Cooperative Pull-Push Cycle for Searching a Hybrid P2P Network

Melanie Gnasa, Sascha Alda, Nadir Gül, Jasmin Grigull, and Armin B. Cremers
Institute of Computer Science III
University of Bonn
Bonn, Germany
Email: {gnasa, alda, guel, grigull, abc}@iai.uni-bonn.de

Abstract

Information acquisition is a great challenge in the context of a continually growing Web. Nowadays, large Web search engines are primarily designed to assist an information pull by the user. On this platform, only actual information needs are handled without assistance of long-term needs. To overcome these shortcomings we propose a cooperative system for information pull and push on a peer-to-peer architecture. In this paper we present a hybrid network for a collaborative search environment, based on a local personalization strategy on each peer, and a highly-available Web search service (e.g. Google). Each peer participates in the Pull-Push Cycle, and has the function of an information consumer as well as an information provider. Hence, long-term information needs can be identified without any context restrictions, and recommendations are computed based on Virtual Knowledge Communities.

1. Introduction

Nowadays, many Web users rely on autonomous systems, which take over a huge amount of complex tasks. One of these tasks is the provision of information from the Web. The data set is very large and heterogeneous with respect to content, structure, and quality. Classical Web search engines provide multiple results to single requests, and often, only a small fraction of Web sites is relevant to the users' information need. In addition to these major difficulties, two general retrieval strategies can be divided: (1) *information pull* and (2) *information push*. According to Cheverst and Smith [6] information pull is any flow of information that occurs as a result of a conscious initiation by the user. For the selection of relevant information the search result is explicitly analyzed by the user with respect to the actual information need. Lately, Web personalization strategies have been developed to handle this situation. These can be defined as actions to adapt information or services provided by a Web site to the needs of an individual user [8]. For this task, navigational behavior and individual interests are taken into account. Thus, the main goal of Web personalization is the determination of relevant information without an explicit request [14]. An exploratory study by Khopkar et al. [12] shows that, despite the high level of interest in this topic, most Web search engines currently offer none, or limited personalization features, at all. In contrast to this attempt, the usage of an information push assumes the passive attitude of a user, where any information flow occurs unexpectedly by him [6]. The push technology has first been proposed by H. P. Luhn [13] in 1961 as "selective dissemination of information" (SDI). According to Bates [1], hundreds of millions of dollars have been invested during the Internet boom, but the push technology has largely failed. For example, in the mid nineties the Push Service InfoGate (formerly PointCast) rapidly enjoyed great popularity during the peak with over 1.5 million members. This Push Service provided information through special channels for stocks, sports, weather or business news. Designed as a client-server system, InfoGate paralyzed many networks. Several shortcomings caused the cancelling of the service by many users, for example the continuous connection to the server, and the restriction to specific topics. Today, the InfoGate service no longer exists. Since March 2004, Google came out with a new Push Service called Web Alerts[1]. At the same time a personalized Web search[2] was launched. With Google as the major Web search engine, new approaches for Pull and Push Services began. However, both services are still independent, and results do not influence each other. To improve the user's assistance, the Web system design has to go one step further. No restriction to specific information topics should exist in order to realize a fully personalized Web search. For this reason, we propose an integrated information service, assisting information pull and push based on a peer-to-peer (P2P) ar-

1 http://www.google.com/webalerts
2 http://labs.google.com/personalized

chitecture. A P2P system offers the transparency of such a service, because no personal information of former search requests is stored at a central server. Each peer is anonymous with the optimum assistance of personalized information consuming. On this account, a classical Web search engine is integrated as a Web Service to guarantee an efficient processing of requests. All results of an information pull are filtered by a Personalized Ranking List [10]. This list is built by a Peer Search Memory, which is introduced in Section 2. Other prerequisites for the design of the integrated information service are the grouping of information needs, and the generation of a peer profile. This makes it possible for a information to be consumed permanently. Section 3 describes the fundamental Pull-Push Cycle, where peers cooperatively exchange information in our system. In Section 4, we present development aspects of the prototype using the JXTA framework. We discuss in Section 5 evaluation results regarding the order of peers joining the net. Finally, we present related work in Section 6 and summarize our future plans in Section 7.

