



Exploiting Replication in Peer-to-Peer Search Over Distributed Digital Libraries

Christian Zimmer, Srikanta Bedathur, Christos Tryfonopoulos, Gerhard Weikum

Max-Planck Institute for Informatics, Saarbrücken, Germany

<http://www.mpi-inf.mpg.de>

Outline of the Talk



- Motivation
- Distributed Hash Tables (DHTs)
- System Architecture
- Replication Strategies
 - Successor-Directory Replication (SDR)
 - Top-Result Replication (TRR)
- Experimental Evaluation
- What about Caching?!
- Conclusions & Future Work

Motivation



Basics

- **Potential** of P2P systems:
 - scalable, efficient, resilient to failures and dynamics.
- **P2P Web Search** benefits from intellectual input of users:
 - click streams, query logs, bookmarks, etc.
- **Two conflicting Challenges:**
 - providing high quality results (recall & precision).
 - enabling high scalability (number of participating peers & huge amounts of data).
- **Minerva Search Engine** aims for high quality results by ensuring high scalability.

Motivation (con't)



Fundamental issues of all distributed P2P networks

- **Churn**
 - joining and leaving of peers
 - unknown rates
 - without notifying network

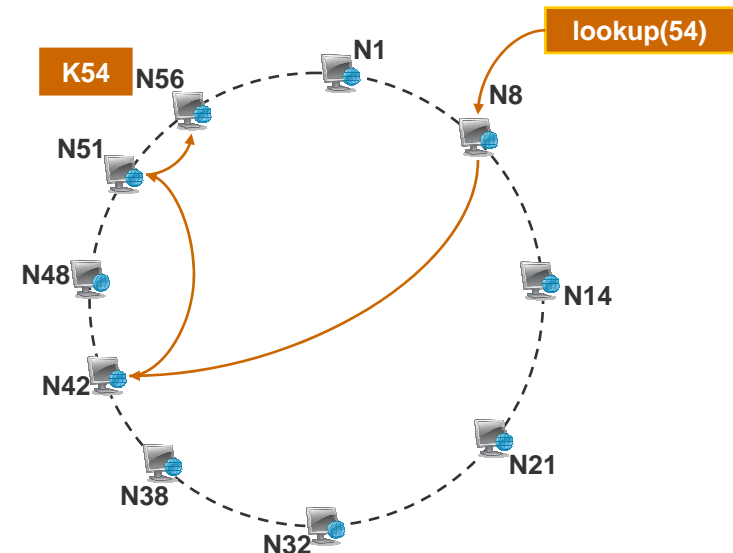
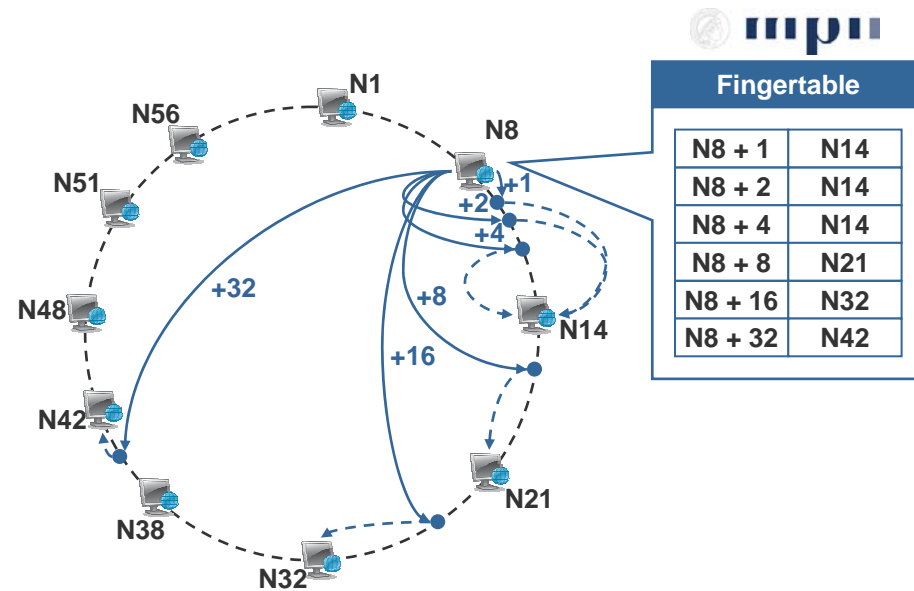
- **Data Dynamics**
 - new data arrivals and modifications
 - disappearing of data

Our approach: **Replication**

Distributed Hash Tables (DHTs)

Distributed Hash Tables (DHTs)

- Minerva search engine based on **Distributed Hash Table (DHT)** to achieve scalability, fault-tolerance, and robustness.
- Second generation of **structured overlay networks**.
- Minerva uses **Chord** with **fingertables**.
- Data items distributed to nodes using **consistent hashing**.
 - $id = Hash(key)$
- Lookup-method** to find location of data items with given *key* in $O(\log N)$ hops.



