Full-text indexing and Information Retrieval in P2P Systems

Odysseas Papapetrou
L3S Research Center, Leibniz Universität Hannover, Germany
papapetrou@l3s.de

ABSTRACT

Current distributed IR approaches are not readily applicable for P2P scenarios. The high dynamics in these networks and the high cost for building and maintaining indices over Distributed Hashtables make full text indexing and information processing difficult to scale for large P2P networks. My work will propose new approaches for enabling distributed IR over P2P without limiting the network size or mutilating the IR. The basis of these approaches is an innovative distributed clustering algorithm, which can cluster peers in a P2P network based on their content similarity. This clustering enables significant network savings and enables new families of distributed IR algorithms.

Categories and Subject Descriptors

H.2.4 [**Systems**]: [Distributed databases]

General Terms

Peer-to-peer, Distributed Information Retrieval

1. INTRODUCTION

The focus of this thesis is to study the problem of efficient high-quality Information Retrieval (IR) over P2P networks. My target is to keep the network and hardware load of the peers at an acceptable level while still enabling quality IR results, comparable to today's state-of-the-art centralized systems. In addition, the proposal needs to gracefully handle peer churn and keep pace with network growth.

P2P IR is necessary in scenarios where the information is inherently distributed in many locations. Consider for instance P2P desktop-sharing applications like Beagle++ [1] or semantic desktop-sharing applications like NEPOMUK[1]. Similar scenarios involve P2P digital libraries like the ones studied in DELOS [2], and P2P social networks [8]. A centralized IR solution for these systems has serious limitations. First, it requires all the peers to send all their documents (or indexes) to a central repository; this comes in contrast with the systems' sizes and their promises for robustness, low cost, and unlimited scalability. Second, the quantity of the peers' data, and, most importantly, the large number of peers, impose challenging network and hardware load on the machines that host the central repositories. The cost for maintaining such a data center is significant and cannot easily be covered from a company for free. The query execution workload can also be beyond the realistic capabilities of a single machine or a data center, especially if advanced IR techniques are required. Third, the expected high churn in P2P systems creates a situation unlike any previous large-scale centralized searching systems, e.g., web search engines. The monetary cost of compensating for churn in the rate of minutes and not days or weeks is just too high for current data centers, assuming large-scale P2P networks. For these applications, a purely P2P IR solution is required.

Most P2P systems implement IR by constructing a distributed inverted index over a Distributed Hashtable (DHT). The distributed index maps each term to a list of relevant peers, their IP addresses and their scores. However, full text indexing, the basis for high-quality IR, is too expensive for P2P systems and cannot scale. In fact, it is estimated [14] that the expected load per peer for a full-text indexing solution is at least an order of magnitude more than the feasible.

Current P2P IR proposals usually face the above problem by indexing only part of the peers' data. For instance, peers index only a limited number of keywords, only the documents' titles or other meta-data. This practice sacrifices the completeness of the inverted index and reduces the IR quality. It also increases the cost of finding satisfactory results for a query, since an incomplete inverted index typically does not give the optimal results; unclear or very short titles, ambiguous terms, absence of keywords can mislead the ranking algorithm.

Even if the full text indexing problem is solved, the problem of advanced P2P IR, comparable to state of the art centralized approaches, is still challenging. Current P2P techniques usually employ a TFxIDF scoring, or something similar. They cannot address traditional IR problems, like polysemy and synonymy, as algorithms for solving these problems require collection-wide information, i.e., co-occurrence statistics. Also, they cannot use more complex techniques for document relevance ranking. Only few systems manage to deploy advanced IR techniques in distributed P2P (see Section 2).

Problem Statement: Our work focuses on building an

[1]http://nepomuk.semanticdesktop.org/

EDBT 2008 Ph.D. Workshop '08, March 25, 2008, Nantes, France

advanced P2P IR system with a realistic network cost. We investigate the following problems:

- Reducing the inverted index maintenance cost. The approaches employed from current DHT-based systems fail to scale for large collections and networks, mainly because of the big number of the DHT lookups.
- Increasing the IR quality. Many approaches outperform TFxIDF-alike algorithms, but they are not applicable in P2P setups. The P2P overlay proposed here enables the adaptation of a class of these approaches with a small execution cost. We will investigate which of these approaches can be adapted to our infrastructure and search for new IR approaches that are enabled from the new cluster-based overlay.

Contribution: My thesis proposes a scalable Peer-to-peer infrastructure that enables advanced Information Retrieval, and imposes low network and hardware load on the peers. The proposal builds on our recent work [17], which clusters peers in P2P systems around super-peers based on content similarity. In summary, each new peer first uses the DHT overlay to find a cluster and submits its inverted index to the super-peer of that cluster. In turn, the super-peers of the clusters publish the keyword scores for each of their peers, but they take advantage of the keyword overlaps and the larger but much less network messages to reduce the total network usage. The new architecture reduces the the inverted index maintenance cost by as much as one order of magnitude. At the same time, the workload of the super-peers remains manageable and is usually even less than the workload of regular peers in earlier P2P systems.