2. Prerequisites

This section describes the prerequisites used to design an integrated information service for information pull and push. Based on a local personalization strategy, all information needs are stored on the local peers. This strategy was chosen to achieve a wide acceptance of the service, where no personal information about search requests are stored at a central server.

Due to the user expectations of a high-performance Web search within a fraction of a second, we can not totally pass on a central entity. For these purposes, we integrated a highly-available Web Service (in our case Google) with an index of several billion Web sites. Hence, a hybrid peer-to-peer network (see Figure 1) summarizes not only nodes to share common information needs and relevant results, but also an efficient Web index. At each peer a *Peer Search Memory (PeerSy)* builds the fundamental basis of our system. On the one hand, PeerSy relevant results can be grouped to *Virtual Knowledge Communities*, in order to assist an effective information pull. On the other hand, an information push is assisted by a *Peer Service Repository* based on former information needs characterized by PeerSy. All three components are described in the following subsections.

2.1. Peer Search Memory (PeerSy)

The *Peer Search Memory (PeerSy)* [9] is developed to overcome the weaknesses of conventional search engines, and the current Web browsing technology accessing well-known Web sites. Due to the hybrid peer-to-peer

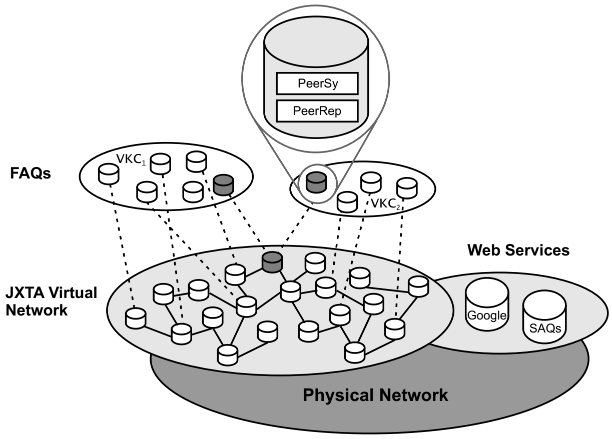

Figure 1. Hybrid Peer-to-Peer Network

network, for all search requests a common Web search engine (Google, in our case) is used for a pre-processing of information needs, and all queries are stored in relation to relevant Web sites on the local peer. The relevance is measured explicitly by the user. All Web sites satisfying a personal information need are flagged as relevant in a system-adapted browser. In a second step, a fundamental association between the query and the relevant result is built. With this processing, Personalized Ranking Lists [10] are built on each peer. The foundation for personalized Web retrieval is formed through the definition of fundamental associations. The Personalized Ranking Lists are used for a local filtering of Web search results. Previously known Web sites can be highlighted in the final presentation for a rapid access of known sites. Furthermore, PeerSy assists the exclusion of well-known sites in order to gain new information for an actual request. Hence, the optimal local set of results cannot improve the effectiveness of new, or extended information needs. For these purposes it is advantageous to consider the Personalized Ranking Lists of all users as described in the next section.

2.2. Virtual Knowledge Communities

The concept of *Virtual Knowledge Communities* [9] assists the process of an information pull in a P2P network. The claim beyond this approach is the collaborative exchange of Personalized Ranking Lists to enhance Web search effectiveness. As arrogated by Bates [1], retrieval systems have to be designed to work with the Bradford Distribution [4]. Frequencies of popular queries to a Web search engine do not conform to standard Gaussian or "normal" distribution. Instead, it works with the Bradford Distribution, where few topics are requested by huge numbers of people. A large number of topics are requested very little, if at all. Through a grouping of search interests over the P2P network, the small amount of popular topics, as well as all the topics requested very little are classified. The first ones are summarized in a set of Frequently Asked Queries (FAQs), and the second ones are collected in a set of Seldom Asked Queries (SAQs) [10]. In our first prototype a Web Service handles all requests for SAQs (cf. Figure 1). As future work the SAQs table will be evenly distributed over groups of users in order to avoid a central storage. These groups of users represent Virtual Knowledge Communities (VKC) for each topic of $FAQs$. Peers can share these local topics, only if they belong to the same VKC. An exchange of information between peers beyond communities memberships is not possible, and search requests are transferred to the integrated Web Services (Google and SAQs). In Figure 1 the grouping of Virtual Knowledge Communities is illustrated, and therefor all heterogenous information providers in the network are identified. Furthermore, the special network design, where users can be members of several Virtual Knowledge Communities, is important for a focussed retrieval of peers during information pushing.

