Efficient Search and Approximate Information Filtering in a Distributed Peer-to-Peer Environment of Digital Libraries

Christian Zimmer, Christos Tryfonopoulos, and Gerhard Weikum

Department for Databases and Information Systems
Max-Planck-Institute for Informatics, 66123 Saarbrücken, Germany
{czimmer,trifon,weikum}@mpi-inf.mpg.de

Abstract. We present a new architecture for efficient search and approximate information filtering in a distributed Peer-to-Peer (P2P) environment of Digital Libraries. The *MinervaLight* search system uses P2P techniques over a structured overlay network to distribute and maintain a directory of peer statistics. Based on the same directory, the *MAPS* information filtering system provides an approximate publish/subscribe functionality by monitoring the most promising digital libraries for publishing appropriate documents regarding a continuous query. In this paper, we discuss our system architecture that combines searching and information filtering abilities. We show the system components of MinervaLight and explain the different facets of an approximate pub/sub system for subscriptions that is high scalable, efficient, and notifies the subscribers about the most interesting publications in the P2P network of digital libraries. We also compare both approaches in terms of common properties and differences to show an overview of search and pub/sub using the same infrastructure.

1 Introduction

Peer-to-Peer (P2P) has been a hot topic in various research communities over the last few years. Today, the P2P approach allows handling huge amounts of data of digital libraries 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. [13, 14]) of a large user community. However, recent research on structured P2P architectures [8, 9] is typically limited to exact-match queries on keys. This 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. In the area of distributed data management, many prototypes have been developed [15, 16, 17] including our P2P Web search engine prototype Minerva [2, 18].

In such a dynamic P2P setting, information filtering (IF) [1, 3, 5, 10], also referred to as publish/subscribe or continuous querying or information push, is equally important to one-time querying, since users are able to subscribe to information sources and be

C. Thanos, F. Borri, and L. Candela (Eds.): Digital Libraries: R&D, LNCS 4877, pp. 328–337, 2007.

notified when documents of interest are published by any digital library. This need for push technologies is also stressed by the deployment of new tools such as Google Alert or the QSR system. In an information filtering scenario, a user posts a subscription (or profile or continuous query) to the system to receive notifications whenever certain events of interest occur, e.g. a document matching the continuous query is added to a digital library.

In this paper, we present the architecture for efficien search and approximate information filtering in a distributed P2P environment of digital libraries. Our MinervaLight search system uses P2P techniques over a structured overlay network to distribute and maintain a directory of peer statistics. Based on the same directory, the MAPS[1] information filtering system provides approximate pub/sub functionality by monitoring the most promising digital libraries for publishing appropriate documents regarding a continuous query.

The paper is organized as follows: In Section 2, we presen the main search architecture of MinervaLight using one-time queries. The pub/sub functionality of MAPS is explained in Section 3. Section 4 compares MinervaLight and MAPS and stresses the common properties of both systems and shows the main differences, and Section 5 concludes the paper.

2 MinervaLight Search Architecture

In MinervaLight, we view every digital library as autonomous and each digital library refers to a peer in the network. MinervaLight [24] combines different building blocks under one common graphical user interface.

2.1 BINGO!

MinervaLight uses *BINGO!* [19], 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, the system 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.

[1] Minerva Approximate Publish/Subscribe.

2.2 TopX

TopX [20] is a search engine for ranked retrieval of XML and plain-text data, that 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.

2.3 Distributed Directory

MinervaLight continuously monitors the local search index and computes compact statistics (called posts) that describe the quality of the index concerning particular terms. These statistics contain 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 to peer directory, mapping terms to the set of corresponding statistics published by digital libraries from across the network. This directory is significantly smaller than naively distributing a full-fledge term to document index, which eventually makes P2P search feasible [21].

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 statistics for terms if it is considered a valuable resource for that term. The publishing process can also be extended beyond individual terms to also account for popular key sets or phrases [6]. The directory implementation is based on Past [22], 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 $retrieve(key)$. A previous version of the Minerva prototypes used Chord [9], another structured overlay network to build-up the distributed directory. The choice of the underlying DHT is not a serious decision since MinervaLight (as MAPS) is network agnostic.