The new architecture does not reduce the Information Retrieval quality as it creates exactly the same full-text inverted index as created by previous approaches: a score for all the keywords in each peer finally gets published at the DHT. Thus traditional DHT-lookup based query execution techniques can still be applied. In addition, our approach paves the road for advanced IR techniques, like semantic indexing, concept indexing, and probabilistic IR, which yield better IR results than the popular TFxIDF approach. Inside each peer cluster, the peers can easily solve problems otherwise difficult to solve over DHTs, like synonymy and polysemy. IDF weighting and novelty estimation can also be inexpensively applied inside each cluster.

The next section discusses the state-of-the-art in the area. Section 3 describes our contribution, giving emphasis to the finished tasks and results. We conclude with Section 4, which describes the foreseen tasks for enabling advanced P2P IR.

2. STATE OF THE ART

This thesis contributes to the areas of IR, P2P IR, and distributed clustering. We will now review the main related works from these areas, and explain how they can be applied in or enhanced by our research.

Centralized Information Retrieval: A very interesting work on IR is built around Semantic Analysis. Latent Semantic Analysis (LSA) [5] uses singular value decomposition for extracting the most important dimensions from a collection and reducing the document representations to these dimensions. LSA also addresses the polysemy and synonymy problem, and gives relevance rankings superior to the rankings of most methods operating on full document dimensions. A work by Papadimitriou based on random projection [16] also makes the LSA execution on large collections computationally feasible. However, it is still very difficult to apply LSA in a P2P setup because it requires system-wide information. In section 4 we show how to overcome this problem without the need of a central repository.

Karypis and Han present concept indexing [11], a Semantic Analysis approach with results comparable to LSA results. Concept indexing is based on document clustering, has low execution cost and scales well with the number of documents. Although concept indexing is not applicable in P2P setups, it is useful for our work. A result from [11] is that few of the clusters' keywords are responsible for a large percentage of the total cluster length. If we detect these keywords, we can use them to index the peer clusters and perform distributed clustering. We can also use a similar approach for reducing the document dimensions if the document clusters and their centroids are already in place.

P2P Information Retrieval: P2P IR focuses on two main problems: (a) efficiently selecting the relevant peers for routing the query to them (collection selection problem), and (b) executing the queries in the peers, and merging the results.

The dominant approach for addressing the collection selection problem is by maintaining a system-wide inverted index over a DHT: upon joining the network, each peer joins the DHT and publishes its data in the distributed inverted index. In this text, we refer to this approach by the name *Flat DHT*, as it does not use an intermediate layer between the peers and the DHT to optimize the index maintenance.

The granularity and completeness of the inverted index varies from system to system. Peers in the ALVIS system [3] index the keywords for each of their documents independently (document-granularity index). The Minerva system [4] reduces the index size by indexing peer scores; the peers aggregate their documents' scores per keyword to produce a single score per keyword. But the network cost for full-text indexing is still too high [14, 18], even with peer granularity data. The main part of the network cost is generated by the huge number of the DHT lookups.

The new version of ALVIS [19] increases the inverted index scalability by: (a) publishing only the top relevant documents per peer for each keyword, and, (b) identifying the highly discriminative keys, which may be two or more words together, and also publishing these in the inverted index. The extension limits the size of the inverted index and speeds up the query execution process. However, it does not reduce the number of the required DHT lookups. An optimization in the P-Grid DHT layer can partially alleviate the problem by packing many small DHT messages together [12]. However, this optimization is orthogonal to ALVIS publishing algorithm, and can be used to further optimize any approach, including the one proposed in this work.

To reduce the high network cost, some other systems do an initial filtering of terms; each peer only publishes the most important terms (e.g., only the terms occurring in the document titles and/or abstracts). Organizing the peers into ontologies [9], or asking each user to manually select the keywords for her files [6] is also suggested in the literature. The systems in this family are scalable, as they restrict the DHT maintenance cost drastically. However, they are not oriented towards full text search, and they cannot of-

fer advanced IR. Moreover, the manual keyword selection is troublesome for the user and an automatic keyword filtering cannot be executed efficiently in a distributed network as it demands global system information (IDF values).

Some P2P IR systems do not use keyword indexing at all. Nottelmann and Fuhr [15] build an IR system over a hierarchical P2P network. The peers there do not maintain a distributed index; instead, some super-peers are assigned the responsibility to keep their peers' summaries, and to forward the queries to the most related of their peers, or to other super-peers. In addition to the infrastructure, the authors present a decision theoretic model for optimal P2P query routing. For selecting the peers for each query, their model considers the cost of query routing and the expected results from each peer. The approach gives expected optimal query results for the query execution cost. This work is very important for our research, as it is one of the few works proposing probability-based query routing in P2P systems. It is also the only work which considers the querying cost per peer for reducing the overall cost without sacrificing the quality of the results. However, the focus of our work is on DHT-based P2P systems. The major cost in these systems is the indexing cost, where each peer indexes its collection in the distributed inverted index. Our cost model will combine the DHT indexing cost and the query execution costs.

pSearch [20] system also avoids term indexing. The work proposes two alternatives for indexing the documents in P2P. The first one, pVSM, is based on the Vector Space model, while the second one, pLSI, reduces the document dimensions using Latent Semantic Indexing. The main problem in both the approaches is the high network cost. In addition, pLSI assigns the LSA computation to a single peer, which causes a bottleneck for big networks. Also, the proposed load balancing causes a huge increase of the network cost.