Virtual Knowledge Communities primarily assist an information pull as described in [9]. Also, they can be used to restrict an information push to specific topics which arise in the network. Because of this recommendations can be computed using a local Peer Service Repository.

2.3. Peer Service Repository

In order to generate user recommendations, the *Peer Service Repository (PeeRep)* contains all necessary information to be sent to the user. These recommendations are based on former information needs localized at the Peer Search Memory. For the generation of qualified recommendations, three initial steps on each peer are necessary:

1st Step: For all items in the Personalized Ranking List several frequencies must be computed: the query frequency, document frequency, and term frequency. These frequencies can be used to weigh queries, documents, and terms with classical IR weighting approaches [16].

2nd Step: The number of accepted results arc_{p_i} is collected in order to rank peers p_i according to their quality of suggested results. During initialization, the value arc_{p_i} for all peers p_i is set to one. Hence, each peer p_i, within a set of peers P, gets the same peer relevance by default. The peer relevance is computed by the formula:

$$PRel(p_i) = \frac{arc_{p_i}}{\sqrt{\sum_{p_j \in P} arc_{p_j}^2}}.$$

As a normalization factor we consider all accepted results of peers within P. The set P of peers is built by all peers with membership to the same Virtual Knowledge Communities as peer p_i. During runtime the peer relevance is updated by the value of accepted recommendations, and joined peers.

3rd Step: The final step computes all basic values of the recommendation function. In analogy to Information Filtering methods (cf. [3]), a Personalized Peer Filtering method is developed. In contrast to PeerSy the PeerRep contains only automatically generated data. Explicit ranking information of Web sites is used to compute implicit feedback information. For this process each fundamental association of a peer p_i between a query and relevant result r (cf. Section 2.1) is weighted by the formula:

$$rank_{p_i,r} = \frac{1}{|T|} \sum_{t \in T} w_{r,t}$$

T is the set of terms used to index the Web site r. The weighting $w_{r,t}$ is based on the cosine coefficient [16] with a special emphasis on terms, which occur in the corresponding query. Additionally, a mean result weighting on each peer is computed based on the result weighting $rank_{p_i,r}$ of all results from peer p_i.

Long-term information needs are identified after all initialization steps. Together with all PeerSies and Virtual Knowledge Communities of the network, the platform for a cooperative Pull-Push Cycle is defined. Due to the application of Personalized Ranking Lists, local databases are used for persistent storage.

3. Cooperative Pull-Push Cycle

Our hybrid P2P architecture consists of a network of communicating peers (cf. Figure 1). Each peer is an information provider, as well as an information consumer. The consumption of information is interpreted as an active part of the peer, and the provision of information is the passive part. The Pull-Push Cycle is depicted in Figure 2, and is defined by a cooperative exchange of information. On this account, each peer works with other peers for a common purpose by generating queries during the pulling phase, and by propagating information during the pushing phase.

3.1. Pulling Phase

The pulling phase starts with an automatic generation of queries on an active peer. For these purposes, the process of query generation passes through a special order. In summary, this sequence is essential for effective processing on each peer, in order to consider the capability of different appliances.

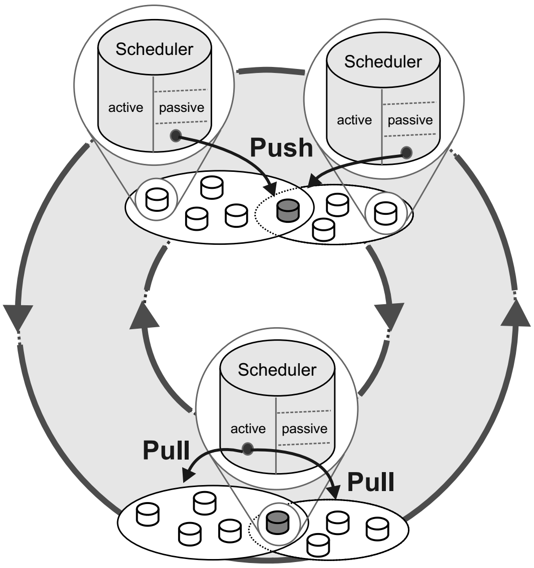

Figure 2. Cooperative Pull-Push Cycle

1. Selection of *languages* for recommended results.
2. Computation of *query terms* describing the information need.
3. Identification of the *connection type* of the active peer.
4. Specification of a *time-stamp* to avoid already known results.

