MAPS*: Approximate Publish/Subscribe Functionality in Peer-to-Peer Networks

Klaus Berberich[1] **Manolis Koubarakis**[2] **Christos Tryfonopoulos**[1]

Gerhard Weikum[1] **Christian Zimmer**[1]

[1] Max Planck Institute for Informatics
Department of Databases and Information Systems, 66123 Saarbruecken, Germany
{kberberi, trifon, weikum, czimmer}@mpi-inf.mpg.de

[2] National and Kapodistrian University of Athens
Department of Informatics and Telecommunications, GR715784 Athens, Greece
koubarak@di.uoa.gr

ABSTRACT

Information filtering has been a research issue for years. In an information filtering scenario users information needs are expressed by user subscriptions, and users are notified about published documents or events that match these interests. The combination of the publish/subscribe scenario with the peer-to-peer (P2P) approach of autonomous peers makes high demands on the scalability and the efficiency of such a given highly distributed network. However, in many cases a subscriber is not interested in all the events that match his profile, but rather in a small representative set. In this paper, we present our approach of an approximate publish/subscribe system, that relaxes the assumption for receiving notifications from every information producer in the network. Our work builds upon distributed hash table technology to create and maintain a distributed global directory that contains information about peers' publishing behavior and combines the current peer state and the prediction of the future publishing behavior of a peer to store a subscription only to the most promising peers in the network. Our experimental evaluation shows that approximate information filtering results satisfying recall level and is able to accommodate changes in peer publishing behaviour.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval - *information filtering*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems - *distributed applications*

General Terms

Algorithms, Performance

Keywords

publish/subscribe, information filtering, P2P overlay networks, distributed hash tables

*Minerva **A**pproximate **P**ublish/**S**ubscribe

Manolis Koubarakis and Christos Tryfonopoulos were partially supported by project Evergrow.
ADPUC'06 November27-December 1, 2006 Melbourne, Australia

1. INTRODUCTION

Much information of interest to humans is available today on the Web, making it extremely difficult to stay informed without sifting through enormous amounts of information. In such a dynamic setting information filtering, also referred to as publish/subscribe or continuous querying, is equally important to one-time querying, since users are able to subscribe to information sources and be notified when documents of interest are published. The deployment of new tools such as Google Alert or the QSR system [22] underlines this observation. Additionally, the growing popularity of P2P networks and also advances in P2P research allow the handling of huge amounts of data in a distributed and self-organised way. The characteristics of P2P networks allow them to capture Web dynamics and potentially offer benefits in terms of scalability, efficiency and fault-tolerance.

The research problem of information filtering has lately received considerable attention from various communities including researchers from information retrieval (IR), databases, distributed computing, digital libraries, and agent systems [6, 18, 19]. However, most of the approaches taken so far have the underlying hypothesis that the subscriber is interested in receiving the events or documents from all the information producers in the network. Our work described here, puts forward Minerva Approximate Publish/Subscribe (MAPS), a novel architecture to support approximate information filtering functionality in a P2P context. The approximate approach is based on the system architecture of Minerva, a P2P search engine of autonomous peers [2]. In MAPS, a user subscribes with a continuous query and monitors some (namely the most interesting) sources on the network. The user query is replicated to these sources and only published documents from these sources are forwarded

to him. The system is responsible for managing the user query, discover new potential sources and move queries to better ones. This approach resembles centralised approaches currently taken for filtering news items, based on a profile of user preferences [8]. In these approaches, however, the emphasis is on duplicate elimination whereas in our case it is on information quality, scalability, and efficiency.

1.1 Contribution

To achieve approximate information filtering we introduce a network-agnostic architecture, and new peer selection techniques suitable for an information filtering scenario. To this end, we treat per-peer IR-style statistics as time series, and show how to use time series analysis in general and double exponential smoothing in particular to predict peer behavior. Then, we show how to use this prediction in combination with standard database selection to improve peer selection in an information filtering scenario. Our experiments emphasise the potential of the approximate publish/subscribe approach such that we get a high information quality in an efficient and scalable way. The main contribution of our work is the approximate information filtering, and, in addition, the use of time series analysis to improve peer selection.

