

Improving Collection Selection with Overlap Awareness in P2P Search Engines

Matthias Bender Sebastian Michel *Peter Triantafillou*

Gerhard Weikum *Christian Zimmer*

Max-Planck-Institut Informatik and University of Patras

Overview

- Motivation
- Structured Peer-to-Peer Systems
- Design Fundamentals
- Query Routing
- How to estimate "Novelty"
- The iterative Query Routing Algorithm
- Experimental Evaluation
- Conclusion
- Future Work

Why P2P Web Search?

Ultimate goal: “Distributed Google” to break information monopolies

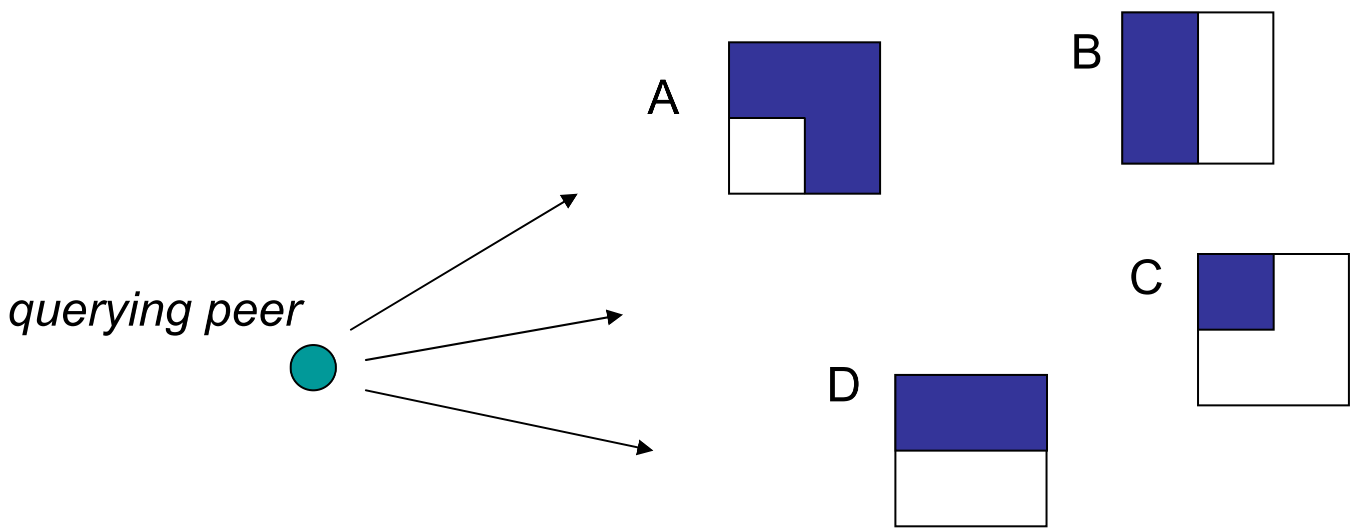
- P2P approach best suitable
  - large number of peers
  - exploit mostly idle resources
  - intellectual input of user community

- Related to distributed IR, but some additional aspects
  - high dynamics
  - each peer has its own collection
  - peers are independently crawling the web