During the selection process all local information is collected corresponding to the selected languages. In a second step the terms t used in all queries Q of the active peer are ranked by the relevance r_t with $t \in Q$:

$$r_t = \frac{qf_t}{|Q|}$$

qf_t is the query frequency which is continuously updated after the first peer initialization (cf. 2.3). The ranked list of all computed query terms is named $TSorted$. All highly ranked query terms represent continuous information needs of the user. Hence, in the next step a relevance threshold, based on the connection type, is computed. For this task we distinguish between different categories as presented in Table 1.

The connection type influences the transfer rate in the P2P network. We currently evaluate the chosen parameters, in order to reduce the network load to a minimum by the optimal effectiveness of our system. To compose the number of query terms we use the following formula:

$$\lfloor selTerms \rfloor = |TSorted| * \frac{1}{(ct * v_a)}$$

$|TSorted|$ quantifies the total number of ranked query terms. ct defines the connection type as depicted in Table 1, and v_a is a parameter to define finer nuances between the connection type categories dependent on the used appliances. This appliances parameter represents different local resources for a peer, and prevents a storage overload. Finally, a time-stamp is computed to record the last request in order to avoid already known documents.

ct	Connection Typ
1	LAN T3
2	LAN T1
3	WLAN (Wireless LAN)
5	Bluetooth
10	Modem

Table 1. Connection types

All selection steps at the active peer compose a set of query terms, which describe long-term information needs. These requests are sent out to Virtual Knowledge Communities with a membership of the active peer (cf. Figure 2). All peers in the same groups with the active peer take over the further processing as discussed in the next section. Hence, to gain a high amount of qualitative results, Virtual Knowledge Communities build the basis for a preselection of relevant resources.

3.2. Pushing Phase

During the pushing phase only the passive part of each peer is responsible for a propagation of information. The receiver of this information is an active peer. For the selection of relevant information on a passive peer the following information is necessary:

1. The *languages* of documents that the active peer wishes to obtain.
2. A *time-stamp* in order to retrieve only new results from a specified date, onward
3. A set of *query terms* which describe the information need of the active peer. These information needs are stored at the local Peer Service Repository.
4. The *connection type* of the active and passive peer.

In terms of an efficient processing of each request, the composition of all recommended information differs in the selection of queries during the information pull. According to these four restrictions, information is composed at each passive peer. In a first step all results in the Personal Ranking List are selected, which conform to the appointed languages of the active peer. From this set of possible recommendations all results are considered with a more recent time-stamp than the active peer. Hence, from this restricted set of results all hits are retrieved that match the requested query terms. These hits are ranked according to their implicit result weighting (cf. 2.3) in the set $RSorted$. Finally, the computed recommendations are composed by considering the connection type. To compute the number of recommendations of the selected result set, the communication type is set to

$$selConn = \begin{cases} AP & \text{when} \quad AP \geq PP \\ PP & \qquad\quad AP < PP \end{cases}$$

AP is the connection type of the active peer, and PP is the connection type of the passive peer (cf. Table 1). Analogous to the computation of the selected terms $selTerms$ (cf. section 4.1), the number of recommendations $selResults$ are computed by

$$\lfloor selResults \rfloor = |RSorted| * \frac{1}{selConn}$$

with $|RSorted|$ the total number of ranked results. Due to this threshold, the number of assigned results is computed. Additionally, to each result further meta-information according to the result weighting is sent to the active peer. More technical details about the whole Pull-Push Cycle and the corresponding communications are described in the next chapter.