Distributed Clustering: The main indexing approach proposed here is based on P2P text clustering. There are several distributed clustering algorithms, but most of them have high communication requirements, especially when the data dimensionality is high, e.g., in text clustering. A P2P clustering approach is proposed by Datta et al. [7]. The algorithm employs gossiping for the distribution of the cluster centroids. Each peer performs a local K-Means clustering and then broadcasts its centroids at its neighbors. Then, it averages the centroids received from its neighbors with its own centroids to produce its new centroids, and repeats the process. The algorithm is accurate and uses only local communication. However, it is tested only with low dimensional data (10-dimensional synthetic data). It also requires too many iterations. Furthermore, the practice of transmitting centroids instead of document summaries is suboptimal for P2P systems, where peers usually publish a small number of documents, less than the number of clusters.

A more recent P2P clustering approach proposed by Hammouda and Kamel [10] uses a hierarchical topology for the coordination of K-Means. Peers are organized into small neighborhoods, and each neighborhood performs a local K-Means clustering. The results are merged hierarchically to give the final k clusters. However, this hierarchical merging causes a significant clustering quality loss, thus the algorithm cannot be used in large P2P networks. In our work we propose a novel P2P clustering algorithm which addresses the limitations of the previous algorithms. The algorithm is based on a DHT inverted index and can be executed with a very small cost.

3. PCIR: A P2P IR INFRASTRUCTURE EMPOWERED FROM PEER CLUSTERING

Our proposal, PCIR (short for Peer Clustering Information Retrieval) combines peer clustering and an inverted index over a DHT (fig. 1). A new peer first joins the DHT, yet without publishing its contents. Following, it discovers and joins a suitable cluster of peers (Section 3.2), and sends its inverted index to the super-peer of that cluster (Section 3.1). In turn, the super-peer of each cluster publishes the inverted indices of all the cluster peers to the DHT, but now by taking advantage of the overlapping content in the peers to optimize the publishing (Section 3.1). The peer clustering and DHT publishing steps are repeated periodically to compensate churn. The total number of clusters is also dynamically adapted to the number of peers and the diversity of their collections.

Note that super-peers in PCIR publish the detailed inverted indices of all their peers, not an aggregated cluster index; this way they generate exactly the same DHT-based inverted index as with previous full-text DHT approaches. Thus, the traditional P2P query execution techniques can be applied and the query execution cost and IR quality are not affected from the PCIR infrastructure.

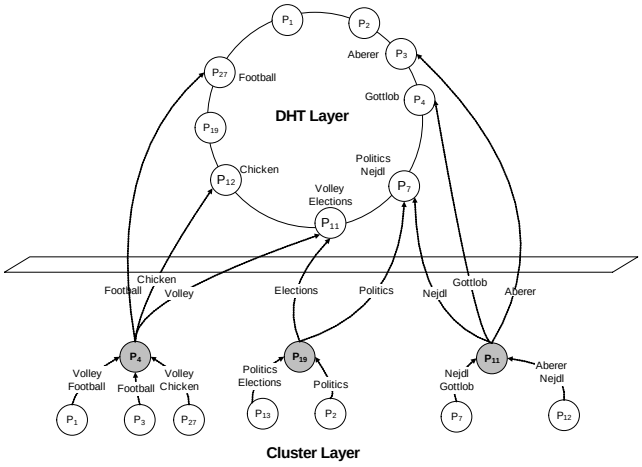

Figure 1: The two-layer architecture combines peer clustering and an inverted index over a DHT (Super-peers are gray shaded)

The DHT overlay, i.e., Chord, is used for building the keyword inverted index. Both super-peers and regular virtual peers participate in the DHT and contribute their resources to store part of the inverted index. We now describe the inverted index maintenance and the the peer clustering algorithm. Although in practice the peer clustering precedes the inverted index maintenance, they are discussed in reverse order here for the sake of clarity. We continue with the experimental evaluation and an outline of our future tasks for further reducing the maintenance cost in PCIR.

3.1 Inverted index maintenance

We assume that the peers have already formed clusters, and each cluster has a super-peer. Then, all the peers filter out the stopwords from their local collection and perform

stemming. Following, they send their Peer Content Summary (PCS) to their responsible super peer. The PCS is a list of terms with their term frequencies in the peer(fig. 2).

Each super-peer then merges all the PCS to create the cluster inverted index (fig. 3), which maps each term to a list of the relevant peers in the cluster and their peer frequencies (the *Peerlist* of the term). Following, the super-peers look up the terms at the DHT. For each each term, they find the responsible peer in the DHT and they submit the Peerlist for their cluster. Note that the Peerlist for a keyword includes the keyword scores per cluster peer separately, it is not an aggregated cluster score. For the λ most frequent keywords in the cluster they also submit the cluster score. A cluster score for a keyword is the average frequency of that keyword in all the cluster peers. These cluster scores are used for efficient peer clustering (explained in next section). Publishing of these scores generates only λ additional records per cluster in the DHT and requires no additional DHT lookups. The resulting inverted index structure is similar to the one in fig. 4 (cluster scores appear gray-shaded). In our implementation, the peer and cluster scores are normalized to the peer resp. cluster length, to account for peers/clusters of different lengths. For clarity we abstract from details of the normalization in this paper.