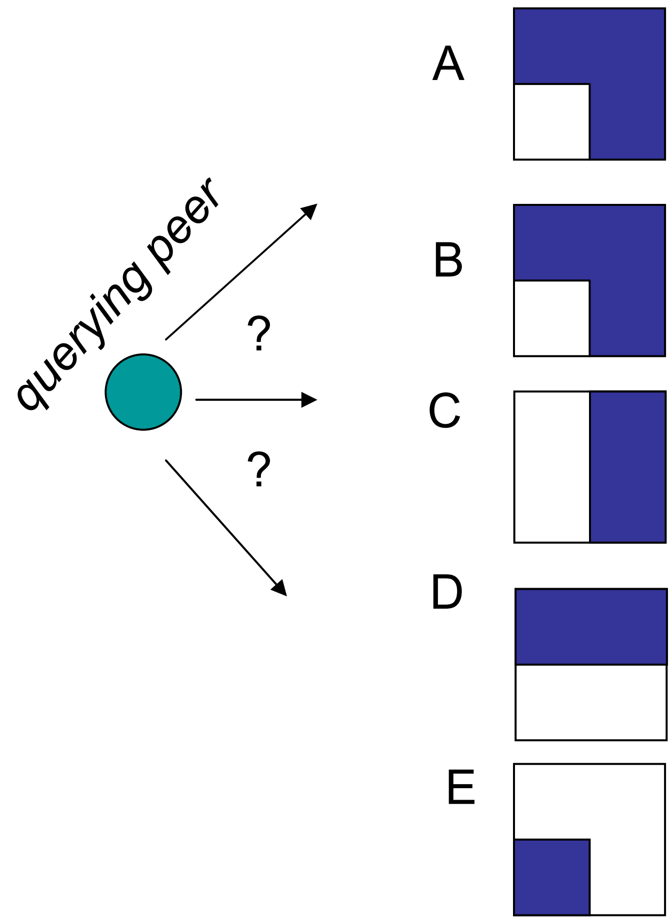
Why Overlap Awareness?

- Large scale distributed web search
- Peers are independently crawling the web

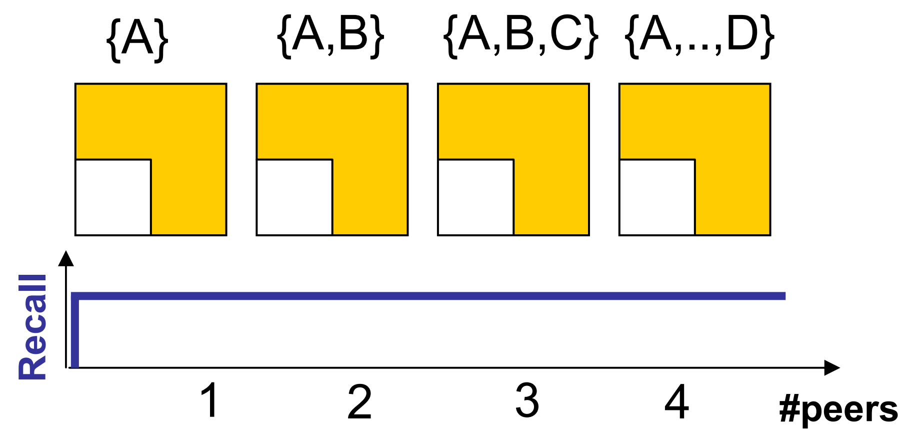
⇨ overlapping collections



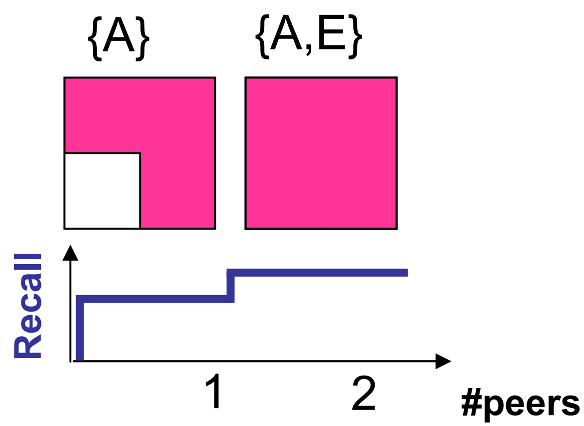
Why Overlap Awareness?



naïve routing strategy:

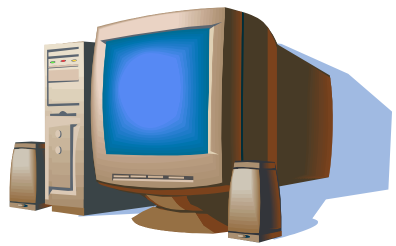


overlap aware routing strategy:



Structured P2P Systems

- Distributed Hashtable (DHT)
- Highly efficient support of one “simple” method

lookup(key) → 

in *O(log n)* routing hops!