4. Development Aspects

For the implementation of our prototype we chose the "JXTA Framework" for a standardized communication (cf. Section 4.1), and organization of peers within peergroups (cf. Section 4.2). According to our findings the usage of JXTA is promising due to the following aspects:

De-facto standard To date, JXTA constitutes the most sophisticated technology for creating P2P architectures. The JXTA standard is fully implemented in terms of an open reference implementation in JAVA.

Peer Grouping JXTA provides suitable concepts for grouping peers into self-governed groups, which can be used for the Virtual Knowledge Communities approach.

High Scalability JXTA's efficient routing and retrieving algorithms support our demand for a widely used decentralized application.

4.1. Peer Communication

The underlying communication protocols are built on top of the JXTA protocols. For our approach, a communication protocol is specified for messages (requests or replies) with the following criteria: binary coding, data security, and version control. Binary coding enables the system to reduce the network load, and we can refer to this property by the JXTA platform. To guarantee a secure exchange of information we plan to integrate a classical RSA approach [7]. Version control is very important for further development of the protocol. The implementation of the communication within the network occurs through *Pipes*, which are provided by the JXTA Framework (cf. [5]). Pipes are virtual connections between peers, and can be used as channels between members to support file sharing. An `InputPipe` defines an interface for receiving messages of a `PipeService`. At the same time, an `OutputPipe` defines an interface for sending messages of a `PipeService`. The main action within the Pull-Push Cycle is a continuous searching of group members. These members must be identified during the process of requesting an information need, as well as during the computation of recommendations restricted to a Virtual Knowledge Group. This task is realized by a `JXTABiDiPipe` (cf. [5]). A `JXTABiDiPipe` is a bidirectional pipe with a communication channel in both directions between sender and receiver. Once an `InputPipe` is initialized, it waits for a request to construct the `PipeConnection`. This pipe uses the `PipeService` for the initial connection to the pipe, and the address of the pipe endpoint is used for the reverse connection. Within the JXTA Framework the bi-directional pipe is internally implemented with two uni-directional pipes. Figure 3 depicts a typical peer communication to gain other group members. By default, the application instantiates a `JXTABiDiPipe`, that afterwards waits for a connection request (*createBiDiP-*

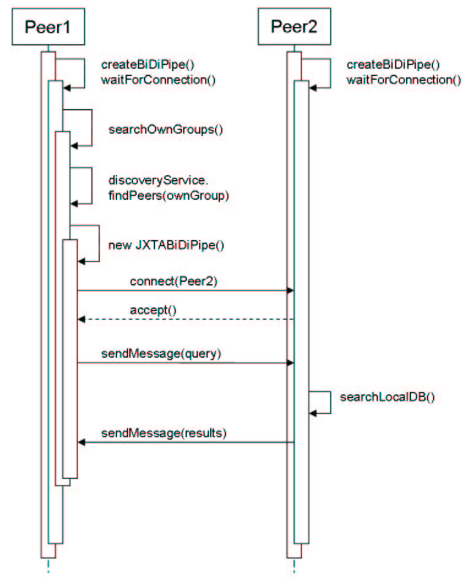

Figure 3. Communication of two peers

ipe(), *waitForConnection()*). The scenario in Figure 3 visualizes a search request of peer 1. In the following step the discoveryService is used to search all groups with a membership of peer 1 and their members (*discoveryService.findPeers(ownGroup)*). For example, a connection is built to peer 2 using a JXTABiDiPipe. Once the connection is established, search and recommendation requests can be handled with this connection.

4.2. Peer Grouping

In our prototype Virtual Knowledge Communities are represented through peergroups in JXTA. The members share all relevant parts of their locally stored associations with other members of the group. In JXTA a PeerGroupAdvertisement is assigned to each peergroup, which publishes information about the group in the network. A number of parameters, e.g. name and description of the peergroup, can be assigned to an advertisement. The PeerGroupAdvertisements are published in order to inform other peers in the network about the existence of the peergroup. This way new groups can be discovered throughout the network. A PeerGroupAdvertisement is created whenever a new group is formed. The peergroup name of the advertisement includes the main query terms of the group, and the initial associations of the FAQs-list are used as description. The main query terms are overlapping terms on which the grouping of associations of a knowledge community is based. The following example shows a typical PeerGroupAdvertisement for a VKC named ”java” (cf. Figure 4). The <Desc>-tag of the advertisement describes all queries and related links, which led to the formation of the group. <GID> specifies the PeerGroupID internally assigned by JXTA, which is associated with the instance of the group. <MSID> declares the ModuleSpecID that the group uses. This ID is used to find a module that references the services of the group.