1.2 Outline

The rest of the paper is structured as follows: Section 2 discusses related work from structured P2P overlay networks, collection selection in distributed IR, and P2P publish/subscribe systems. The MAPS system architecture including all the extensions used for continuous querying is explained in Section 3 together with the time series analysis to predict the future peer behavior. The experimental evaluation of our approach on two datasets is presented in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORK

Recent research on P2P systems has proposed various forms of distributed hash tables (DHTs) like Chord [17], CAN [13], or Pastry [14]. DHTs support mappings from keys (e.g., titles or authors) to locations in a decentralized manner such that routing scales well with n, the number of peers in the system. Typically, an exact match key lookup can be routed to the proper peer(s) in at most $O(log\ n)$ hops, and no peer needs to maintain more than $O(log\ n)$ routing information. These architectures have a high failure resilience and can deal with the high dynamics of a P2P system such as peers joining or leaving at a high rate and in an unpredictable manner. However, the approaches are limited to exact-match, single keyword queries on keys.

In the research area of collection or database selection in distributed IR, many approaches have been proposed including the decision-theoretic framework by Fuhr et al. [10], the GlOSS method [11], and approaches based on statistical language models [16, 21]. Callan [5] gives an overview of algorithms for distributed IR style result merging and database content discovery. In the P2P scenario of autonomous peers with possibly overlapping datasets, collection selection has to be extended to an enhanced peer selection approach considering the novelty as described in [2].

The P2P publish/subscribe system Scribe [15] only supports subject-based subscriptions whereas pFilter [18] is a global-scale, decentralized information filtering and dissemination system for unstructured documents. [20] describes local filtering algorithms using a model based on named attributes. The problem of offering publish/subscribe functionality on top of structured overlay networks using data models and languages from information retrieval is studied in [19]. Dittrich et al [9] present *Context-aware Information Filters* (CIF) using two input streams with messages and context updates. [1] shows a solution for efficiently supporting queries over string-attributes involving prefix, suffix, containment, and equality operators on top of a DHT. The main difference of all these approaches to our work is that our system architecture follows an approximate strategy whereas we are only interested in the best published documents for a subscription. By sending the continuous query only to the most promising peers, our system is scalable and more efficient providing good notifications.

3. MAPS SYSTEM ARCHITECTURE

In this section, we present the main system architecture of MAPS (shown in Figure 1) based on the P2P search engine Minerva [2]. Each peer that participates in MAPS implements three types of services: a *publication*, a *subscription*, and a *directory service*.

A peer implementing the publication service 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 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 Resource Publication

Publications in a peer p occur when new documents are crawled from p's crawler or when p's user decides to make one of the documents residing on his computer available to the rest of the network. Each publication of p is matched against its local query index using appropriate local filtering algorithms such as [20], and triggers notifications to subscribers. 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 publi-

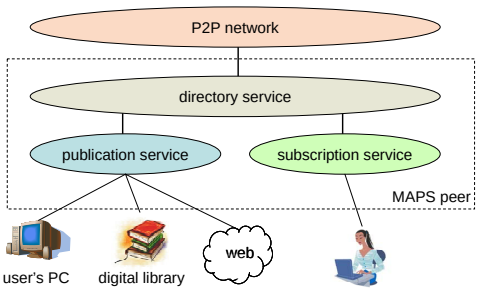

Figure 1: MAPS architecture with publication, subscription, and directory service.

cation and notification process does not need any additional communication costs.

3.2 Forwarding of Continuous Queries

When a peer p receives a continuous query q from the user, p has to determine which peers 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 *database 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 (peers 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.

3.2.1 Database Selection

The function $sel(p,q)$ returns a score for a peer p and a query q, and is calculated using standard database selection algorithms from the IR literature (such as simple tf-idf, CORI etc.). 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 peer. The values for all query terms t are summarizes as follows:

$$sel(p,q) = \sum_{t \in q} \beta \cdot \log\left(df_{p,t}\right) + (1-\beta) \cdot \log\left(tf_{p,t}^{max}\right)$$

The value of the parameter β can be chosen between 0 and 1 and is used to emphasize the importance of df vs. tf^{max}. Experiments with database selection have shown that 0.5 represents a satisfying value for β. [4] and [12] give an overview of other approaches to select peers with different database selection strategies.