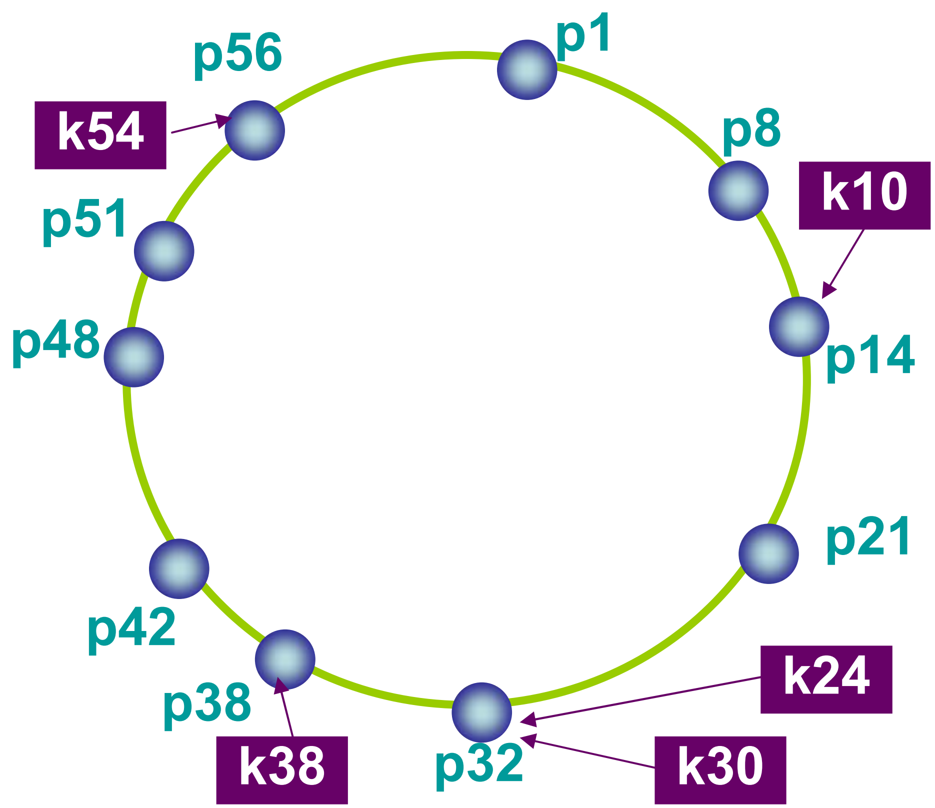
+ robustness to load skew, failures, dynamics