4.3. User Interface

For the design of the user interface two main constraints have been defined: platform independence, and browser independence. To implement these constraints we chose an architecture with a WBI Proxy[3]. During the design phase special attention was paid to the ISO norm 9241-10. These principles describe the design of graphical user interfaces, and a user friendly operation of a system. Figure 5 depicts a screenshot of the actual prototype. The user interface is divided into three tabs: ”MySearch”, ”MyGroups”, and

3 http://www.almaden.ibm.com/cs/wbi/index.html

```
 <?xml version=''1.0'' encoding=''UTF-8''?>
<!DOCTYPE jxta:PGA> <jxta:PGA
xmlns:jxta=''http://jxta.org''>
 <GID>
   urn:jxta:uuid-35DF64686B64414A9D53F58E7429363602
 </GID>
 <MSID>
   urn:jxta:uuid-DEADBEEFDEAFBABAFEEDBABE000000010306
 </MSID>
 <Name>iskodor.peersy.jxta.java</Name>
 <Desc>
  <initialAssociations>
   <query>java</query>
   <doc>http://www.sun.com</doc>
   <query>java</query>
   <doc>http://www.java.org</doc>
   <query>java tutorial</query>
   <doc>http://java.sun.com/docs/books/tutorial/</doc>
  </initialAssociations>
 </Desc>
</jxta:PGA>
```

Figure 4. VKC Advertisement

”MyPush”. The ”MySearch” tab assists actual information needs during the pulling phase. The search interface is designed analogous to common Web search engines, in order to assist an optimal usability. In ”MyGroups” existing Virtual Knowledge Communities are summarized, and the peer membership of these groups is administrated. The ”MyPush” tab organizes recommendations in order to present all consumed and provided information. Furthermore, administration settings like connection type, peer name, and peer trustability can be set with this interface. For example the screenshot visualizes a recommendation, which consists of the following information: *Title*, *Summary*, *URL*, *Additional Information* and *Ranking*.

5. Evaluation

The performance of the Pull-Push Cycle is affected by several parameters. In comparison to Recommender Systems, no tagged evaluation corpus exists, and a new training set was built. Our training set consists of 163 query terms, and 279 results, with 483 associations. In this paper, we present an evaluation setting to analyze the influence of the peer order joining the network on the recommendation result. With respect to the training set, N peers are initialized with the training set. Additionally, one peer acts as an active peer (called p_N), and he has one association less than all of the passive peers p_1 to p_{N-1}. Each passive peer computes implicit weights for each association with the formula:

$$weight(p_i) = \frac{i}{N}$$

for $1 \leq i \leq N-1$. Hence, all passive peers hold different result weights. Furthermore, this evaluation setting differentiates between four test cases for the incremental joining of passive peers to the network:

