly installing a small piece of software from a USB memory stick.

1 Introduction

The peer-to-peer (P2P) approach allows handling huge amounts of data in a distributed and self-organizing way. These characteristics offer enormous potential benefit for search capabilities powerful in terms of scalability, efficiency, and resilience to failures and dynamics. Additionally, such a search engine can potentially benefit from the intellectual input (e.g., bookmarks, query logs, click streams, etc.) of a large user community.

The original architectures of structured P2P networks are typically limited to exact-match queries on keys. More recently, the data management community has focused on extending such architectures to support more complex queries [11, 23, 14, 10]. All this related work, however, is insufficient for text queries that consist of a variable number of keywords, and it is absolutely inappropriate for full-fledged Web search where keyword queries should return a ranked result list of the most relevant approximate matches [7].

The crucial challenge in developing successful P2P Web search engines is based on reconciling the following conflicting goals: on the one hand, delivering high quality results with respect to precision / recall, and, on the other hand, providing unlimited scalability in the presence of a very huge peer population and the very large amounts of data that must be communicated in order to meet the first goal. We put forward MINERVA whose architecture, design, and implementation satisfies these conflicting goals. Novel aspects of the MINERVA architecture are the way we leverage DHT-based overlay networks for efficiently managing compact, aggregated information that peers publish about their local indexes and the way we use these metadata to appropriately select promising peers in order to limit the number of peers involved in a query. This keeps the global DHT-based directory manageable and reduces network traffic. We expect MINERVA to scale very well as more and more peers jointly maintain the moderately growing DHT-based directory and present first experimental evidence. We have implemented a fully operational system based on our home-brewed implementation of a Chord-style DHT [19]. We consider the quality of search results, the system overhead, and the run-time efficiency of the system conducting experiments on real-world web data that we harvest from our own extensive focused web crawls. Challenging research aspects of MINERVA are the development of query routing strategies for our dynamic environment (cf. Section 4) and the evaluation and consolidation of search results returned from individual, autonomous peers.

The demonstration illustrates our system by giving a live demo of the complete querying process: we assume peers that have crawled thematically focused portions of the web based on their own personal preferences and that are willing to share parts of their knowledge at their own discretion. After joining the P2P network, every peer can pose arbitrary keyword queries. The system identifies promising peers within the network for these queries. Carefully selected peers execute the queries leveraging their local indexes and forward their

Proceedings of the 31st VLDB Conference, Trondheim, Norway, 2005

[1]Roman goddess of science, wisdom, and learning. Project homepage available at http://www.minerva-project.org

local results to the query initiator, where the local results are combined to form one global result set. Our demonstration invites everybody to join our network instantly by installing a small piece of software from a USB memory stick.

2 Related Work

Recent research on structured P2P systems, such as Chord [19], CAN [15], or Pastry [17] is typically based on various forms of distributed hash tables (DHTs) and supports mappings from keys to locations in a decentralized manner such that routing scales well with the number of peers in the system.

In the following we briefly discuss some prior and ongoing projects towards P2P Web search.

Galanx [24] is a P2P search engine implemented using the Apache HTTP server and BerkeleyDB. The Web site servers are the peers of this architecture; pages are stored only where they originate from. In contrast, our approach leaves it to the peers to what extent they want to crawl interesting fractions of the Web and build their own local indexes.

PlanetP [8] is a publish-subscribe service for P2P communities, supporting content ranking search. PlanetP distinguishes local indexes and a global index to describe all peers and their shared information. The global index is replicated using a gossiping algorithm. The system appears to be limited to a few thousand peers.

Odissea [20] assumes a two-layered search engine architecture with a global index structure distributed over the nodes in the system. A single node holds the complete, Web-scale, index for a given text term (i.e., keyword or word stem). Query execution uses a distributed version of Fagin's threshold algorithm [9]. The system appears to cause high network traffic when posting document metadata into the network, and the presented query execution method seems limited to queries with at most two keywords. The paper actually advocates using a limited number of nodes, in the spirit of a server farm.