MinervaLight passes *(term, post)* pairs to the DHT, which transparently stores it at the peer in the network that is currently responsible for the key term. For this purpose, we have extended the DHT 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 peer 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.

2.4 P2P Search and Ranking

MinervaLight offers a simple search interface that allows a user to enter query terms, which starts the global query execution using the DHT as follows: for each term appearing in the query, MinervaLight executes retrieve(term) to retrieve all applicable post lists from the directory, which serve as the input to query routing, i.e., selecting a small subset of promising digital libraries that are most likely to provide high-quality results for a particular query. MinervaLight uses the DHT route primitive to send the user query to these selected digital libraries, which evaluate the query using their local TopX engines on top of their local indexes and return their top-matching results to the query initiator. MinervaLight appropriately combines the URLs from these autonomous sources (result merging) and returns the final results to the user.

Lately, the JXP algorithm [23] to efficiently compute PageRank scores in a distributed environment of autonomous peers with overlapping local indexes is integrated into MinervaLight. 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 statistics 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.

3 MAPS Information Filtering Architecture

In this section, we present the main system architecture of MAPS based on the P2P search engine Minerva. Each peer or digital library that participates in MAPS implements three types of services: a publication, a subscription, and a directory service.

A peer implementing the publication service of MAPS has a (thematically focused) web crawler and acts as an information producer. The publication service is used to expose content crawled by the peer's crawler and also content published by the peer's user to the rest of the network. Using the subscription Service users post continuous queries to the network and this service is also responsible for selecting the appropriate peers that will index the user query. Finally, the directory service is used to enable the peer to participate in the P2P network, and is also responsible for acquiring the IR statistics needed by the subscription service to perform the ranking.

3.1 Publishing of Resources

Publications in a peer or digital library p occur when new documents are made available to the rest of the network. Each publication is matched against its local query index using appropriate local filtering algorithms, and triggers notifications to subscriber peers. Notice that only peers with their continuous query indexed in p will be notified about the new publication, since the document is not distributed to any other peer in the network. This makes the placement of a peer's continuous query a crucial decision, since only the peers storing the query can be monitored for new publications, and the publication and notification process does not need any additional communication costs.

3.2 Selecting Appropriate Publishers

When a peer p receives a continuous query q from the user, p has to determine which peers or digital libraries in the network are promising candidates to satisfy the continuous query with similar documents published in the future. To do so, p issues a request to the directory service for each term contained in q, to receive per-peer statistics about each one of the terms. Statistics from the retrieved lists are gathered and a peer score is computed based on a combination of resource selection and peer behavior prediction formulas as shown by the equation below.

$$score(p, q) = (1 - \alpha) \cdot sel(p, q) + \alpha \cdot pred(p, q)$$

The tunable parameter α affects the balance between authorities (digital libraries with high $sel(p, q)$ score) and promising peers (peers with high $pred(p, q)$ score) in the final ranking. Finally, based on the total score calculated for each peer a ranking of peers is determined, and q is forwarded to the first k peers in the list, where k is a user specified parameter. The continuous query is then stored in these peers, and a notification is sent to the user every time one of the peers publishes a document that matches the query.

A continuous query needs to get updated after a specific time period. For this reason, a query contains a time-to-live (ttl) value such that the peer holding the query can remove it after the ttl is expired. The peer initiating the continuous query process requests new statistics from the directory and reselects the updated most promising peers for q.

3.3 Resource Selection

The function $sel(p, q)$ returns a score for a peer or digital library p and a query q, and is calculated using standard resource selection algorithms from the IR literature (such as simple tf-idf, CORI etc. [4,7]). Using $sel(p, q)$ we can identify authorities specialised in a topic, but as we show later this is not enough in a filtering setting. In our experimental evaluation we use a simple but efficient approach based on the peer document frequency (df) as the number of documents in the peer collection containing a term, and the maximum peer term frequency (tf^{max}) as the maximum number of term occurrences in the documents of the digital library.