System Architecture

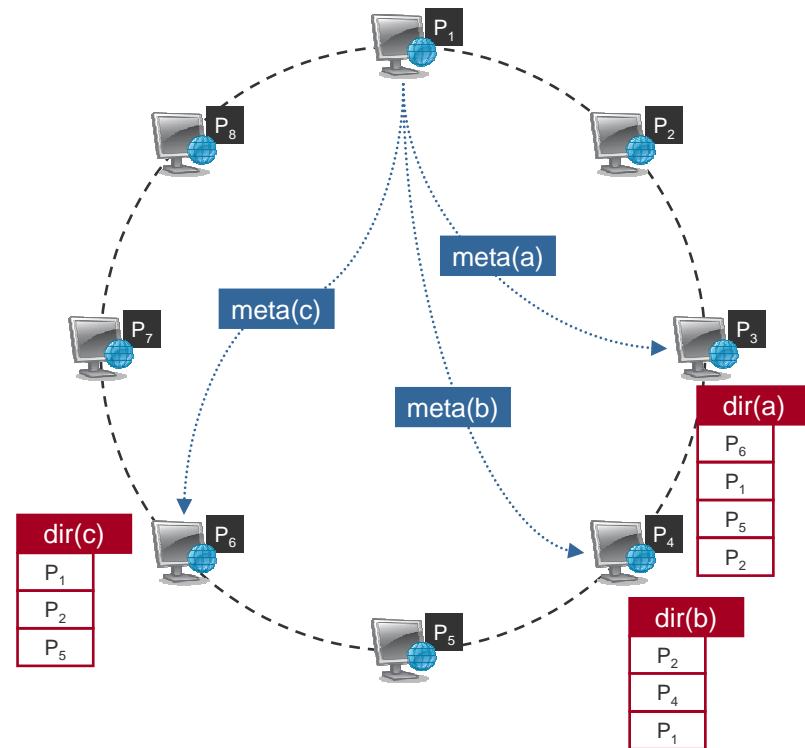


Local Search Engines

- Peers build **local indexes**.
- Peers can execute queries on local indexes.

Global Directory

- Distributed directory** layered on top of distributed hash table (DHT).
- DHT **partitions term space** such that each peer is responsible for subset of terms.
- Peers distribute **per-term summaries** (Posts) to global directory (size of the index, number of documents containing this term, etc.).
- Directory manages **aggregated statistical information** in compact form.

