



Supporting Information Retrieval in Peer-to-Peer Systems

Wolf-Tilo Balke
L3S Research Center
University of Hannover
Germany

Schloß Dagstuhl, 27.03.06



Overview

- Content Searching in Peer-to-Peer Applications
- Index structures for Query Routing
- Supporting Effective Information Retrieval
- Summary and Conclusion



Applications of P2P Technology

■ File sharing was main application area

- Retrieval by simple metadata matching or keyword lookups
- Exact match model, substring matching
- Music sharing (Napster, Gnutella, KaZaa,...)



- Mp3 files, small videos, etc.
- Metadata like encoding quality or playing time
- Keywords like song title, artist, and full text description



Information Retrieval in P2P Systems

■ Information Retrieval has to deal with complex text documents

- Meta-data can only capture some aspects of a documents, but not anticipate all possible semantic searches
 - E.g. sports-related news item, but no names, locations, etc.
- Support for full-text searches needed
- Ranked retrieval model
 - Similarity between documents
 - Degree of match wrt. query or user's information needs

■ Find the best-matching document from the best-connected peer

- Unlike in file sharing emphasis is on the document quality
- If there are multiple sources offering similar quality documents, choose best peer according to connection, etc.



Challenges

- Challenges due to distributed nature of P2P environment
 - Efficient **query evaluation** scheme
 - How to disseminate a peer's query?
 - Central inverted index of documents is expensive to maintain
 - Simple flooding of all queries is not scalable, if not just some matching, but 'best' documents have to be found
 - Dealing with **network churn**
 - A peer can always alter the set of documents offered, or significantly change individual documents
 - Peers may join and leave the network, i.e. whole document collections may disappear, or can be added



Challenges 2

- Integration of **collection-wide information**
 - Peers are not able to calculate IR-style scorings from local knowledge
 - Constant dissemination of collection-wide information needs a lot of bandwidth
 - Example: Popular IR measure TFxIDF:

$$s_q(D) := \frac{TF_q(D)}{\max_{t \in D}(TF_t(D))} * \log\left(\frac{N}{DF_q}\right)$$



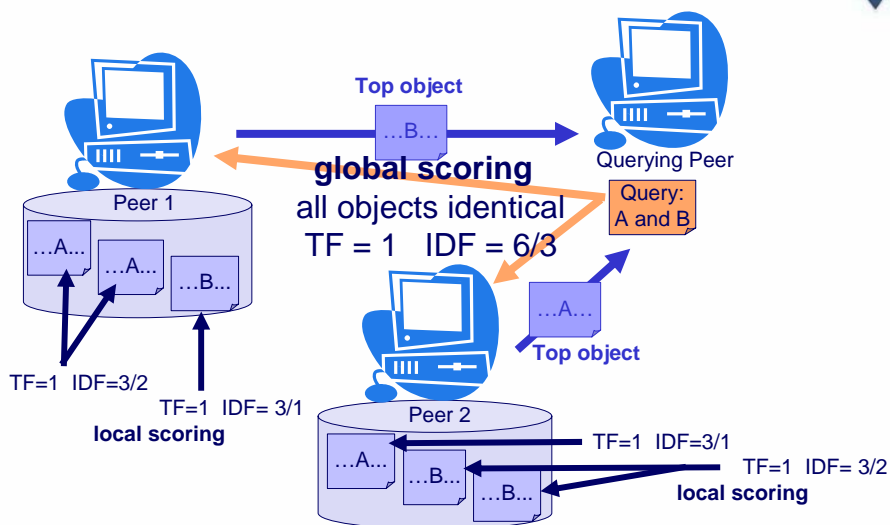
Example: Collection-wide Information

- Different news collections, query on keyword 'basketball'

- General news collection, e.g. **Los Angeles Times**
 - Many articles, only few about basketball, therefore IDF low
 - Keyword discriminates well between articles



- NBA news collection
 - Few articles, almost all about basketball, therefore IDF high
 - Keyword hardly discriminates between articles





Distributed Information Retrieval

- Distributed information retrieval techniques grew increasingly important for searching Web sources
 - Abstracts of information sources
 - To support distributed retrieval sources have to register abstracts or keyword sets
 - Such abstracts can be compactly represented by Bloom filters
 - Abstracts can either be kept in a central repository or distributed by gossiping algorithms, e.g. PlanetP [F. Cuenca-Acuna et al., '03]
 - Collection selection
 - Having no central index needs a sophisticated way of choosing the most promising collections for querying
 - E.g. CORI [J. Callan, et al. '95], GLOSS [L. Gravano, et al. '99]