- Chord: I. Stoica et al.
- CAN: S. Ratnasamy et al.
- P-Grid: K. Aberer

e.g., the hash value of a term can serve as the key

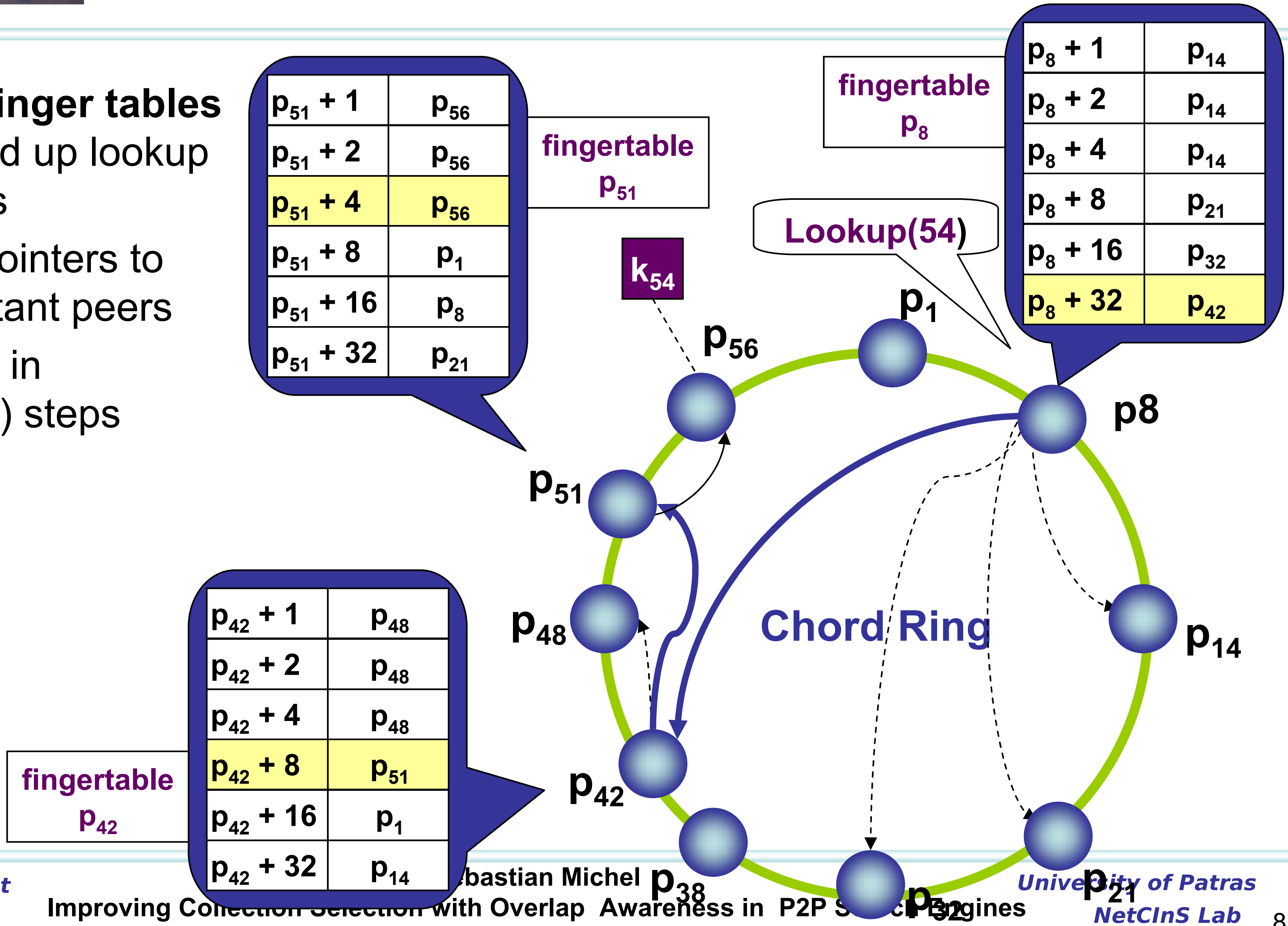
Chord

- Peers and keys are mapped to the same cyclic ID space using a hash function
- Key k (e.g., hash(file name)) is assigned to the node with key p (e.g., hash(IP address)) such that k ≤ p and there is no node p‘ with k ≤ p‘ and p‘<p



Chord

- Using **finger tables** to speed up lookup process
- Store pointers to few distant peers
- Lookup in O(log *n*) steps



p_{51} + 1	p_{56}
p_{51} + 2	p_{56}
p_{51} + 4	p_{56}
p_{51} + 8	p_{1}
p_{51} + 16	p_{8}
p_{51} + 32	p_{21}

p_{8} + 1	p_{14}
p_{8} + 2	p_{14}
p_{8} + 4	p_{14}
p_{8} + 8	p_{21}
p_{8} + 16	p_{32}
p_{8} + 32	p_{42}

p_{42} + 1	p_{48}
p_{42} + 2	p_{48}
p_{42} + 4	p_{48}
p_{42} + 8	p_{51}
p_{42} + 16	p_{1}
p_{42} + 32	p_{14}