Peer Content Summary	
Term	**PF**
Football	12
Chicken	3
Volley	13
...	...

Figure 2: The Peer Content Summary is a (term, peer frequency) matrix, for all the peer terms

Cluster Inverted Index		
Term	**Peer**	**PF**
Football	$Peer_1$	12
	$Peer_7$	9
Volley	$Peer_{27}$	13
...	...	...

Figure 3: The Cluster Inverted Index stores the PeerLists for all the cluster keywords

Keyword	**Peer**	**PF**	**Super-peer**
Football	$Peer_1$	12	P_4
	$Peer_3$	9	P_4
	$Peer_9$	1	P_9
	ClusterScore	7	P_4
Volley	$Peer_1$	11	P_4
	...	...	...
...	...	...	...

Figure 4: Logical Top DHT Routing Table for clustering-enhanced approach

The above steps are repeated periodically to compensate churn. In this way, the distributed inverted index does not need to be updated every time a peer leaves or fails. Instead, its postings expire and are removed from the inverted index.

The clustering-enhanced approach reduces the DHT maintenance cost by an order of magnitude compared to the flat DHT maintenance algorithm (e.g., Minerva). The main reasons for this drastic reduction are the following three:

1. The number of the required DHT lookups is drastically reduced since only one lookup is executed per distinct cluster keyword.
2. The publishing messages for a keyword are now packed together for all the peers in the cluster, requiring less messages and causing less network overhead.
3. The size of the messages is further reduced with compression. Message compression does not have positive effects in the case of flat DHT systems, since the messages there are very small to be compressible.

Both regular virtual peers and super-peers benefit from the network optimizations. In fact, in all our experiments (see Section 3.3), the average network load (both number of messages and transfer volume) in the super-peers in the clustering-enhanced approach is even below the load of the peers in the flat DHT scenario. In the large networks experiments (more than 3000 peers), the super-peers have on average less than 50% of the network load of regular peers in the Flat DHT. The reduction in the super-peers load is attributed mainly to the significant reduction of the DHT lookups, which equally affects all the participating peers.

3.2 Distributed P2P Clustering

The clustering overlay is built incrementally in a distributed fashion, based on the contents of the peers. A new peer finds and joins a cluster as follows. First, the peer joins the DHT, but without publishing any data. Then it finds its top-λ frequent keywords, and it looks them up at the DHT to discover the clusters that have published a *ClusterScore* (not a normal peer score) for at least one of these keywords in the DHT. If such clusters exist, the peer retrieves the cluster scores for these keywords (e.g., the gray-shaded row in fig. 4), and computes a partial cosine similarity for each candidate cluster (based only on its top-λ keywords).

If no clusters are retrieved from the top-λ keywords lookup, the new peer creates its own cluster and becomes the super-peer. But if candidate clusters are retrieved, the top-μ most similar ones are found, based on their partial cosine similarity with the peer's collection. Then, the new peer sends a compact summary of its collection (as a bloom filter) at the super-peers of these clusters. The super peers use the summary for estimating the term overlap between the cluster centroid and the peer collection. The peer retrieves these scores from the super-peers, and uses them to select and join the most similar cluster. At the end of the clustering algorithm, each peer belongs to one cluster, and each cluster will have a super-peer.

Clustering is repeated periodically to compensate churn. The number of clusters constantly adapts to the present peers and their data. The super-peers also change regularly, thus the super-peer workload is evenly distributed among all peers.

The peer clustering algorithm is inexpensive in network resources. The total cost for clustering a new peer is $O(\lambda * \log(n)+\mu)$ messages (λ and μ are typically less than 5). The values of λ and μ are important for our algorithm. Too small values can result to a clustering of inferior quality, thus less keyword overlaps in the super-peers, while too large values result to large communication costs, making the approach less appealing for P2P. We are working on a complete system model which will allow us to optimize these values given the size of the network and the keyword distributions. However, large-scale experimental evaluation with real data has shown that λ and μ less than 5 already result in good clusterings.

Bloom filter representations: Bloom filters can com-

pactly represent a set of objects, i.e., words, especially when there is error tolerance. We use bloom filters for representing cluster and peer centroids. Instead of transmitting the centroids over the network, we send their bloom filter representations. The bloom filters are created by hashing all the cluster resp. peer keywords. They are used for estimating the keyword overlap between a peer and a cluster, as we show in [18].

Managing Peer Collection Diversity by Virtual Peers: Real-life peer collections, as real persons' interests, are often diverse with respect to their topics; a peer may collect documents about many different, even orthogonal topics. Finding the best cluster for a multi-thematic peer is difficult. We solve this problem by employing document clustering for splitting a peer to a set of virtual peers, each with more homogeneous documents. First, the documents in a peer are clustered using a partition clustering algorithm, and each document cluster is assigned to a virtual peer. Then, each virtual peer proceeds independently, joins the best matching peer cluster for its documents, and posts its contents at the super-peer of that cluster. This increases the keyword overlap in the super-peers and reduces the inverted index maintenance cost.

The document clustering method can vary. Document clustering in this work was performed with K-Means. In a previous work [17] we performed the document clustering using the MeSH ontology, with similar results.