Index Structures for Query Routing

- Traditional index structures cannot be readily employed in P2P systems
 - High degree of distribution
 - High degree of volatility (churn) } High degree of index maintenance
- Distributed paradigms needed to route queries to appropriate peers
 - Simple flooding method does not scale
 - Distributed hash table lookup
 - Using indexed routing information
 - Using shortcut overlays



Using Distributed Hash Tables for IR

■ Distributed Hash Tables (DHTs)

▪ **Cool:**

- Route queries to appropriate peers with number of hops logarithmic in network size, no peer needs to maintain more than logarithmic amount of routing information,...

▪ **But...**

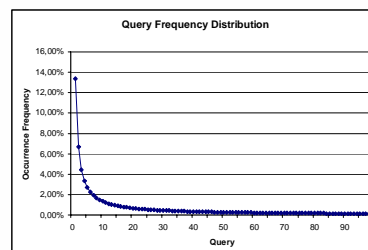
- Exact match queries only
- All content has to be (un-)published, if peers join/change/leave
- Documents added/removed contain a lot of different terms to be (un-)published. Many index peers have to be addressed
- Conjunction of query terms needs to access many peers, but there is still no guarantee that a *single* document with the conjunction exists



DHTs for Information Retrieval

■ Improvement: Hybrid P2P infrastructures (B. T. Loo et al., '04)

- Efficiency of DHT is worst, if highly replicated items are requested
 - Experiments show worse behavior than flooding, degrading with churn
- Querying and content allocation follow Zipf-distribution
 - Only few highly replicated and often queried items
 - 'People are looking for hay, not for needles' (S. Shenker)



- Hybrid P2P infrastructures use DHTs only for the less replicated and rarely queried items, all other queries are flooded



Routing Indexes for Information Retrieval

- Routing indexes are local collections of (key, peer) pairs
 - *Key* is either a keyword or a query
 - *Peer* is the address of a peer that either offers relevant results, or routes the query to other peers with relevant result
- In contrast to flooding only 'interesting' directions are queried
 - Often distinguished between links in the default network (directions of content providers) and overlay structure of direct links to content providers ('shortcuts')
- First introduced to choose best neighbors in the default network for query forwarding by [A. Crespo, et al. '02]
 - Index maintenance is of local nature and index coverage is usually high due to Zipf distribution of requests
 - Correctness of index is influenced by network volatility/churn



Routing Indexes Example: top k queries

- Example for routing indexes in structured P2P networks with super-peer backbone that holds routing indexes
- Progressive P2P top-k algorithm [W. Balke et al., '04]
 - If query q is indexed, distribute query and collect results
 - Otherwise flood query and
 - Compute ranks at local peers
 - Merge results at super-peers
 - Use statistics for new entry in routing index (routing information, collection-wide information, etc.)



Locality-Based Routing Indexes

- Refinement of routing indexes by social metaphors
- Similar retrieval process like in real life
 - Every person has only limited knowledge of the environment
 - Who knows about a certain topic?
 - Who might know other people who know about the topic?
 - Try to build (short) 'chains of acquaintances' that will bring you close to the requested information
- Aims at building 'social networks' as overlays
- Peers semantically connected by certain topics form '*small world networks*', e.g. [S. Milgram, '67; J. Kleinberg, '00]
- Paradigm of *interest-based locality*
 - If a peer has relevant content for a user's query, it very often also has some other content that this user might be interested in



Locality-Based Routing Indexes

- For information retrieval in P2P network this enables new routing in interest-based overlay structures
 - Route queries to peers with documents matching semantically close queries
 - Traces on practical data collections show that
 - peers get well-connected
 - the overlay graph shows highly-clustered characteristics with a small minimum distance between any two nodes
 - 'Overhearing' of communications routed through a peer can be used to enhance its local index
 - Randomly sending queries also to peers from the default network helps to extend knowledge and can remedy the effect of network churn