Design Fundamentals

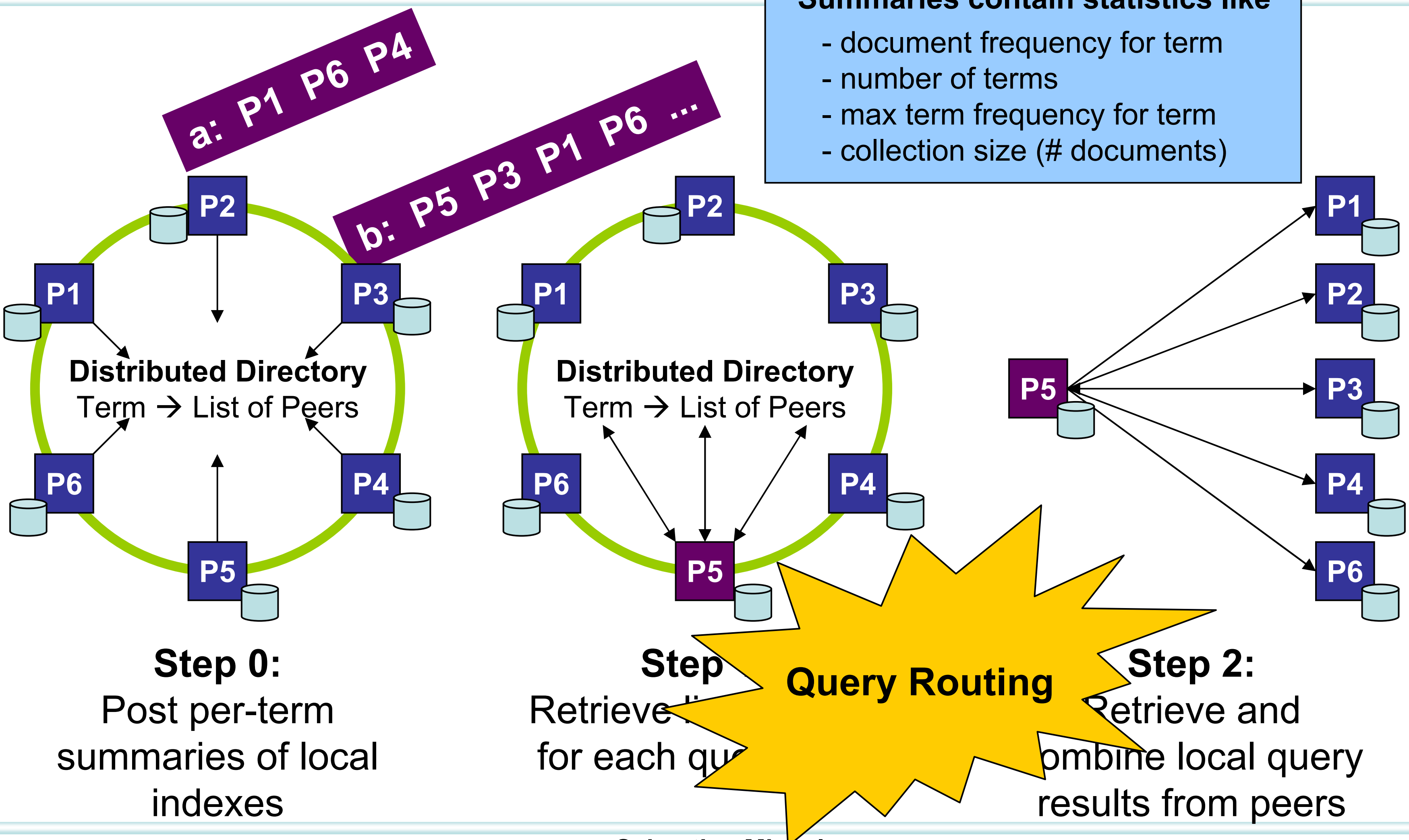
- Peers autonomously and independently crawl the web, according to their interest profiles, to build local indexes