Load Balancing: The popularity of subjects is not uniform. It is thus natural for a meaningful clustering algorithm to produce clusters of non-uniform sizes, and among them, some large clusters. The super-peers of these large clusters can still cause bottlenecks due to their cluster sizes. We solve this problem by restricting the maximum cluster size. Each super-peer sets the maximum number of peers in its cluster, so that no cluster becomes difficult to manage. The solution has only a slight performance penalty (see [18]), but it manages to distribute the load evenly among super-peers.

Next steps on Distributed Clustering.

Distributed P2P clustering can be useful in other contexts, so we continue our work on it. We now focus on dynamically adjusting the values of the parameters (λ and μ) based on the probability of correct clustering to occur. In particular, we use statistics from the peer collections to decide on the cutoff thresholds. Having said that, we should stress that errors in the clustering algorithm for PCIR do not mean errors in query execution; the only penalty that comes from a clustering error is less keyword overlap in the super-peers, thus slightly more messages. In fact, in the theoretical system model that we currently work on, we are not reducing the clustering errors, but the overall network cost.

3.3 Experimental Evaluation

We evaluated our approach experimentally using the first 160,000 documents from the Reuters Lyrl2004 document collection[13]. The documents belonged to a total of 148 categories and each of the used categories had at least 400 documents. Some of these categories were very broad, while others were very focused.

Building the peer collections: Real-life peer collections, as in real personsŠ interests, are often multi-thematic. Some users may be well-focused, having very specific documents of only one topic. Other users may focus on a couple of non-related topics, while still others may just collect many diverse documents. We simulated all such kinds of users by using the 148 Reuters categories. Each peer was randomly selecting 3 of the 148 categories, and then 20 random documents for each category. At the end, a total of 60 distinct documents were assigned to each peer.

In particular, we compared the following approaches:

Flat DHT: The flat DHT publishing, where each peer publishes its own collection directly to the DHT inverted index. The flat DHT publishing serves as the baseline.

PCIR: The PCIR approach. We use bloom filters of length 128Kbits and 3 hash functions, and break each real peer to 3 virtual peers using K-Means. We set $\lambda = 6$ and $\mu = 2$.

Random: The PCIR approach with random peer grouping. The peers do not break to virtual peers, and they form random groups instead of semantic clusters. Each peer randomly selects and joins a peer group. The super-peer of each group again updates the inverted index for the peers in the group. We implement this approach to evaluate the effect of peer clustering on PCIR.

We constructed networks of the following sizes: 500, 1000, 2000, 3000, 4000 and 5000 peers. Each experiment was executed 8 times and the average costs were taken over all executions.

Each setup was let to run for 4 iterations/periods. The experiments were repeated with and without churn. We implemented churn by randomly selecting 20% of the peers from all the setups at every iteration/period, and replacing them with new ones. In PCIR, in order to avoid data loss from the peer churn, when a super-peer was selected to be replaced, it first had to publish the cluster's inverted index at the DHT and then depart. The costs for publishing were added to the total network costs for the setup.

We measured and compared the following network requirements: (a) number of messages, and (b) transfer volume. All the network traffic was measured, except of the traffic for building the Chord ring. The cost for building the Chord ring was exactly the same for all the 3 approaches (and for all structured P2P systems), thus it was ignored.

Results: As expected, churn did not have an impact on the network usage for the Flat DHT approach, because of the periodic republishing. It slightly affected the PCIR configurations because when a super-peer was selected to be removed, it had to publish all the cluster's data at the DHT prior disconnection. This caused duplicate peer score publishings in some cases. However, since the difference between the results with and without churn was less than 3%, we only report the results of the experiments with churn.

Figure 5 plots the network requirements for each setup. For illustration purposes, we normalize the network cost in each setup using the network cost of the Flat DHT setup as a baseline (100%). The other costs are presented as percentages of the network cost of the Flat DHT approach. The absolute average values per peer for these setups are presented in table 1.

Even by random peer grouping, the network savings are significant; for the 5000 peers setup the random grouping approach has a network cost around 18% of the flat DHT cost. The network savings for the PCIR approach are more. In the larger networks, the PCIR approach has around 5% less

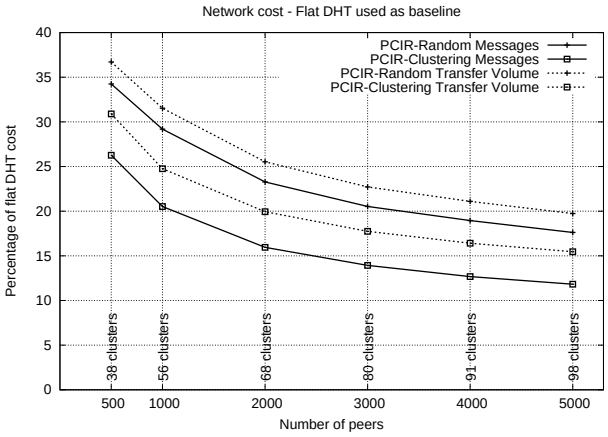

Figure 5: Network usage. The figure is annotated with the number of clusters created at each setup

