

# Standing On the Shoulders of Peers: Caching in Peer-to-Peer Information Retrieval

Christian Zimmer, Srikanta Bedathur, and Gerhard Weikum  
 Max-Planck Institute for Informatics, Saarbrücken, Germany  
 {czimmer, bedathur, weikum}@mpi-inf.mpg.de

## Peer-to-Peer Information Retrieval

State-of-the-art P2P IR systems suffer from their lack of **response time guarantees**, especially with scale. To address this issue, a number of techniques for caching of multi-term inverted list intersections and query results have been proposed recently. Although these enable speedy query evaluations with low network overhead, they fail to consider the **potential impact of caching on result quality improvements**.

We propose the use of a **cache-aware query routing in structured P2P IR setting, that exploits the availability cached results to speed up query execution and to continuously improve the result quality**.

## Minerva System Architecture

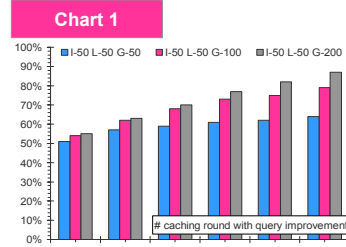
The Minerva system is a fully operational distributed search engine consisting of autonomous peers with local document collections. A conceptually global but physically distributed directory layered on top of a Chord-style distributed hash table (DHT) manages aggregated metadata information about the peers local knowledge in compact form. Each peer is responsible for a randomized subset of terms and stores peer-lists of posts (per-term summaries).

Figure 1 and 2 show the two main steps of the query execution:

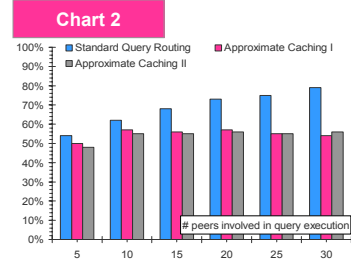
- > **Metadata Retrieval:** The querying peer sends peer-list requests to the directory for looking up promising remote peers, for each query term separately. Using a peer scoring function with the collected metadata, the querying peer determines the most promising peers (e.g., CORI).
- > **Local Result Retrieval:** The querying peer sends the complete query to the selected peers. These peers return their local query results such that the query initiator can merge them to one final query result.

### References:

- [1] M. Bender, S. Michel, C. Zimmer, P. Triantafillou, and G. Weikum: Improving Collection Selection with Overlap-Awareness. *SIGIR, 2005*.
- [2] S. Michel, M. Bender, N. Ntarmos, P. Triantafillou, G. Weikum, and C. Zimmer. Discovering and Exploiting Keyword and Attribute-Value Co-Occurrences to Improve P2P Routing Indices. *CIKM, 2006*.



**Query-Result Improvement:** Relative recall for different local, global and ideal result sizes increases with the number of rounds.



**Approximate Caching:** Relative recall comparison of the standard query routing with two approximate caching scenarios

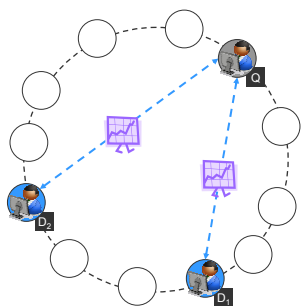
## Result Caching in the Directory

We have developed two different caching strategies using a distributed directory storing term-peer statistical metadata based on structured P2P overlay network:

- > **Exact Caching (EC)** is the P2P counterpart of traditional centralized result caching where cached results are used only if a **new query exactly matches the earlier query**. Figure 3-5 show the extensions to the standard query execution: the cached result is stored in the directory in a routing-aware manner (Figure 3); in the metadata retrieval, the querying peer sends the complete query to the directory peers such that the cached result can be detected (Figure 4); if the querying peer wants to improve the result-quality, it asks additional peers not involved in the cached result (Figure 5). After that the cached result gets updated. Chart 1 shows experimental results.
- > **Approximate Caching (AC)** aggressively reuses cached results for subsets of the query to produce an approximate result with limited query dissemination. Chart 2 shows experimental result for different scenarios: **AP-I** considers the cached results of all sub-queries, **AP-II** the results of all two-term sub-queries.

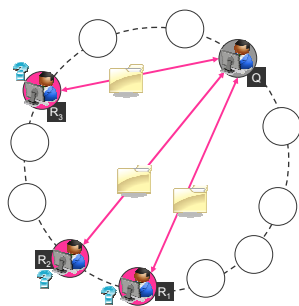
**“Standing on the shoulders of peers gives not only faster execution, but also higher result quality”**

Figure 1



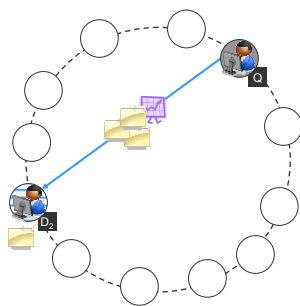
**Metadata Retrieval:** The querying peer Q sends peer-list request to the directory peers  $D_1$  and  $D_2$  returning the metadata concerning the query terms.

Figure 2



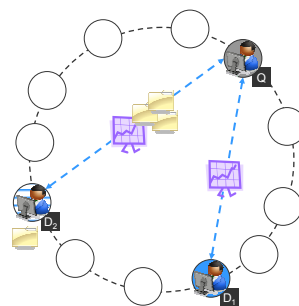
**Local Result Retrieval:** Q sends the complete query to the selected most promising peers.  $R_1$ ,  $R_2$  and  $R_3$  return their local top-k results.

Figure 3



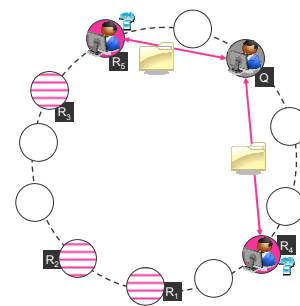
**Storing Cached Results:** Q sends the merged query result to one directory peer. This peer stores the cached result in a routing-aware manner.

Figure 4



**Retrieving Cached Results:** Instead of retrieving only metadata, the directory peers also return the cached result for the complete query.

Figure 5



**Additional Peer Inclusion:** To improve the result-quality of the cached result, the querying peer asks additional peers and merges their results.