- Peers share *meta*data about local indexes to form a (conceptually global) but physically distributed directory; layered on top of a DHT (e.g., Chord)

Project homepage: ***http://www.minerva-project.org***

Design Fundamentals (2)

Summaries contain statistics like
- document frequency for term
- number of terms
- max term frequency for term
- collection size (# documents)



Step 0:
Post per-term summaries of local indexes

Step
Retrieve li
for each qu

Step 2:
Retrieve and
ombine local query results from peers

Quality based Query Routing

- CORI [Callan98], GlOSS [Gravano99], Decision-Theoretic Approach [Fuhr99],…
  - based on **quality measures**
    - document frequency
    - maximum term frequency
    - number of documents

What we want………

- Overlap Awareness: Combine quality and novelty when selecting most promising peers

  → **usefulness := α*quality + (1- α)*novelty**

  Goal: achieve high recall with fewer peers queried than in

  the traditional approach

- Select ***all*** peers to query a-priori
  – based on statistics (not their actual query results!)

  → allows parallel query execution, no additional latency

What's missing: Way to predict mutual overlap

- Add statistics that allow novelty estimation

Summaries contain statistics like
- document frequency for term
- number of terms
- max term frequency for term
- collection size (# documents)

+

description of Peer X's index list for term t

- Two possible approaches:
  - represent whole collection
  - use single representations for (term-specific) index lists

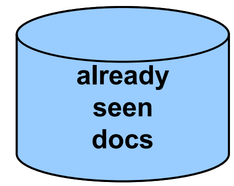
➔ Term-specific representations allow query-specific overlap estimation

- For multi-keyword queries:
  - combine per-term descriptors of a peer to form per-query descriptor

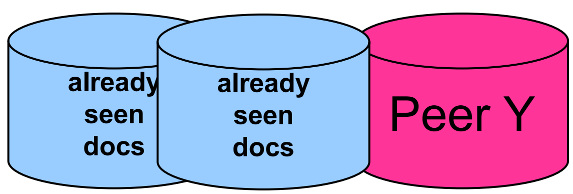
The Iterative Routing Algorithm

- Choose "first" **Peer X** based on quality only

use X's **per-query** descriptor as initial representation of "already seen" documents



- Then choose **Peer Y** with the highest **usefulness** w.r.t. the "already seen" docs
- Merge representations for the peers selected so far and iterate



Bloom Filters [Bloom70]

- bit array of size *m*
- *k* hash functions h_i: *docId_space* → *{1,..,m}*
- insert *n* docs by hashing the ids and settings the corresponding bits
- document is in the Bloom Filter if the corresponding bits are set
- probability of false positives (*pfp*) $pfp = (1 - e^{-kn/m})^k$
  - tradeoff accuracy vs. efficiency

- in the following we use only one hash function
- **important property**: given two BF for collections A and B one can easily form the BF for the **union** and/or the **intersection**

Working with Bloom Filters

- Combining per-term summaries to per-query summaries (intersection)

	0	0	1	0	1	1	0	1	BF for term a
&	1	0	1	0	0	0	0	1	BF for term b
=	0	0	1	0	0	0	0	1	BF for "a *and* b"

- Estimating Peer *i*'s novelty w.r.t. to known documents

$$|\{k|\underbrace{bf_{comb}[k] = 0}_{\text{already known documents}} \wedge \underbrace{bf_i[k] = 1}_{\text{Peer } i\text{'s documents}}\}|$$

already known documents *Peer i's documents*

Testbed

- MINERVA:
  - 100% Java
  - layered on top of a DHT (Chord)
- CHORD:
  - home-brewed java-based re-implementation
- Oracle 10g Database