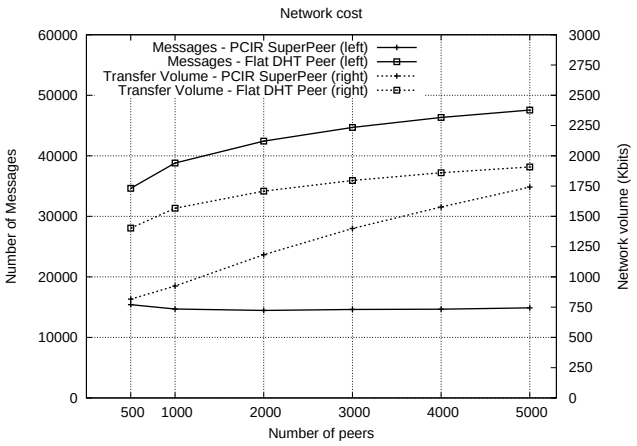

Figure 6: Network load of super-peers in PCIR and regular peers in flat DHT

network cost compared to the random grouping approach, and about one order of magnitude less cost than the flat DHT approach.

Peers	Number of Messages			Transfer Volume (Kbits)		
	F	R	C	F	R	C
500	17314	5929	4546	701	257	216
1000	19404	5663	3982	783	247	193
2000	21214	4936	3383	854	218	170
3000	22346	4587	3112	898	204	159
4000	23167	4389	2936	930	196	152
5000	23782	4190	2811	954	188	147

Table 1: Average cost per peer: F:Flat DHT | R: PCIR Random | C: PCIR Clustering

All the experiments were repeated with the load balancing extension, and the results were similar to the above; the extra cost introduced from the load balancing was negligible.

We also investigated how the load of the peers in PCIR that host a super-peer compares with the load of regular peers in the flat DHT approach. We measured all the network messages that the super-peers send or receive in PCIR, and likewise for the flat DHT peers. In both the setups each message was measured twice, once in the sender and a second time in the recipient. For PCIR in particular, since each peer consisted of 3 virtual peers, the network load of the peers that were hosting a super-peer was computed by summing the network loads of all their virtual peers (super-peer and regular virtual peers). Figure 6 summarizes the results. An important observation is that the average super-peer load still remains below the average regular peer load in the flat DHT scenario for all the setups. This reduction is attributed to the significant reduction of the DHT lookups in PCIR, which equally affects all the participating peers. In the smaller setups, the super-peer network load (both number of messages and transfer volume) is around 50% of the average peer load in the flat DHT approach. For the larger setups, the super-peers handle as much as 2/3 less network messages compared to the messages handled by the regular peers of the flat DHT approach. As far as the transfer volume is concerned, the load in PCIR super-peers still compares favorably with the load in the flat DHT regular peers, but not with as much difference as before. The main reason for this change is that some messages received by the super-peers in PCIR are large (the messages between the super-peer and the regular virtual peers), while in the flat DHT all the messages are approximately of the same size. While for our setups this was never a problem for the super-peers (they always handled less network transfer volume compared to the flat DHT peers), evenly distributing the super-peer workload is part of our current work. In any case, each of the super-peers can control its network load by setting a strict upper limit on the size of its cluster (the load balancing mechanism). Thus the super-peers in PCIR do not risk getting overloaded.

The reader can find a detailed description of the evaluation methodology and further interesting results (including experiments with the Medline collection and other bloom filter lengths) in [18]. In the same document we present a cost analysis, which gives an insight for the results presented here.

3.4 Next steps on PCIR

Load Balancing: We now work on the load balancing between the regular peers and the super-peers. While the super-peers in PCIR still have less network load than the load of regular peers in flat DHT systems, they have more network load compared to the load of regular peers in PCIR (see figure 6). This can be a deterrent for a peer to become a super-peer. For equally distributing the workload at the peers, we can build an internal DHT in each cluster and use it for distributing the responsibilities of the super-peer at all the cluster peers. Each of the cluster peers is assigned the task of publishing at the top DHT all the cluster keywords that hash into a specific range. While this introduces a small DHT for each cluster, the extra costs are less than the savings from PCIR. In particular, the maximum number of messages that need to be exchanged for the intra-cluster maintenance for a cluster C is $O(\|C\|^2)$, but since clusters are deliberately kept small, this cost is not high. Having a cost model, we can also compute the optimal cluster sizes that minimize the publishing cost for an expected keyword frequency distribution (i.e., Zipfian).

Dynamic Network Optimization: The performance of cluster-based PCIR is strongly dependent on the values of λ and μ. As already noted, if these values are high, better clustering is performed and the publishing value is reduced. On the other hand, the clustering cost increases in parallel to these values. While finding the cost model for the system is not difficult, theoretically deciding on the optimal λ and μ values which would minimize the overall cost (both clustering and publishing) requires a relation between the distribution of keywords in the peers and the number of distinct keywords in the cluster. For the empirical Zipf distribution, this relation is very difficult to estimate. Thus we are currently focusing on experiments which will allow us to see how *'far from the optimal'* is a setup with generic λ and μ values for different data collections. If this performance difference justifies dynamic network optimization we will continue our effort on estimating the relation between the number of distinct keywords in a cluster and the peers' keyword distributions.