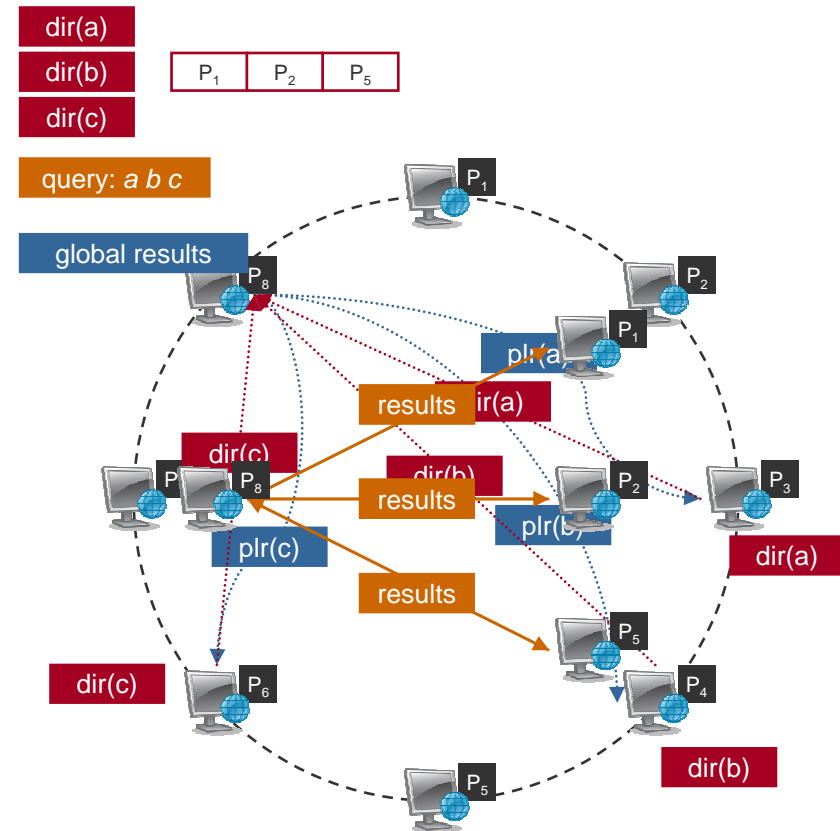


System Architecture (con't)



Query Execution

- Multi-term query *a b c*.
- **Peerlist requests** to retrieve metadata from directory.
- Compute **most promising peers** for complete query (e.g., CORI, DTF).
- Complete **query forwarded** to these peers executing query locally.
- Local results returned and **merged to global query result**.



Replication Strategies



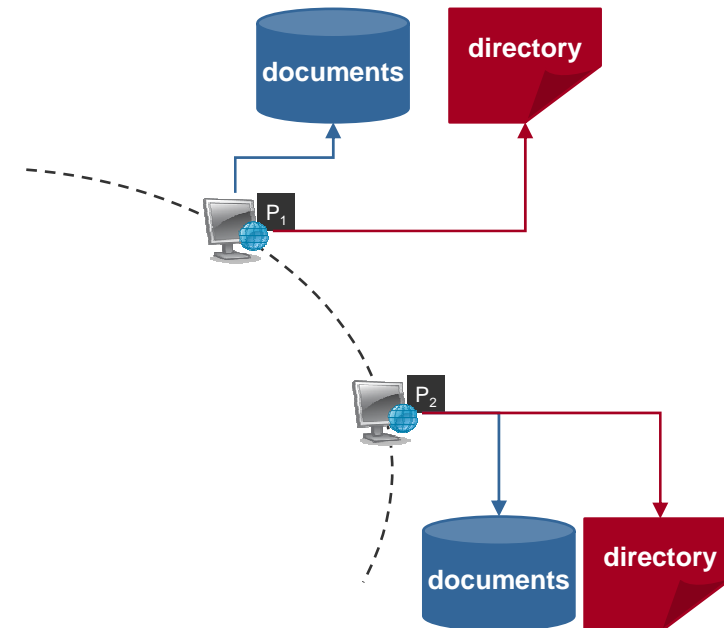
What is Replication?

- Provision of redundant resources (e.g., documents) in the network.
- **Goals:** reliability, high availability of data, fault-tolerance, load-balancing, performance and response time improvements, query result improvements.

Possibilities of Minerva System Architecture

- *data replication* (documents)
- *directory replication* (metadata)

Two replication strategies in Minerva



Replication Strategies (con't)



Successor-Directory Replication (SDR)

- Ensures **directory replication**.
- Peerlists are stored on more than one directory peer.

Alternative Approaches:

- **Several DHT-networks** (requires overhead to manage the networks)
- **Several lookup-functions** in the same DHT (different routings needed)
- **SDR**: Peerlists are also stored at successors; directory peers forward per-term summaries to direct successors in Chord-DHT
 - no overhead for several networks
 - no additional routing needed
 - consistent hashing ensures availability transparently
- **Replication degree**: number of successor forwardings (number of replications)

Replication Strategies (con't)



Top-Result Replication (TRR)

- Performs **data replication**.
- **Perfect strategy**: replicate each document at every peer (communication and storage cost explode).
- Strategy has to balance between replication cost and gain in availability.

- **TRR**: Replicate only the *important documents*, i.e., query results.
 - Querying peer downloads the **top-result documents** of a query requests.
 - Replicated documents added to local collections and directory statistics.
- **Need for Controlling Mechanism**: prevents excessive and unnecessary replication
 - approximate solution to avoid additional message overhead
 - counting the document appearance within the local query results

Experimental Evaluation



Setup

- Web data collection with **253,875 documents**.
- Categorized to one of **10 categories** (e.g., Travel Sports, or Finance).
- **50 digital library peers** (5 peers per category) with 1,000 category documents and 1,000 documents from other topics.
- **20 multi-term queries** (strong representatives of the categories) with two, three, or four query terms (e.g., *biology institute*, *opera concert voice*).