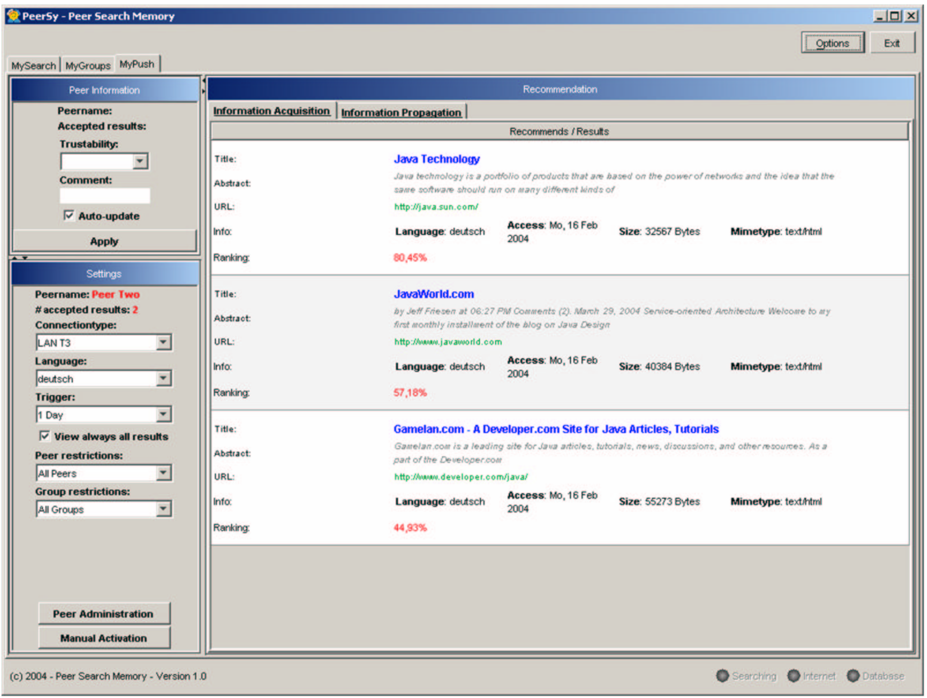

Figure 5. Screenshot of the prototype

1. Starting with peer p_1, the peers p_2 to p_{N-1} join the network in ascending order .

2. Starting with peer p_1, the peers p_{N-1} to p_1 join the network in descending order.

3. Starting with peer p_1, all passive peers incrementally join the network according to the sequence $p_1, p_{N-1}, p_2, p_{N-2}, p_3, p_{N-3}, \ldots$.

4. Starting with peer p_1, all passive peers incrementally join the network according to the sequence $p_{N-1}, p_1, p_{N-2}, p_2, p_{N-3}, p_3, \cdots$

After each step progresses, the network recommendations are computed dependent on the number of peers. Figure 6 shows the results considering 10 peers ($N = 10$). With a growing number of peers and identical peer relevances, the recommendation value depends on the implicit weights, and not on the overall number of peers. A recommendation value of 0.5 is optimal, because all result weights are evenly distributed in this evaluation setting. Test cases 1 and 2 describe a monotone approximation to the value 0.5. In contrast to this behavior in test cases 3 and 4, the usage of extreme values affects a rapid approximation. A new peer can significantly influence the result in a positive or negative manner. We plan to further evaluate whether the results can be proven with a larger number of peers.

6. Related Work

Contemporary retrieval systems are designed to assist only an information pull. Through active requests the user attempts satisfaction of his information need. However, all systems assisting information push are implemented as a Pull Service. Due to the independence of events, and the simplicity of implementation, Push Services use a periodic pull and unicast connections. To avoid a technical comparison between systems and their design, the following systems are discussed on a semantical level. On this basis, they could be classified by the assistance of information pull or push. Presently, there is no system which supports both information gathering methods in an active, or passive way. On this account, first the distributed search application *YouSearch* [2] is considered for users working in a shared working context. It supports the aggregation of peers into overlapping (user defined) groups, and the search over specific groups. The hybrid peer-to-peer architecture is augmented with a light-weight centralized component. In comparison to our approach, YouSearch does not provide bookmark-sharing, but more or less file-sharing, which is supported by search mechanisms. Another difference is that groups are formed via manual user actions. This system does not conduct any proposals to support and enhance this process, for example directly approaches the peers concerned to recommend a group creation. *Tapestry* is an experimental subsystem of the Xerox Mail Service at the Palo