3.2.2 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, referred as $\hat{df}^{*}$[1], and we use the progression of the *collection size* (cs) to predict the publishing rate, referred as $\hat{cs}^{*}$. The values for all terms of the multi-term query are again summarized:

$$pred(p,q) = \sum_{t \in q} log\Big(\hat{df}_{p,t}^{*} + \log\left(\hat{cs}_{p}^{*} + 1\right) + 1\Big)$$

The publishing of relevant documents is more accented than the dampened publishing rate. If a peer publishes no documents at all, or, to be exact, the prediction of $\hat{cs}^{*}$ is 0 (and the prediction of $\hat{df}^{*}$ is 0), then the $pred(p,q)$ value should also be 0.

3.3 Why Peer Behavior Prediction?

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

Assume that peer p_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, peer p_2 is a peer that is not specialised in soccer but currently its crawler is crawling pages in a big soccer portal. 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 database selection algorithms would always choose peer p_1 to index the user query. To get a high ranking score, and thus get selected for indexing the user profile, peer p_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 database 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 database selection algorithms. The above shows the need to include better *reactions* in slow-paced selection algorithms, to cope with dynamics.

3.4 Time Series Analysis

The main idea behind predicting peer behavior 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 peer currently crawling a soccer portal, is publishing many documents about soccer. It is likely that this crawler in the near future will also be publishing documents about soccer. [7] gives a good overview about time series analysis and shows the advantages of different approaches.

Smoothing techniques are generally divided into two categories: averaging methods and exponential smoothing methods. Averaging methods include the calculation of the mean of past values, as well as more sophisticated techniques as moving average and double moving average. The major drawbacks of these techniques are the assignment of equal weights to all of the past values, and their inability to reveal

[1]The * denotes that this is a predicted value, and the $\hat{}$ denotes that it is a difference between two values.

trend in a series of observations. In our setting recent values are of greater importance than older values, since peer activity in the near past can provide a better reflection of peer behavior compared to the far past. In contrast to averaging methods exponential smoothing techniques assign exponentially decreasing weight to the values, as the observation gets older. This way, recent observations become more important in predicting than older ones. Single, double and triple exponential smoothing use one or more smoothing parameters to determine weights assigned to the observation values. Single exponential smoothing is usually utilised for non-seasonal data showing no trend. Double exponential smoothing utilises an extra parameter that takes trend into account, whereas triple exponential smoothing is a generalised method to include seasonality into the predicting. In our setting a peer's publishing behavior shows a trend for its future activity, but this is not the case for seasonality. Although seasonality is also existent in user profiles (e.g., consider the frequency of the word *gift* in queries, and how it is increased around Christmas), no seasonality can be observed in the small time intervals we take into account. 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. This approach is incorporated in the prediction formula already mentioned and predicts the two relevant values $\hat{df}^*$ and $\hat{cs}^*$.

Next, we will now shortly explain the double exponential smoothing approach. We assume an infinite series of values $x_1, x_2, x_3, ..,$ and x_n^* denotes the predicted value for x_n after having seen the first $n-1$ values of the series. This predict is computed by the following formula containing two parts:

$$x_n^* = L_n + T_n$$

The L_n describes the current *level* that the series has reached and is initiated by $L_1 = x_1$. The T_n considers the *trend* of the series and has the starting values $T_0 = 0$ and $T_1 = x_2 - x_1$. The values for the level and the trend are computed by the following two equations, where β and γ are parameters (in our experiments we used $\eta = \gamma = 0.5$):

$$L_n = \eta \cdot x_n + (1-\eta) \cdot (L_{n-1} + T_{n-1})$$
$$T_n = \gamma \cdot (L_n - L_{n-1}) + (1-\gamma) \cdot T_{n-1}$$

3.5 IR Statistics Maintenance

As shown before, accurate per-peer statistics are necessary for the peer ranking and selection process. We follow the approach of [2] 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 peers local knowledge 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 peer 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 *PeerList* 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.