Supporting P2P Information Retrieval

- P2P information retrieval has to deal with the trade-off between
 - Efficient local maintenance of statistics / index information, where information can be stale (incorrect)
 - Expensive global maintenance of statistics / index information, where information always is accurate
- Needed is 'just the right level' of dissemination of statistics to still guarantee a 'sufficiently effective' retrieval
- Some techniques help to support efficient retrieval
 - Providing adequate collection-wide information
 - Estimate document overlap between peers
 - Pre-structure collections by categories / taxonomies



Providing Collection-Wide Information

- Collection-wide information (CWI) is important for retrieval quality, but cannot be calculated locally like e.g. IDFs
 - Some systems like e.g. PlanetP, do not use CWI directly, but circumnavigate the problem by using an inverted peer frequency

$$IPF_t := \log\left(1 + \frac{N}{N_t}\right)$$

where N is the number of all peers and N_t is the number of peers offering documents on term t

- If summarizations of peers (abstracts) are eagerly disseminated, each peer can locally decide values for N and N_t
- The relevance of peers in multi-keyword queries is simply the sum of IPFs for the individual terms



Providing Collection-Wide Information

- Tests in Web information retrieval, e.g. [C. Viles & J. French, '95], show that CWI stays relatively stable over the whole collection of Web Sites even in the presence of churn
 - Only joining/leaving corpora on completely new topics result in significant change
- Indexing CWI in a similar way as the routing information for queries is possible [W. Balke et al., '05]
 - In structured networks CWI can be aggregated along the backbone and indexed CWI can be distributed together with the query
 - New queries have to be flooded/routed twice
 - The first flooding collects and aggregates CWI
 - The second one provides the correct CWI for local scorings
 - Non-expired indexed CWI can always be used when available



Estimating the Document Overlap

- Assessing the novelty of collections also supports retrieval quality
 - Pre-computed statistics about expected result quality in each collection can minimize the number of queried collections
 - Choosing collection with high overlap for querying will not improve result sets sufficiently to justify the access costs
 - Estimated overlap measure e.g. given in [M. Bender et al., '05]
- The novelty of a collection can only be calculated with respect to some reference collection(s)
 - e.g. those collection(s) already in a peers local routing index



Prestructuring Collections

- Retrieval in P2P systems generally considers two basic paradigms
 - Fulltext-based queries
 - Metadata-based queries
- Integrating these paradigms can support retrieval effectiveness
 - Structuring document collections
 - Disambiguation of query terms
- Peers often host collections of similar documents, e.g. similar kind of information (newspaper articles, etc.) on similar topics, etc.
 - Scalability and successful use of statistics are strongly improved, if a common system of categories to classify the documents can be used

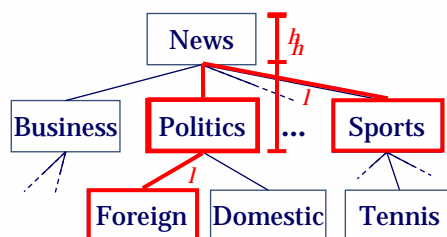


Prestructuring Collections

- Topical similarity within a taxonomy is defined by [Y. Li et al., '03]

$$sim(c_1, c_2) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}}$$

- l : shortest path between categories c_1 and c_2
- h : level of common subsumer
- Common values $\alpha = 0.2$, $\beta = 0.6$ (experimentally determined)



$sim(\text{Politics}, \text{Sports})$:

$l = 2$
 $h = 2$
 $sim = 0.88$



Summary

- In today's P2P systems only exact match keyword retrieval is prevalent (usually on meta-data)
- Information retrieval in P2P scenarios is needed
 - Individual, loosely coupled document collections need fulltext retrieval and ranking techniques
 - Applications range from shared working environments e.g. in project groups, to distributed digital libraries
- Almost all IR systems use at least some global statistics, but in P2P infrastructures the dissemination of necessary statistics becomes a performance bottleneck
 - Trade-off between *cached, but sometimes stale statistics* and *new, but expensively updated statistics* needs to be managed
 - How much staleness does a 'sufficient' retrieval effectiveness allow?



Summary

- Choosing the 'right' collections for querying improves retrieval efficiency
 - Those containing most promising documents with little overlap
 - Small worlds offer quick connections to semantically close collections
- Query routing indexes can handle a certain amount of network churn while providing results of sufficient quality
 - Local indexes can be efficiently maintained
 - Can exploit advantages by Zipf-distributed content allocations and querying behavior
 - Need to contact only small numbers of peers
- Supporting techniques like efficient CWI estimation/dissemination or taxonomies of document categories