The system outlined in [16] uses a fully distributed inverted text index, in which every participant is responsible for a specific subset of terms and manages the respective index structures. Particular emphasis is put on minimizing the bandwidth used during multi-keyword searches. eSearch [21] is a P2P keyword search system based on a hybrid indexing structure. eSearch selects a small number of important terms in a document and publishes the complete term list for the document to nodes responsible for those top terms. [12] considers content-based retrieval in hybrid P2P networks where a peer can either be a simple node or a directory node. Directory nodes serve as super-peers, which may possibly limit the scalability and self-organization of the overall system. The peer selection for forwarding queries is based on the Kullback-Leibler divergence between peer-specific statistical models of term distributions.

In addition to this recent work on P2P Web search, prior research on distributed IR and metasearch engines is potentially relevant, too; see [6, 25] for overviews. However, this work has assumed a relatively small number of databases and a fairly static configuration.

3 System Design

We assume that every peer is autonomous and has a local index that can be built from the peer's own crawls or imported from external sources and tailored to the user's thematic interest profile. The index contains inverted lists with URLs for Web pages that contain specific keywords or terms in IR jargon.

Our conceptually global but physically distributed directory, which is layered on top of a distributed hash table (DHT), holds only very compact, aggregated meta-information about the peers' local indexes and only to the extent that the individual peers are willing to disclose. We use a Chord-style DHT to partition the term space, such that every peer is responsible for the meta-information of a randomized subset of terms within the global directory. For failure resilience and availability, the entry for a term may be replicated across multiple peers. The DHT offers a *lookup* method to determine the peer responsible for a particular term.

First, every peer publishes per-term summaries (*Posts*) of its local index to the directory. The DHT determines the peer currently responsible for this term. This peer maintains a *PeerList* of all postings for this term from across the network. Posts contain contact information about the peer who posted this summary together with statistics to calculate IR-style measures for a term (e.g., the size of the inverted list for the term, the maximum average score among the term's inverted list entries, or some other statistical measure). These statistics are used to support the query process, i.e., determining the most promising peers for a particular query.

The querying process for a multi-term query proceeds as follows: first, the query is executed locally using the peer's local index. If the result is considered unsatisfactory by the user, the querying peer retrieves a list of potentially useful peers by issuing a *PeerList request* for each query term to the underlying overlay network. A number of promising peers for the complete query is computed from these PeerLists. This step is referred to as *query routing*. The challenges of this process and possible approaches are introduced in section 4. Subsequently, the query is forwarded to these peers and executed based on their local indexes using a cutting-edge probabilistic TA-sorted algorithm ([22]). This communication is done in a pairwise point-to-point manner between the peers, allowing for efficient communication and limiting the load on the global directory. Finally, the results from the various peers are combined at the querying peer into a single result list; this step is referred to as *result merging*.

4 Query Routing

Database selection has been a research topic for many years, e.g. in distributed IR and metasearch [6]. Typically, the expected result quality of a collection is estimated using precomputed statistics, and the collections are ranked accordingly. Most of these approaches, however, are not directly applicable in a true P2P environment, as

- the number of peers in the system is substantially higher (10^x peers as opposed to 10-20 databases)
- the system evolves dynamically, i.e. peers enter or leave the system autonomously at their own discretion at a potentially high rate
- the results from remote peers should not only be of high quality, but also complementary to the results previously obtained from one's local search engine or other remote peers

In [2, 3], we have adopted a number of popular existing approaches to fit the requirements of such an environment and conducted extensive experiments in order to evaluate the performance of these naive approaches.

As a second step, we have extended these strategies using estimators of mutual overlap among collections [1].

In Minerva, all queries are first processed by the query initiator itself on the locally available index. We expect that many queries will be answered this way without incurring any network costs. But when the user is not satisfied with the query result, the query will be forwarded to a small number of promising peers. In this situation selecting those peers merely on the basis of their data quality, like size of indexed data or freshness and authority of the data, and the "semantic" or statistics-based similarity to the thematic profile of the query initiator does not work well. We may often end up choosing remote peers that, albeit having high-quality data, do not provide additional information, for their indexed data may overlap too much with the data that the query originator already searched in its own local index. Thus, overlap-awareness is crucial for cost-beneficial query routing.