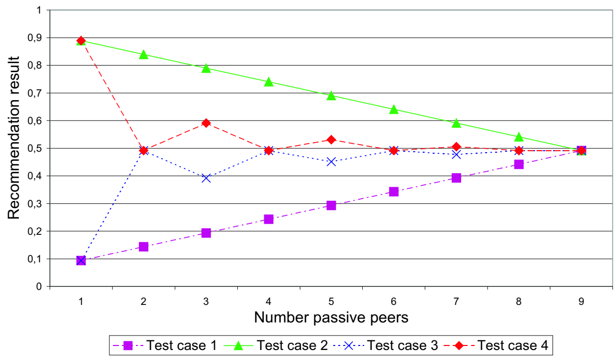

Figure 6. Results of peer order analysis

Alto Research Center [11]. This system is based on a client server architecture, and can also be integrated in other systems, for example in NetNews systems. E-mails are classified through Collaborative Filtering methods as relevant or irrelevant according to special user interests. For these purposes, users are asked to give feedback to read e-mails. The system *GroupLens* [15] extracts relevant subsets of NetNews articles for a user. In analogy to the Tapestry system, a Collaborative Filtering technique is used to generate recommendations. This process exploits positive user feedback of the past for future interests. For this task, GroupLens expects an explicit numerical ranking of an article in the range of one (not recommended) to five (excellent). The system is designed as a distributed system in order to collect rankings of several users for recommendations to other users.

7. Conclusions

This paper proposes a cooperative Pull-Push Cycle enhancing the search of information within a hybrid P2P network. Long-term search requests are identified through a local Peer Search Memory. In cooperation with other peers, all local Personalized Ranking Lists are involved to assist the user by requesting information needs. All peers assume active and passive tasks in order to realize an information push. The relevance of special peers represents the usability of other peers according to the pushed information. In contrast to existing Push Services, our approach underlies no context restrictions. Moreover, a first prototype on a peer-to-peer platform is realized. Future work on the prototype will lead in two directions. First, we will investigate the impact of additional Web Services like news, stocks or weather services. Another option is to integrate more Web search engines in order to enlarge a meta-search service on a P2P basis. Second, we will integrate more security functions with the goal to construct a "Network of Trust". This trust concept is based on the recommendation of a user, who only can join the network by invitation. Consequently, measuring the peer relevance is an important aspect for further investigation.

References

[1] M. Bates. After the dot-bomb: Getting web information retrieval right this time. *First Monday*, 7(7), 2002.

[2] M. Bawa, R. Bayardo, S. Rajagopalan, and E. Shekita. Make it fresh, make it quick – searching a network of personal webservers. In *Proceedings of the 12th International World Wide Web Conference*, pages 577–586, Budapest, Hungary, 2003.

[3] N. Belkin and B. Croft. Information filtering and information retrieval: two sides of the same coin? *CACM*, 35(2):29–38, 1992.

[4] B. Brookes. Theory of the bradford law. *Journal of Documentation*, 33(3):180–209, 1977.

[5] D. Brookshier, D. Govoni, and N. Krishnan. *JXTA: Java P2P Programming*. SAMS Verlag, Indianapolis, IN, United States, 2002.

[6] K. Cheverst and G. Smith. Exploring the notion of information push and pull with respect to the user intention and disruption. In *International workshop on Distributed and Disappearing User Interfaces in Ubiquitous Computing*, pages 67–72, 2001.

[7] D. E. Denning. Digital signatures with rsa and other public-key cryptosystems. *CACM*, 327(4):388–392, 1984.

[8] M. Eirinaki and M. Vazirgiannis. Web mining for web personalization. In *Proceedings of the ACM Transactions on Internet Technology*, volume 3, pages 1–27, February 2003.

[9] M. Gnasa, S. Alda, J. Grigull, and A. B. Cremers. Towards virtual knowledge communities in peer-to-peer networks. In *SIGIR 2003 Workshop on Distributed Information Retrieval*. LNCS, 2003.

[10] M. Gnasa, M. Won, and A. B. Cremers. Three pillars for congenial web searching. In *WWW 2004 Workshop on Web Search Effectiveness: User Perspective*, 2004.

[11] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to waeve an information tapestry. *CACM*, 35(12):61–70, 1992.

[12] Y. Khopkar, A. Spink, C. L. Giles, P. Shah, and S. Debnath. Search engine personalization: An exploratory study. *First Monday*, 8(7), 2003.

[13] H. Luhn. Selective dissemination of new scientific information with the aid of electronic processing equipment. *American Documentation*, 12(2):131–138.

[14] M. Mulvenna, S. Anand, and A. Buchner. Personalization on the net using web mining. *CACM*, 43(8):122–125, 2000.

[15] P. Resnick, N. Iacovou, M. Suchak, P. Bergstorm, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *Proceedings of ACM Conference on Computer Supported Cooperative Work*, pages 175–186, Chapel Hill, North Carolina, 1994.

[16] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.