4. EXPERIMENTAL EVALUATION

For the experimental evaluation of our approximate information filtering approach, we used two different data collections: *Amazon data collection* containing extracted customer reviews from *Amazon.com* reviewing system, and a *crawled Web data collection* as a result of a focused crawl containing web documents from different categories. Both experimental series make the following assumption: a querying peer issues a set of continuous queries to the network and waits for notifications. The number of peers, a continuous query is sent to, is our parameter to determine the system efficiency. All peers in the system publish documents (order and rate depends on the experimental series) and the querying peer updates the subscription periodically. In our experimental setting our retrieval measurement is *recall*, which is defined as the ratio of received notifications over the number of relevant published documents. Remember that peers publishing a relevant document for a continuous query only send a notification to the querying peer when the initiating peer has registered the query at the publishing peer. A document is relevant for a continuous query if and only if all query terms are contained in the document.

4.1 Amazon Data Collection

The Amazon data collection contains 266,552 reviews of data items from the *Amazon.com* rating system. We used the original categories but grouped them into 30 main categories representing 30 peers. The smallest category contained about 1,000 reviews and the largest one about 50,000. The reviews have a chronological order we used to publish the documents. Each review is assigned to 1.65 categories on average such that if a review is published, on average 1.65 peers publish the review at the same time. We used 250 different queries containing a set of keywords from 2 to 5. The keywords were extracted from the reviews by selecting the most representative keywords of each category. Examples for queries are *culture war*, *civil rights movement american*, or *mechanics physics*. On average, each review contains 56 keywords, and overall, there are 180,024 different terms in the data collection included.

We started the subscription and publication process from when a starting point of about 20,000 published documents.. This way, the peer selection of the most promising peers can already use existing statistics in the global directory. After the first subscription of all 250 continuous queries, the publishing process starts, and after publishing a certain number of reviews, the subscriptions are refreshed. In this experiment, the influence of predicting the peer behavior is marginal because the publishing rate and the thematic directions of the peer are almost constant. Nevertheless, the experiment shows that approximate information filtering provides a good recall value by only selecting a few peers.

Figure 2 shows the results for the case of 20 rounds with 1,000 documents per round published[2]. To reach a recall

[2]We investigated different scenarios with varying the number of rounds and the number of published peers per round,

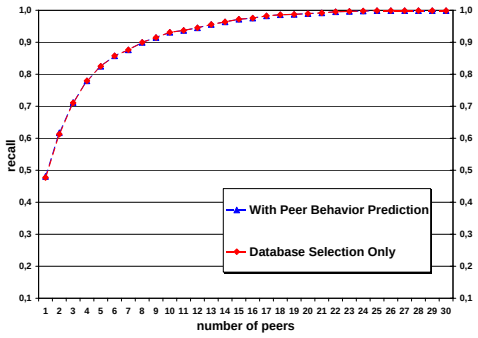

Figure 2: Avg. recall: *Amazon Dataset*

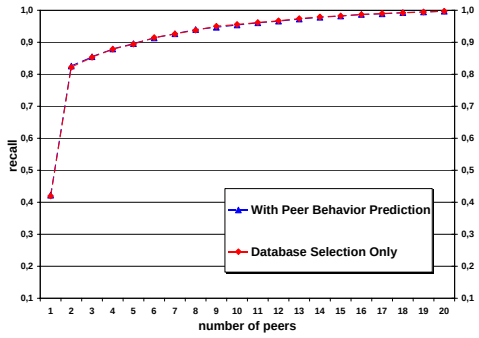

Figure 3: Avg. recall: *Constant Publishing*

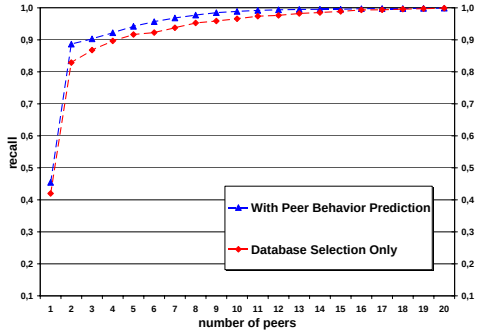

Figure 4: Avg. recall: *Half of the Peers Publishing*

level of 0.8 we need about 4 to 5 peers to send the continuous query to. The gain of sending the query to more than 8-10 peers is very small, and sending the queries to more than 20 peers results in getting notifications to almost all relevant documents.