Measurements

- **Relative Recall**: fraction of *ideal result documents* included in results of P2P query processing.
- **Ideal results** as result of centralized query execution including combined document collection.
- **Influencing parameters**: # ideal results ($I=50$), # global results ($G=100$), and # local results ($L=50$).

Experimental Evaluation (con't)



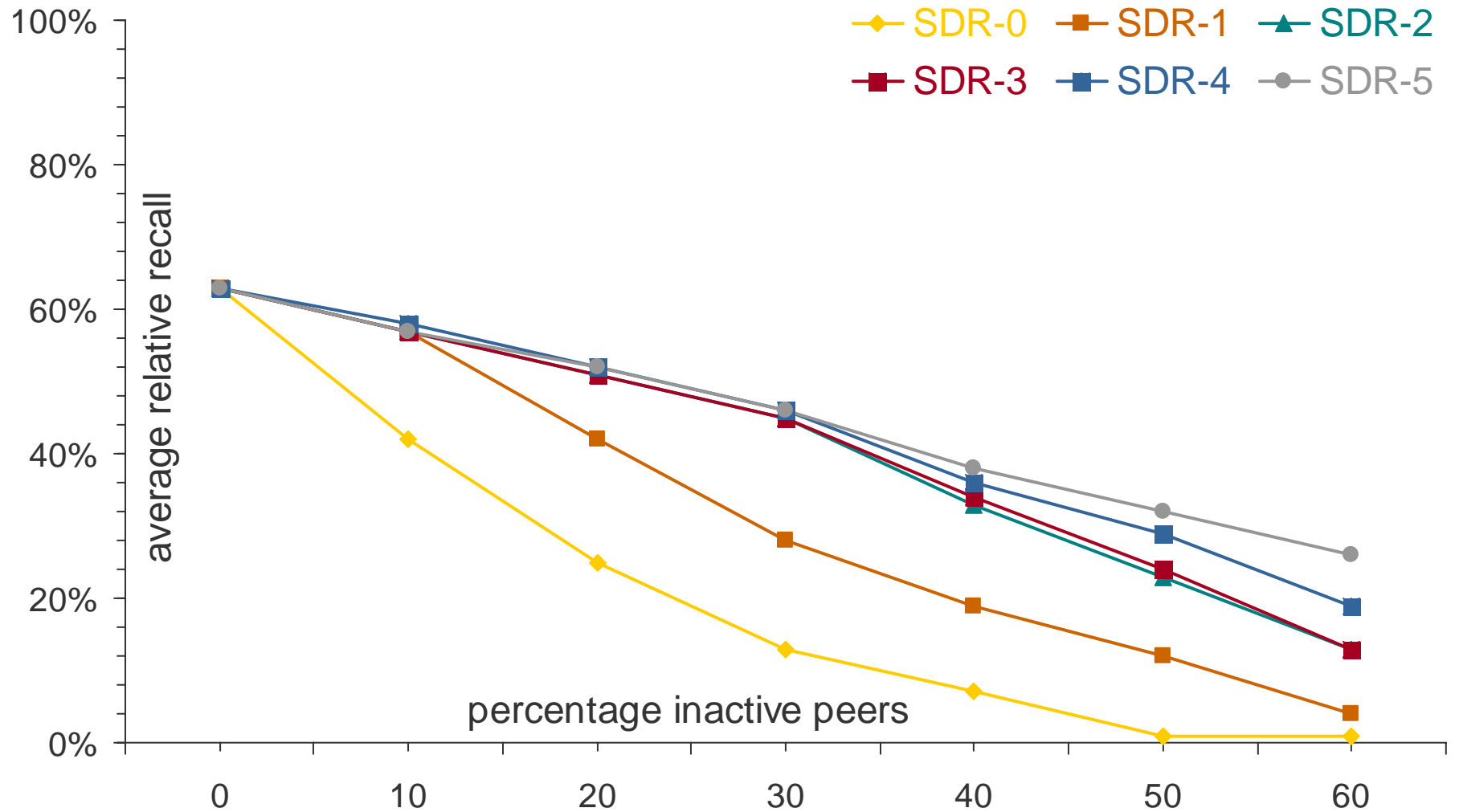
Experimental Series

- **Relative Recall depending on number of inactive peers** in system.
- Query execution considers **local results of maximum 10 peers**.
- Recall values averaged over **5 experimental runs** with **random inactive peers**.