The periodicity trade-off (index accuracy vs maintenance cost) is also very important on the system's configuration. A very small republishing period can increase the index accuracy in expense of the maintenance cost. Ideally the system should be able to adjust the length of the republishing period for providing the required accuracy at a minimal cost. For this, the system needs to keep statistics on the peer churn, and adjust the length of the publishing period accordingly.

Indexing throughput evaluation: Until now the experiments measure only the network cost, not the time for completing the index. It would be interesting to run the experiments on a large-scale distributed platform like Planetlab. This will allow us to account further network factors, like network distance and peer bandwidth, and see how important it is to take these factors into account. Furthermore, it will enable larger experiments with more peers and documents per peer.

Unstructured P2P systems: Some P2P scenarios are incompatible with DHTs. For instance, when the peer and document churn is very high, the cost of maintaining the DHT can be prohibitively expensive. Especially when the application scenario does not require exact answers or when the query frequency is very low, we consider replacing the *full-text* index with an unstructured network. Unstructured P2P systems have difficulties scaling to large networks because the IR quality in these systems suffers. In our approach, due to the peer clustering, the relevant peers for a query are all expected to belong in the same cluster (or few clusters). Thus, it suffices for our approach to find only a single link for routing the query inside the right cluster(s). Then, the query can be sent to all the peers of the cluster.

4. INFORMATION RETRIEVAL IN PCIR

4.1 Current IR algorithm

Our focus up to now was more on reducing the inverted index maintenance cost, and not so much on enabling advanced IR. Our current query execution algorithm resembles the keyword-based query execution used in previous systems, like, e.g., Minerva [4]. The query initiator looks up the query keywords in the DHT (Fig. 4) and retrieves the Peerlist for each. From the Peerlists, it discovers the most promising peers and routes the query to them. The peers independently execute the query and return their results to the query initiator.

The query execution algorithm is highly configurable. We currently use a simple cosine similarity as a ranking function for both peers and documents. It is however easy to replace it with CORI, BM25 or any other ranking function. Computing or estimating the IDF/IPF which is required from these ranking functions is a notorious issue, common to all the P2P systems. Any existing solution or approximation can be used orthogonally to PCIR.

Note that the DHT-based inverted index produced from PCIR is of the same quality and resolution as the one constructed by any flat DHT peer-granularity maintenance algorithm (e.g., Minerva): the detailed peer scores (not only the cluster-aggregated scores) for all the peers are published in the DHT. Thus, the information retrieval quality is not affected by the PCIR enhancement. Also, the query execution cost is the same as in other DHT-based approaches, because the promising peers are detected and queried directly from the query initiator without any intervention from the super-peers. As such, we do not perform a query execution evaluation in this work.

4.2 Tasks for enabling Advanced IR

We are now working towards enabling advanced P2P information retrieval with reasonable costs. For this goal, the proposed peer clustering architecture can prove very helpful.

Cluster hypothesis and cluster-based IR.

An approach we currently study stems from the cluster hypothesis: documents that are clustered together tend to be relevant to the same query requests [21]. We now investigate whether the cluster hypothesis applies to our scenario, and study the keyword distributions (frequencies) inside each cluster. If the cluster hypothesis is applicable, we can further limit the size of the DHT inverted index: instead of maintaining the full inverted index for query routing purposes, the peers can maintain an inverted index with cluster aggregated data, like minimum, maximum and average peer frequency values, or even histograms. The cluster-granularity inverted index can be used for routing queries to clusters. For instance, given a query, if the computed cosine similarity for a cluster using the maximum values is less than the computed cosine similarity for another cluster using the minimum values, then no peer in the first cluster needs to be queried. After the queries are routed inside the clusters, the super-peers can handle further routing of the queries at the cluster peers; they already have all the peer content summaries, so no further network cost is induced, and the cost of maintaining the DHT is reduced significantly.

Intra-cluster IR: Inside each cluster, we can also apply more complex IR techniques. The super-peers have all the necessary data for applying centralized IR techniques, like Latent Semantic Analysis, for all the cluster peers. And unlike previous P2P LSA implementations [20], this approach will not suffer from scalability issues; the clusters are deliberately kept small and manageable, thus the LSA computation for only the cluster data will have low computational requirements. For avoiding the overload of the super-peers during query execution, the LSA transformation can also be sent to all the cluster peers, so that any peer inside the cluster is equally able to answer and route any given query. Thus, query execution does not need to be routed through the super peers; any cluster peer will be able to execute it

or route it to the most relevant cluster peers.

Distributed Probabilistic IR.

Another approach with cluster granularity data involves probabilistic IR along the lines of the Decision Theoretic Framework (DTF). Nottelmann and Fuhr in [15] already successfully used DTF to enable a probabilistic selection of P2P digital libraries for query execution. Applying their work in PCIR, the super-peers can publish the contents of their clusters as compact term frequency distributions (we can approximate them with histograms or in a similar manner as in [15]). Then the query initiator will be able to make a formal decision on the top most promising clusters and on the top most promising peers per cluster to be queried.

We can further extend the DTF approach, so that it also takes into account the peer clustering and indexing cost (these costs are not relevant to the original work). Then, assuming a certain ratio of queries per republishing period, the super-peers can decide how detailed the cluster publishings need to be, so that the overall system cost is optimized. Such a model can also lead to formal decisions for the clustering step for optimizing the cluster structure (e.g., break too unfocused clusters or devote more resources on getting a better peer clustering result) so that querying can finish faster.