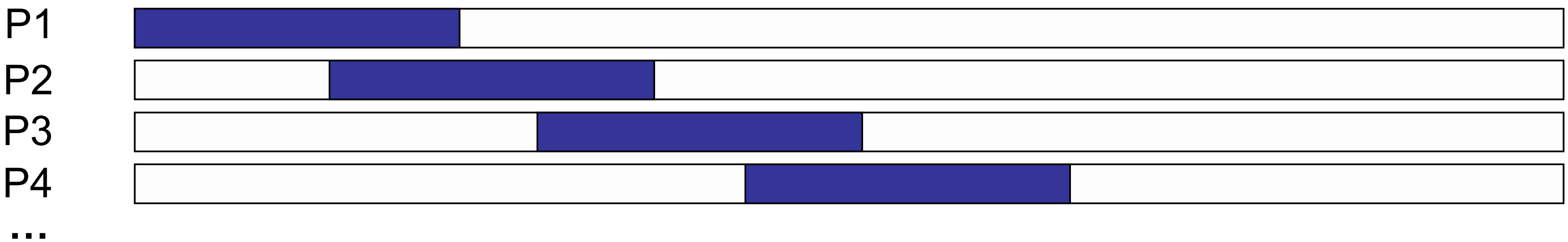
Experimental Evaluation

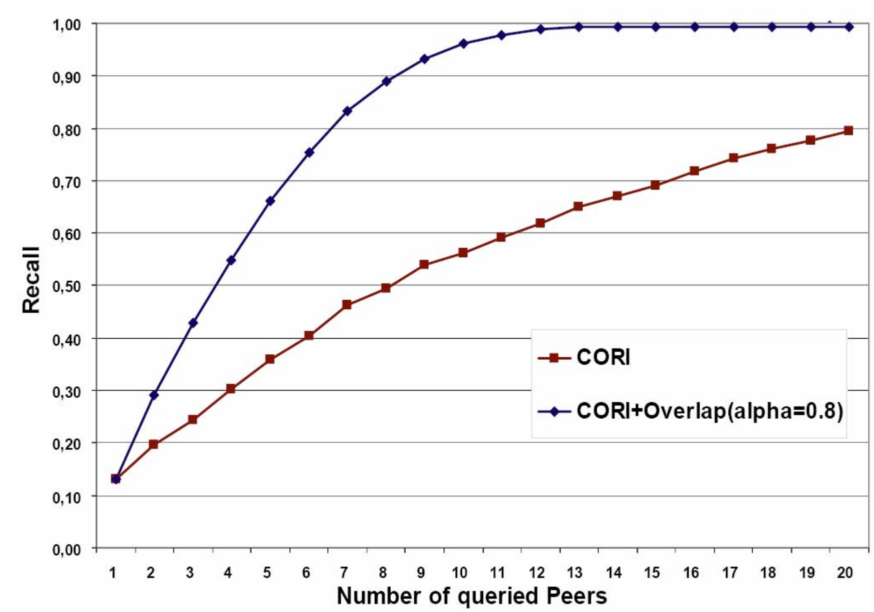
- Algorithms:
  - CORI [Callan98]
  - CORI + overlap prediction

- Datasets
  - Subset of the official TREC .GOV collection split into disjoint fragments. Building peers using…
    - sliding window over these fragments
    - mirrored collections
    - ...