4.2 Crawled Web Data Collection

This data collection contains 253,875 documents from a focused web crawl. All documents are categorised to one of 10 categories, e.g. *Travel*, *Finance*, or *Sports*. The smallest category has about 18,000 documents, the largest about 35,000 documents. There are more than 700,000 different terms included in all the documents. The documents have no real order such that we use the documents from different categories to simulate certain peer behaviors, e.g. one peer publishes only documents from category *Sports*, and another peer only documents from *Nature*.

We used four different publishing scenarios to investigate the influence of the prediction part. In all cases, we started 20 peers (two peers per category) containing 1,000 random documents from one category each. We extracted 40 two-term queries out of the documents where the query terms are strong representatives of the categories that means that we used terms that appear very frequent in documents of one category and infrequent in documents of the other categories. Example queries are *biology institute*, *pga golf*, or *opera concert*.

All four scenarios used the formula $score(p, q)$ as described before, but in one case, we ignored the prediction part by setting $\alpha = 1.0$. the other case stressed the influence of the peer behavior prediction by using a value of $\alpha = 0.5$.

4.2.1 Constant Publishing

This publishing scenario assumes a constant publishing behavior. Each peer publishes only random documents from the same category as the documents from its starting set. So, every peer publishes 100 documents per round, and at the end of the round the subscriptions are updated. Figure 3 shows the results after 10 rounds. The usage of predicting the peer behavior has almost no effect. This way, the recall values for both cases (with and without peer prediction) are identical. This is caused by the fact that the prediction values correspond to the database selection values because of the constant publishing behavior.

4.2.2 Half of the Peers Publishing

The second simulation assumes that only half of the peers publish any documents at all. So, one peer per category publishes over 10 rounds 100 documents per round. Figure 4 shows that in this scenario, predicting peer behavior improves the recall such that in contrast to the simple database selection approach, we need less peers to reach a satisfying notification level.

4.2.3 Category Changes

In contrast to the first simulation, the peers do not publish documents from their starting set category, but they select another category at random and publish in 10 rounds 100 documents per round out of this category. This simulation stresses that the approach with considering the time series of statistics can respond to category changes and learn that a peer now publishes documents from another category as before. Figure 5 shows the intuition that peer selection benefits from peer behavior prediction. The recall values are lower than in the experiments before because the peer selection needs some time to learn the category changes, but the improvements with peer behavior prediction are remarkable.

4.2.4 Different Publishing Rates

The last scenario simulates different publishing rates. Despite having same sized starting sets, the peers publish documents - from the appropriate category - with different rates. The rate is computed as 100 divided by the peer number such that the first peer publishes 100 documents, the second peer 50 documents, and the last peer releases only 5 documents per round. The results with and without peer behavior prediction after 10 rounds are shown in Figure 6. The gain corresponds to the second scenario where we have improvements when selecting only a few peers, but the results do not reach the improvements of the third simulation.

4.3 Result Discussion

but the results are similar.