Distributed dimensionality reduction.

Another IR approach that we will consider is concept indexing [11]. Concept indexing was also used in the past for increasing the IR quality in clustered data. The advantage of concept indexing is that it can be used for both clustering of new peers, and also for answering queries. In addition, it is not sensitive to the quality of the clustering solution. Our approach is suitable for direct application of concept indexing, since clustering is already in place.

5. WORKSHOP FEEDBACK

The main feedback received from the workshop was for the importance of a theoretical system model, which can be used for minimizing the network cost. Namely, the performance of PCIR depends on 3 system parameters, λ, μ and the resolution of the bloom filter representations of the peers. We have shown that generic values for these parameters result in significant performance improvements compared to the flat DHT case.

In detail, we have seen that increasing the values of λ and μ, as well as increasing the resolution of the bloom filter representations for the peer contents, increases the probability of a peer to find and join the right peer cluster. However, this comes with an increase in the peer clustering network costs, which at some point outweigh the benefits of better peer clustering. A model can: (a) give the probability of correct clustering to occur, and (b) give the optimal values for λ and μ for minimizing the total network cost.

A second comment was about the process of breaking each peer to virtual peers. We currently employ the standard K-means algorithm, with a fixed $k = 3$ for all the peers. In real P2P networks, each peer will need to decide for the value of k based on its collection diversity and size. We are currently investigating this problem.

Acknowledgements

This work is performed under the supervision of Prof. Wolfgang Nejdl and Dr. Wolf Siberski.

6. REFERENCES

[1] Beagle++, http://beagle2.kbs.uni-hannover.de/.

[2] DELOS Network of Excellence, http://www.delos.info/.

[3] K. Aberer, F. Klemm, M. Rajman, and J. Wu. An architecture for peer-to-peer information retrieval. In *Workshop on P2P Information Retrieval*, 2004.

[4] M. Bender, S. Michel, G. Weikum, and C. Zimmer. The minerva project: Database selection in the context of P2P search. In *Proceedings of Datenbanksysteme in Business, Technologie und Web*, pages 125–144, Karlsruhe, Germany, 2005.

[5] M. W. Berry, S. T. Dumais, and G. W. O'Brien. Using linear algebra for intelligent information retrieval. Technical Report UT-CS-94-270, 1994.

[6] A. Crespo and H. Garcia-Molina. Semantic overlay networks for P2P systems. In *AP2PC*, 2004.

[7] S. Datta, C. Giannella, and H. Kargupta. K-means clustering over a large, dynamic network. In *Siam Conference of Data Mining*, 2006.

[8] A. Fast, D. Jensen, and B. N. Levine. Creating social networks to improve peer-to-peer networking. In *KDD*, pages 568–573, 2005.

[9] P. Haase, R. Siebes, and F. van Harmelen. Peer selection in peer-to-peer networks with semantic topologies. In *International Conference on Semantics in a Networked World (ICNSW'04)*, LNCS, 2004.

[10] K. Hammouda and M. Kamel. HP2PC: Scalable hierarchically-distributed peer-to-peer clustering. In *SIAM Conference on Data Mining (SDM07)*, 2007.

[11] G. Karypis and E.-H. Han. Fast supervised dimensionality reduction algorithm with applications to document categorization and retrieval. In *Proc. 9th ACM CIKM*. ACM Press, New York, US, 2000.

[12] F. Klemm and K. Aberer. Aggregation of a term vocabulary for P2P-IR: A DHT stress test. In *DBISP2P*, pages 187–194, 2005.

[13] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, 2004.

[14] J. Li, B. Loo, J. Hellerstein, F. Kaashoek, D. Karger, and R. Morris. On the feasibility of peer-to-peer web indexing and search. In *2nd International Workshop on Peer-to-Peer Systems*, 2003.

[15] H. Nottelmann and N. Fuhr. A decision-theoretic model for decentralised query routing in hierarchical peer-to-peer networks. In *ECIR*, pages 148–159, 2007.

[16] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *Proc. of PODS:Symposium on Principles of Database Systems*, 1998.

[17] O. Papapetrou, W. Siberski, W.-T. Balke, and W. Nejdl. DHTs over peer clusters for distributed information retrieval. In *21st International Conference on Advanced Information Networking and Applications (AINA)*, pages 84–93, 2007.

[18] O. Papapetrou, W. Siberski, W. Nejdl, and W.-T.

Balke. A combination of DHTs and peer clustering for distributed information retrieval. Technical report, 2007. http://www.l3s.de/~papapetrou/reports/dhtclusters.pdf.

[19] I. Podnar, M. Rajman, T. Luu, F. Klemm, and K. Aberer. Scalable peer-to-peer web retrieval with highly discriminative keys. In *23rd International Conference on Data Engineering*, pages 1096–1105, Istanbul, Turkey, 2007.

[20] C. Tang, Z. Xu, and M. Mahalingam. pSearch: information retrieval in structured overlays. *SIGCOMM Comput. Commun. Rev.*, 33(1), 2003.

[21] C. van Rijsbergen. *Information Retrieval second edition*. Butterworths, 1979.