3.4 Peer Behavior Prediction

Function $pred(p, q)$ returns a score for a peer p and a query q that represents how likely peer p is to publish documents containing terms found in q in the future. This prediction mechanism is based on statistical analysis of appropriate IR metrics such as the document frequency of a term. These statistics are made available through appropriate requests form the directory service, and are treated as time series data. Then an appropriate smoothing technique is used to model peer behavior and predict future publications. In our prototype implementation, we use the evolution of the peer document frequency (df) to predict a number of documents in the next period containing a certain term, and we use the progression of the collection size (cs) to predict the publishing rate. The values for all terms of the multi-term query are again summarized. The publishing of relevant documents is more accented than the dampened publishing rate.

The main idea behind predicting peer behavior or publishing behavior of digital libraries is to view the IR statistics as time series data and use statistical analysis tools to model peer behavior. Time series analysis accounts for the fact that the data points taken over time have some sort of internal structure (e.g., trend, periodicity etc.), and uses this observation to analyse older values and predict future ones. In our context this hypothesis is valid; a digital library currently crawling publishing many documents about soccer is likely to publish documents about soccer also in the future.

There are many different techniques to predict future values: moving average techniques can not cope well with trends in the data values and assign equal weights to past observations. Since both weaknesses are critical in our scenario, we use the second group of techniques, exponential smoothing techniques. We have chosen double exponential smoothing as the most appropriate method to model a peer's behavior and to predict publication activity in the future. Double exponential smoothing considers trends in contrast to single exponential smoothing. For an application with many long-lasting queries, one could use triple exponential smoothing, so that seasonality is taken into account.

3.5 Why Prediction is Necessary

A key component of the peer selection procedure is the prediction mechanism introduced here. Prediction is complementary to resource selection and the following example demonstrates its necessity in a filtering setting:

Assume that a digital library dl_1 is specialised in soccer, and thus it has become an authority in articles about soccer, although it is not publishing new documents any more. Contrary, digital library dl_2 is not specialised in soccer but currently it publishes documents concerning soccer. Now imagine a user subscribing for documents with the continuous query soccer world cup 2010 to be held in four years in South Africa. A ranking function based only on resource selection algorithms would always choose digital library dl_1 to index the user query. To get a high ranking score, and thus get selected for indexing the user profile, digital library dl_2 would have to specialise in soccer, a long procedure that is inapplicable in a filtering setting which is by definition dynamic. The fact that resource selection alone is not sufficient is even more evident when news items are published. News items have a short shelf-life, making them the worst candidate for slow-paced resource selection algorithms. The above shows the need to include better reactions in slow-paced selection algorithms, to cope with dynamics.

3.6 Directory Maintenance

As shown before, accurate per-peer statistics are necessary for the peer ranking and selection process. The MAPS system uses the same directory as Minerva to maintain the IR statistics. A conceptually global but physically distributed directory, which is layered on top of a Chord-style distributed hash table (DHT), manages aggregated information about each digital library in compact form. This way, we use the Chord DHT to partition the term space, such that every peer is responsible for the statistics of a randomized subset of terms within the directory. To maintain the IR statistics up-to-date, each one distributes per-term summaries (*posts*) of its local index along with its contact information to the global directory. The DHT determines a peer currently responsible for this term and this peer maintains a list of all posts for this term.

Notice that our architecture is network-agnostic. The directory service implemented by the peers does not have to use Chord, or any other DHT to provide this information;

our architecture allows for the usage of any type of P2P network (structured or unstructured), given that the necessary information (i.e., the per-peer IR statistics) is made available to the rest of the services. Thus, unstructured networks with gossip-based protocols, hierarchical networks where the super-peers collect this type of information as well as any structured overlay can implement the directory service. Nevertheless, user the same infrastructure as the Minerva search engine enables to save message costs.

3.7 Experimental Results

First experiments with MAPS have shown promising results. We used different publishing scenarios to investigate the system properties in terms of recall as the ratio of received notifications and the number of relevant published documents. We also considered a benefit/cost ratio to investigate the number of filtering messages needed to get one notification. In comparison to existing exact information filtering approaches [11, 12], MAPS enhances the scalability of P2P information filtering. Especially, by exploiting peer specialization, we can achieve high recall even when indexing queries to a small fraction of publishing digital libraries.