Experiments show that such a combination can outperform popular approaches based on quality estimation only, such as CORI [6]. As shown in Figure 1, taking overlap into account when performing query routing can drastically decrease the number of peers that have to be contacted in order to reach a satisfactory level of recall, which is a great step towards the feasibility of distributed P2P search.

We also want to incorporate the fact that every peer has its own local index, e.g., by using implicit-feedback techniques for automated query expansion (e.g., using the well-known IR technique of pseudo relevance feedback [5] or other techniques based on query logs [13] and click streams [18]). For this purpose, we can benefit from the fact that each peer executes the query locally first, and also the fact that each peer represents an actual user with personal preferences and interests. For example, we want to incorporate local user bookmarks into our query routing [3], as bookmarks represent strong recommendations for specific documents. Queries could be exclusively forwarded to thematically related peers with similarly interested users, to improve the chances of finding *subjectively* relevant pages.

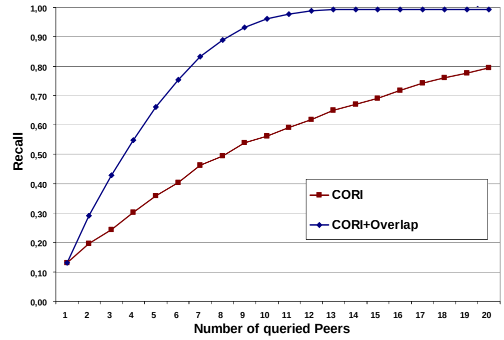

Figure 1: Recall of overlap-aware query routing

Ultimatively, we want to introduce a sophisticated $benefit/cost$ ratio when selecting remote peers for query forwarding. For the benefit estimation, it is intuitive to consider such measures as described in this section. Defining a meaningful cost measure, however, is an even more challenging issue. While there are techniques for observing and inferring network bandwidth or other infrastructural information, expected response times (depending on the current system load) are changing over time. One approach is to create a distributed Quality-of-Service directory that, for example, holds moving averages of recent peer response times.

5 Demonstration

Our demonstration aims at illustrating the whole functionality of our system as well as its ease of use, as we invite all visitors to instantly join our network. Our notebook PCs and all visitors' notebooks are connected via a LAN network and run a number of peers. Each peer in our P2P system represents a personal user with his personal data. To ease the live deployment, visitors can choose from topic-oriented collections in our central database.

One pivotal issue when designing experiments was the absence of a standard benchmark. While there exist a number of benchmark collections for (centralized) Web search, it is not clear how to apply these collections to our scenario.

Our group has performed extensive Web crawls to gather real-world experimental data. In the absence of a standard benchmark for our scenario we demonstrate MINERVA with collections that have been crawled and classified into thematically focused collections using BINGO [4]. Additionally, we have partitioned the .GOV data into 50 overlapping collections and conducted experiments on this data. We are also currently performing experiments on the GOV document collection (cf. http://trec.nist.gov) and on more structured data, as for example taken from the IMDB movie database (http://www.imdb.com).

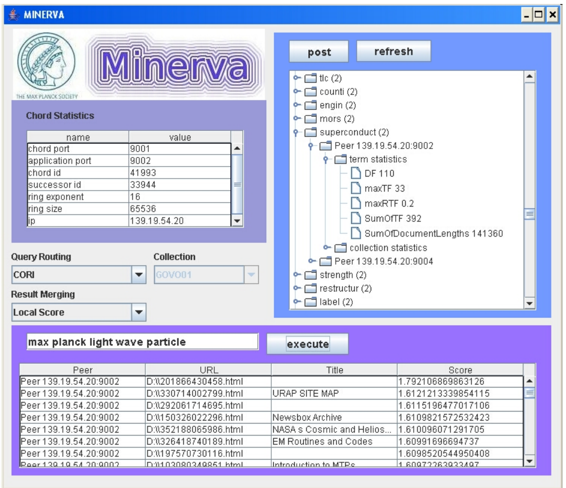

Figure 2: MINERVA GUI

Peers Registering with MINERVA

We present the process of peers registering with MINERVA, i.e. joining the DHT-style directory and posting statistical information about local indexes to the network as explained in Section 3. Afterwards, users can instantly type arbitrary keyword queries into the GUI (Figure 2) of any peer, just like in one of today's popular web search engines.