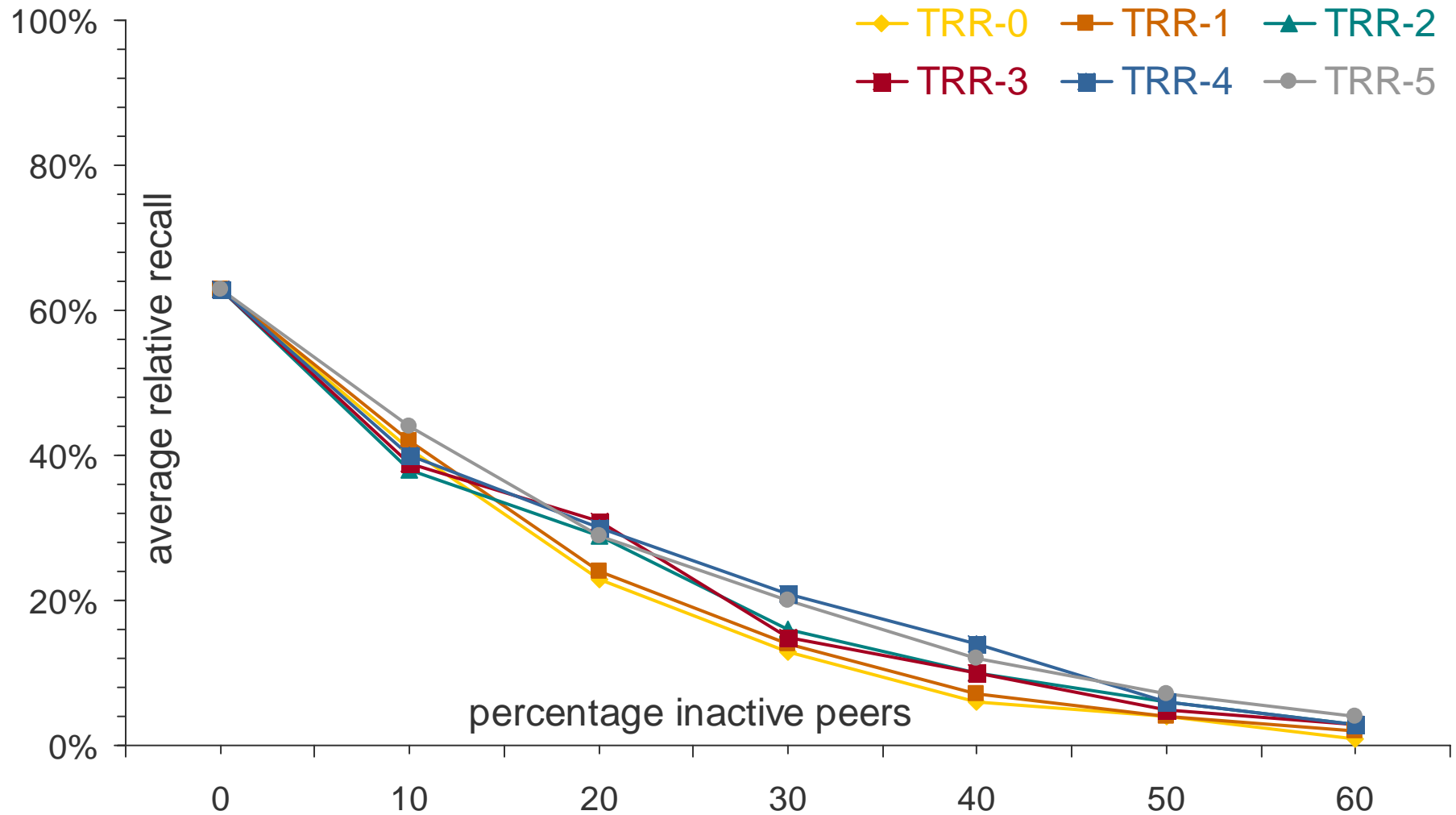
Four Scenarios

1. **SDR only**: replicated directory statistics and sudden disappearance of peers.
2. **TRR only**: no directory replication and certain number of random peers have replicated top-result documents.
3. **TRR & SDR**: combines both replication strategies.
4. **TRR & SDR enhanced**: considers the expected number of inactive peers.