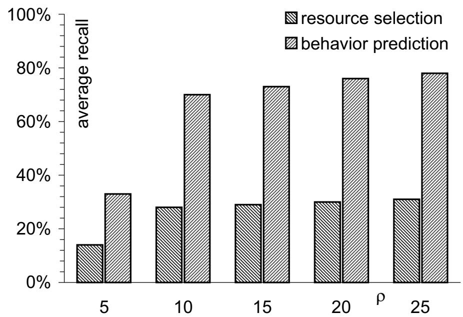

Fig. 1. Experimental Results for MAPS using Resource Selection and Behavior Prediction

Figure 1 shows the results of an experimental evaluation using 1,000 publishers and 30 two-, three- and four-term continuous queries. The peers are specialized in one of the ten categories (e.g., *Sports*, *Music* etc.). The chart shows the filtering effectiveness in terms of recall (ratio of total number of received notifications to the total number of published documents matching a continuous query) where each peer publishes 30 documents within a specific time period and we investigate the results after ten publishing rounds. We vary the percentage ρ of publisher peers that store the continuous query. Figure 1 illustrates that the use of behavior prediction improves recall over resource selection as it manages to model more accurately the publishing behavior of peers.

4 Architectural Comparison

MinervaLight and MAPS are two different systems dealing with different issues, but both approaches have several properties in common. In this section, we provide a

	MinervaLight	MAPS
Directory	Both use the same infrastructure to store statistical information of peers. This index forms a conceptually global, but physically distributed directory with term-peer statistics layered on top of a DHT.	
Overlay network (DHT)	Past DHT, using FreePastry's routing primitives; other DHT implementations possible.	Chord DHT; other DHT implementations possible (Pastry, CAN, Tapestry…)
Query Character	▪ one-time queries ▪ multi-term queries ▪ other queries possible	▪ continuous queries (subscriptions) ▪ multi-term queries ▪ other queries possible
Main Problem	**Query Routing:** MinervaLight uses the directory to select the most promising peers to answer a one-time user query.	**Query Assignment:** MAPS uses the peer statistics to select the most promising peers on which a continuous query subscription is placed.
	Peer Selection: In both approaches, the selection of the most promising peers is the most critical system decision at query run-time.	
Peer Selection	MinervaLight uses the current statistics to apply resource selection methods (e.g., CORI, GlOSS, etc.).	MAPS combines resource selection methods with peer behavior prediction based on time-series analyses.
Extensions to improve Peer Selection	MinervaLight uses knowledge about correlated terms, and overlap-awareness to improve the peer selection step. Distributed PageRank improves result quality.	MAPS can be extended to consider term correlations. Other extensions are future work.
Retrieval Measurements	**Relative Recall** as the percentage of top-k results of a centralized search engine.	**Recall** as the ratio of received notifications to the total number of relevant documents published in the network
Scalability	MinervaLight scales well by only sending a query to a set of promising peers.	MAPS improves scalability of P2P information filtering in contrast to exact information filtering system.

Fig. 2. Salient features and Comparison of MinervaLight and MAPS

comparison considering architectural issues. The following table 2 surveys salient system properties of MinervaLight and MAPS in a side-by-side fashion:

5 Conclusion

In this paper, we presented a system architecture for efficient search and approximate information filtering in a distributed P2P environment of digital libraries. P2P techniques are used to build a global directory of peer statistics. This directory is used for peer selection in two scenarios: searching and approximate information filtering. To build-up the directory, the architecture can use different underlying P2P system such as Chord or Pastry.

We compared MinervaLight and MAPS in terms of various system properties: MinervaLight is a distributed P2P system to search one-time queries in a network of digital libraries. The approximate information filtering MAPS system uses the same infrastructure as MinervaLight including the distributed directory with term to peer statistics. In MAPS, peer selection determines the most promising digital library peers to publish documents of interest in the future to satisfy a continuous query. The selection process combines well-known resource selection techniques with new peer behavior prediction approaches.