Query Routing and Processing with MINERVA

The system selects a tuneable number of promising remote peers from the network by gathering the statistical information posted to the directory for each term and subsequently applying query routing strategies (cf. Section 4). The peers that have been selected indicate this fact in real-time in their graphical user interfaces, i.e. the user can interactively see which peers have been selected to answer the query.

Query Result Merging and Display

At the peer initiating the query, the local results returned from each of these peers are merged into one global result list, which is displayed to the user. It indicates which remote peer has delivered the respective results. The user can easily click on the query results to open the original documents in order to verify their relevance to the query.

References

[1] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap awareness in p2p systems. In *Proceedings of the SIGIR Conference*, 2005.

[2] M. Bender, S. Michel, G. Weikum, and C. Zimmer. The MINERVA project: Database selection in the context of P2P search. In *BTW 2005*, Karlsruhe, Germany, 2005.

[3] M. Bender et al. Bookmark-driven query routing in peer-to-peer web search. In *Proceedings of the SIGIR Workshop on P2P-IR*, pages 46–57, 2004.

[4] Bookmark-Induced Gathering of Information. http://www.mpi-sb.mpg.de/units/ag5/software/bingo/.

[5] C. Buckley, G. Salton, and J. Allan. The effect of adding relevance information in a relevance feedback environment. In *SIGIR*. Springer-Verlag, 1994.

[6] J. Callan. Distributed information retrieval. *Advances in information retrieval, Kluwer Academic Publishers.*, pages 127–150, 2000.

[7] S. Chakrabarti. *Mining the Web: Discovering Knowledge from Hypertext Data*. Morgan Kaufmann, San Francisco, 2002.

[8] F. M. Cuenca-Acuna et al. PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities. Technical Report DCS-TR-487, Rutgers University, Sept. 2002.

[9] R. Fagin. Combining fuzzy information from multiple systems. *J. Comput. Syst. Sci.*, 58(1):83–99, 1999.

[10] A. Gupta et al. Meghdoot: Content-based publish/subscribe over p2p networks. In *Middleware*, pages 254–273, 2004.

[11] R. Huebsch et al. Querying the internet with pier. In *VLDB*, pages 321–332, 2003.

[12] J. Lu and J. Callan. Content-based retrieval in hybrid peer-to-peer networks. In *Proceedings of CIKM03*, pages 199–206. ACM Press, 2003.

[13] J. Luxenburger and G. Weikum. Query-log based authority analysis for web information search. In *WISE04*, 2004.

[14] V. Papadimos, D. Maier, and K. Tufte. Distributed query processing and catalogs for peer-to-peer systems. In *CIDR*, 2003.

[15] S. Ratnasamy et al. A scalable content-addressable network. In *Proceedings of ACM SIGCOMM 2001*, pages 161–172. ACM Press, 2001.

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

[17] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In *IFIB/ACM Middleware*, pages 329–350, 2001.

[18] J. Srivastava et al. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.

[19] I. Stoica et al. Chord: A scalable peer-to-peer lookup service for internet applications. In *Proceedings of the ACM SIGCOMM 2001*, pages 149–160, 2001.

[20] T. Suel et al. Odissea: A peer-to-peer architecture for scalable web search and information retrieval. Technical report, Polytechnic Univ., 2003.

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

[22] M. Theobald, G. Weikum, and R. Schenkel. Top-k query evaluation with probabilistic guarantees. *VLDB*, pages 648–659, 2004.

[23] P. Triantafillou and T. Pitoura. Towards a unifying framework for complex query processing over structured peer-to-peer data networks. In *DBISP2P*, pages 169–183, 2003.

[24] 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.*

[25] C. Yu et al. Efficient and effective metasearch for a large number of text databases. In *CIKM99*, 1999.