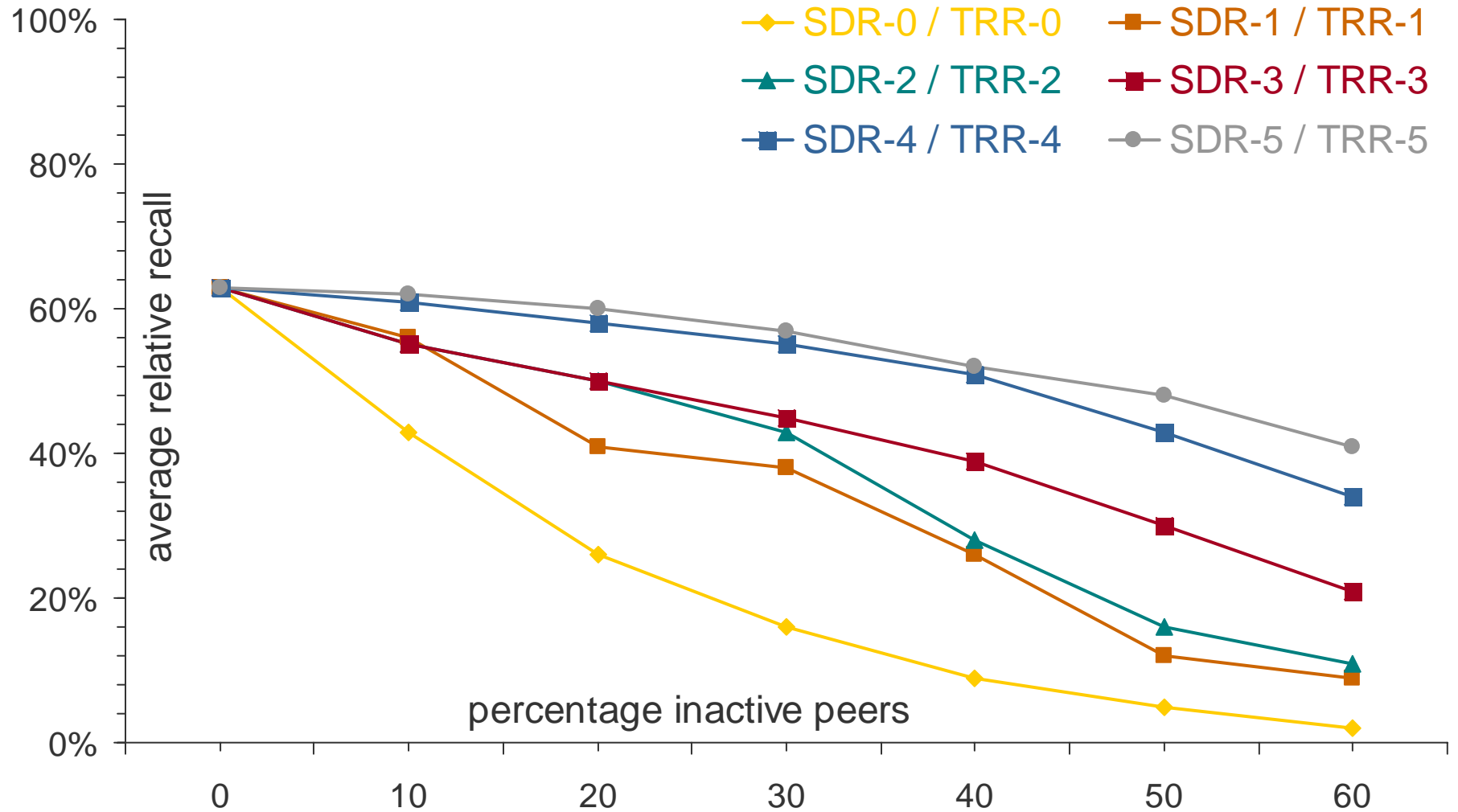
Experimental Evaluation (con't)