References

1. Bell, T., Moffat, A.: The Design of a High Performance Information Filtering System. In: SIGIR (1996)
2. Bender, M., Michel, S., Triantafillou, P., Weikum, G., Zimmer, C.: Improving Collection Selection with overlap-awareness. In: SIGIR (2005)
3. Callan, J.: Document Filtering With Inference Networks. In: SIGIR (1996)
4. Callan, J.: Distributed Information Retrieval. Kluwer Academic Publishers, Dordrecht (2000)
5. Hull, D., Pedersen, J., Schuetze, H.: Method Combination for Document Filtering. In: SIGIR (1996)
6. Michel, S., Bender, M., Ntarmos, N., Triantafillou, P., Weikum, G., Zimmer, C.: Discovering and Exploiting Keyword and Attribute-Value Co-Occurrences to Improve P2P Routing Indices. In: CIKM (2006)
7. Nottelmann, H., Fuhr, N.: Evaluating Different Methods of Estimating Retrieval Quality for Resource Selection. In: SIGIR (2003)
8. Ratnasamy, S., Francis, P., Handley, M., Karp, R.M., Shenker, S.: A scalable Content-Addressable Network. In: SIGCOMM (2001)
9. Stoica, I., Morris, R., Karger, D.R., Kaashoek, M.F., Balakrishnan, H.: Chord: A scalable Peer-to-Peer Lookup Service for Internet Applications. In: SIGCOMM (2001)
10. Tang, C., Xu, Z.: pFilter: Global Information Filtering and Dissemination using structured Overlay Networks. In: FTDCS (2003)
11. Tryfonopoulos, C., Idreos, S., Koubarakis, M.: LibraRing: An Architecture for Distributed Digital Libraries Based on DHTs. In: Rauber, A., Christodoulakis, S., Tjoa, A.M. (eds.) ECDL 2005. LNCS, vol. 3652, Springer, Heidelberg (2005)
12. Tryfonopoulos, C., Idreos, S., Koubarakis, M.: Publish/subscribe functionality in IR environments using structured overlay networks. In: SIGIR (2005)
13. Bender, M., Michel, S., Zimmer, C., Weikum, G.: Bookmark-driven query routing in Peer-to-Peer Web Search. In: P2PIR (2004)
14. Luxenburger, J., Weikum, G.: Exploiting community behavior for enhanced link analysis and Web search. In: Zhou, X., Su, S., Papazoglou, M.M.P., Orlowska, M.E., Jeffery, K.G. (eds.) WISE 2004. LNCS, vol. 3306, Springer, Heidelberg (2004)
15. Crespo, A., Garcia-Molina, H.: Routing indices for Peer-to-Peer systems. In: ICDCS (2002)
16. Cuenca-Acuna, F.M., Peery, C., Martin, R.P., Nguyen, T.D.: PlanetP: Using gossiping to build content addressable Peer-to-Peer Information sharing communities. In: HPDC (2003)
17. Huebsch, R., Chun, B.N., Hellerstein, J.M., Loo, B.T., Maniatis, P., Roscoe, T., Shenker, S., Stoica, I., Yumerefendi, A.R.: The architecture of Pier: an internet-scale query processor. In: CIDR (2005)
18. Bender, M., Michel, S., Triantafillou, P., Weikum, G., Zimmer, C.: Minerva: Collaborative P2P Search. In: VLDB (2005)

19. Bookmark-Induced Gathering of Information with Adaptive Classification into Personalized Ontologies. http://www.mpi-inf.mpg.de/units/ag5/software/bingo/
20. Theobald, M., Schenkel, R., Weikum, G.: An efficient and versatile query engine for topX search. In: VLDB (2005)
21. Li, J., Loo, B., Hellerstein, J., Kaashoek, M., Karger, D., Morris, R.: On the Feasibility of Peer-to-Peer Web Indexing and Search. In: Kaashoek, M.F., Stoica, I. (eds.) IPTPS 2003. LNCS, vol. 2735, Springer, Heidelberg (2003)
22. Rowstron, A., Druschel, P.: Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems. In: Guerraoui, R. (ed.) Middleware 2001. LNCS, vol. 2218, Springer, Heidelberg (2001)
23. Parreira, J.X., Donato, D., Michel, S., Weikum, G.: Efficient and decentralized PageRank approximation in a Peer-to-Peer Web search network. In: VLDB (2006)
24. Bender, M., Crecelius, T., Michel, S., Parreira, J.X.: P2P Web Search: Make it Light, Make it Fly (Demo). In: CIDR (2007)