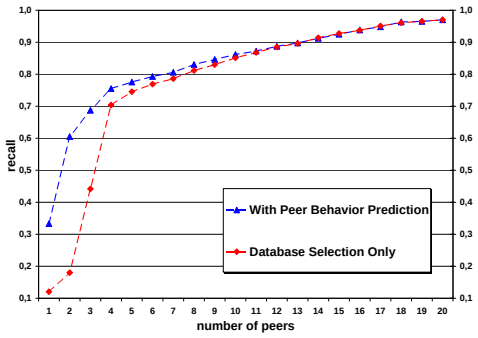

Figure 5: Avg. recall: *Category Changes*

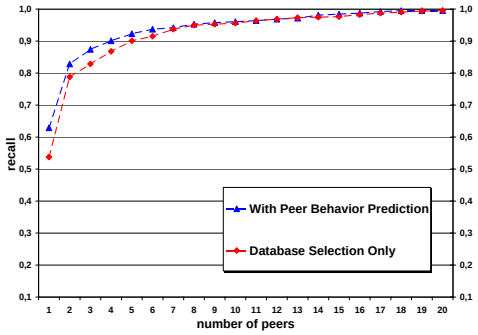

Figure 6: Avg. recall: *Different Publishing Rates*

The experiments point out that approximate information filtering leads to a satisfying recall level by asking only a few peers in the system. If peer behavior change is not significant considering the predicted peer behavior can not further improve the peer selection; but if the peer behavior changes significantly, the results of the time series analysis improve the recall of notifications clearly.

5. CONCLUSIONS AND FUTURE WORK

We have presented a novel approach to approximate information filtering using time series analysis to predict peer behavior. Our efforts concentrate on reducing network traffic, while not sacrificing filtering quality. Using a global directory on top of a distributed P2P overlay network, peer statistics allow the combination of collection selection and peer behavior prediction. This way, a peer subscribing a continuous query selects the most promising peers for this query, and whenever such a promising peer publishes an appropriate document, the subscribing peer gets a notification. Our experiments have shown that approximate information filtering provides satisfactory results, and the necessity to incorporate the predicted peer behavior into the peer selection process. Especially the simulation with thematic changes showed great improvements in contrast to the simple database selection approach.

In future work we plan to explore different prediction approaches and to highlight their behaviour in different peer publication scenarios. We also plan to work on automatic adaptation of prediction parameters in order to further improve the peer selection process. Another direction of ongoing work will consider extensions already proved in the web search scenario, including overlap-awareness [2] and correlation-awareness of terms [3].

6. REFERENCES

[1] I. Aekaterinidis and P. Triantafillou. Internet scale string attribute publish/subscribe data networks. In *CIKM*, 2005.

[2] M. Bender, S. Michel, P. Triantafillou, G. Weikum, and C. Zimmer. Improving collection selection with overlap-awareness. In *SIGIR*, 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, G. Weikum, and C. Zimmer. The minerva project: Database selection in the context of p2p search. In *BTW*, 2005.

[5] J. Callan. Distributed information retrieval., 2000.

[6] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *TOCS*, 2001.

[7] C. Chatfield. *The Analysis of Time Series - An Introduction*. CRC Press, 2004.

[8] G. M. D. Corso, A. Gulli, and F. Romani. Ranking a stream of news. In *WWW*, 2005.

[9] J.-P. Dittrich, P. M. Fischer, and D. Kossmann. Agile: Adaptive indexing for context-aware information filters. In *SIGMOD*, 2005.

[10] N. Fuhr. A decision-theoretic approach to database selection in networked ir. *TOCS*, 1999.

[11] L. Gravano, H. Garcia-Molina, and A. Tomasic. Gloss: Text-source discovery over the internet. *TODS*, 1999.

[12] H. Nottelmann and N. Fuhr. Evaluating different methods of estimating retrieval quality for resource selection. In *SIGIR*, 2003.

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

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

[15] A. I. T. Rowstron, A.-M. Kermarrec, M. Castro, and P. Druschel. Scribe: The design of a large-scale event notification infrastructure. In *NGC*, 2001.

[16] L. Si, R. Jin, J. P. Callan, and P. Ogilvie. A language modeling framework for resource selection and results merging. In *CIKM*, 2002.

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

[18] C. Tang and Z. Xu. pfilter: Global information filtering and dissemination using structured overlay networks. In *FTDCS*, 2003.

[19] C. Tryfonopoulos, S. Idreos, and M. Koubarakis. Publish/subscribe functionality in ir environments using structured overlay networks. In *SIGIR*, 2005.

[20] C. Tryfonopoulos, M. Koubarakis, and Y. Drougas. Filtering algorithms for information retrieval models with named attributes and proximity operators. In *SIGIR*, 2004.

[21] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *SIGIR*, 1999.

[22] B. Yang and G. Jeh. Retroactive answering of search queries. In *WWW*, 2006.