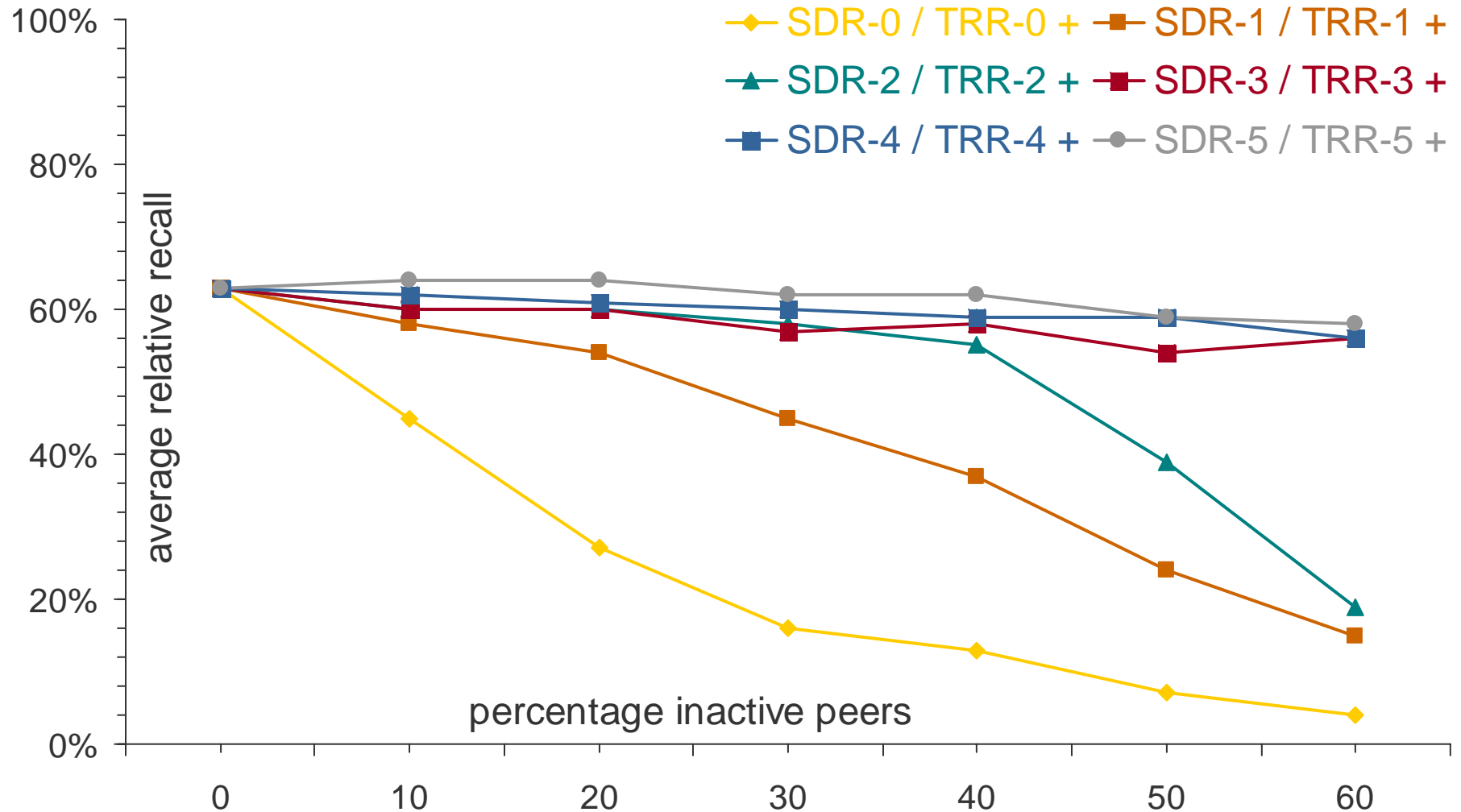
Experimental Evaluation (con't)



Experimental Evaluation (con't)



Experimental Evaluation (con't)



What about Caching?



DASFAA 2008

Flood Little, Cache More: Effective Result-reuse in P2P IR Systems

Christian Zimmer, Srikanta Bedathur, and Gerhard Weikum

- Cache-aware query routing scheme
 - reduces response delays
 - improves result quality

Two Caching Approaches

- **Exact Caching (EC) Strategy:** P2P counterpart of traditional result caching
- **Approximate Caching (AC) Strategy:** aggressively reuse cached results of query subsets

Conclusions & Future Work



Conclusion and Future Work

- Study about **replication strategies** to avoid information loss.
- Combination of **Successor-Directory Replication** and **Top-Result Replication** managed to avoid dramatical decrease in relative recall.
- **Caching** complements replication (with other/new goals).
- **Future Work:**
 - automatic **adaptation of replication degree**.
 - other replication strategies with **different goals** (e.g., load balancing).

Thank You For Your Attention!

Questions or Comments?