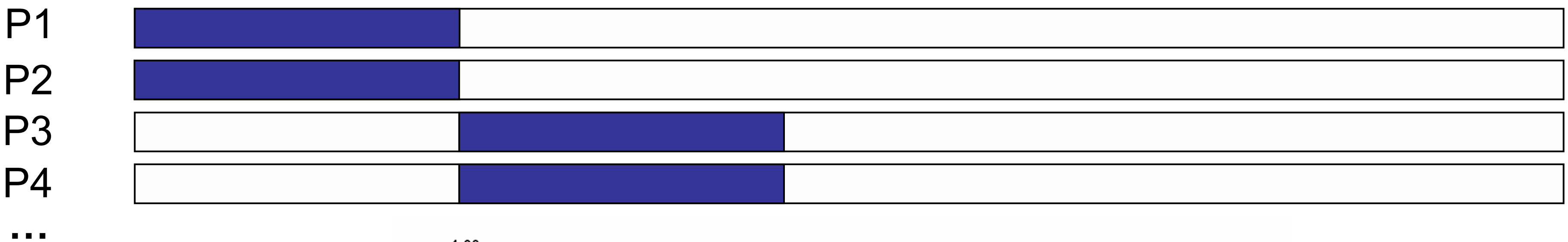
- Queries: 50 TREC-2003 Web queries, e.g. juvenile delinquency
- Measure the recall w.r.t. the query results of the whole document set (relative recall)

Sliding Window Benchmark

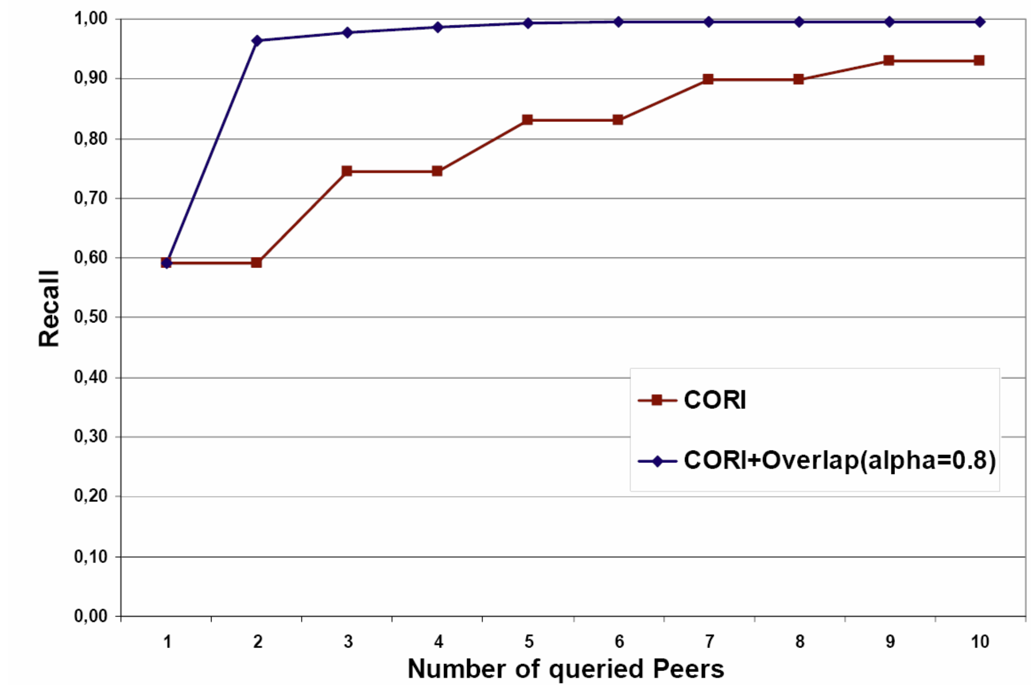


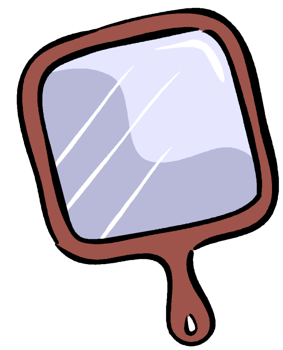


Mirrored Collections Benchmark



20 distinct peers





Conclusion

- Shown the benefits of an overlap aware technique
- Combination of quality and novelty measures
- Plugged into the P2P Web Search Engine MINERVA

- Limitations:
  - fixed BF size. Too small or too big BFs.
  - big BFs (e.g. suitable for the biggest collection) waste network resources
  - not clear which documents should be put into the BF

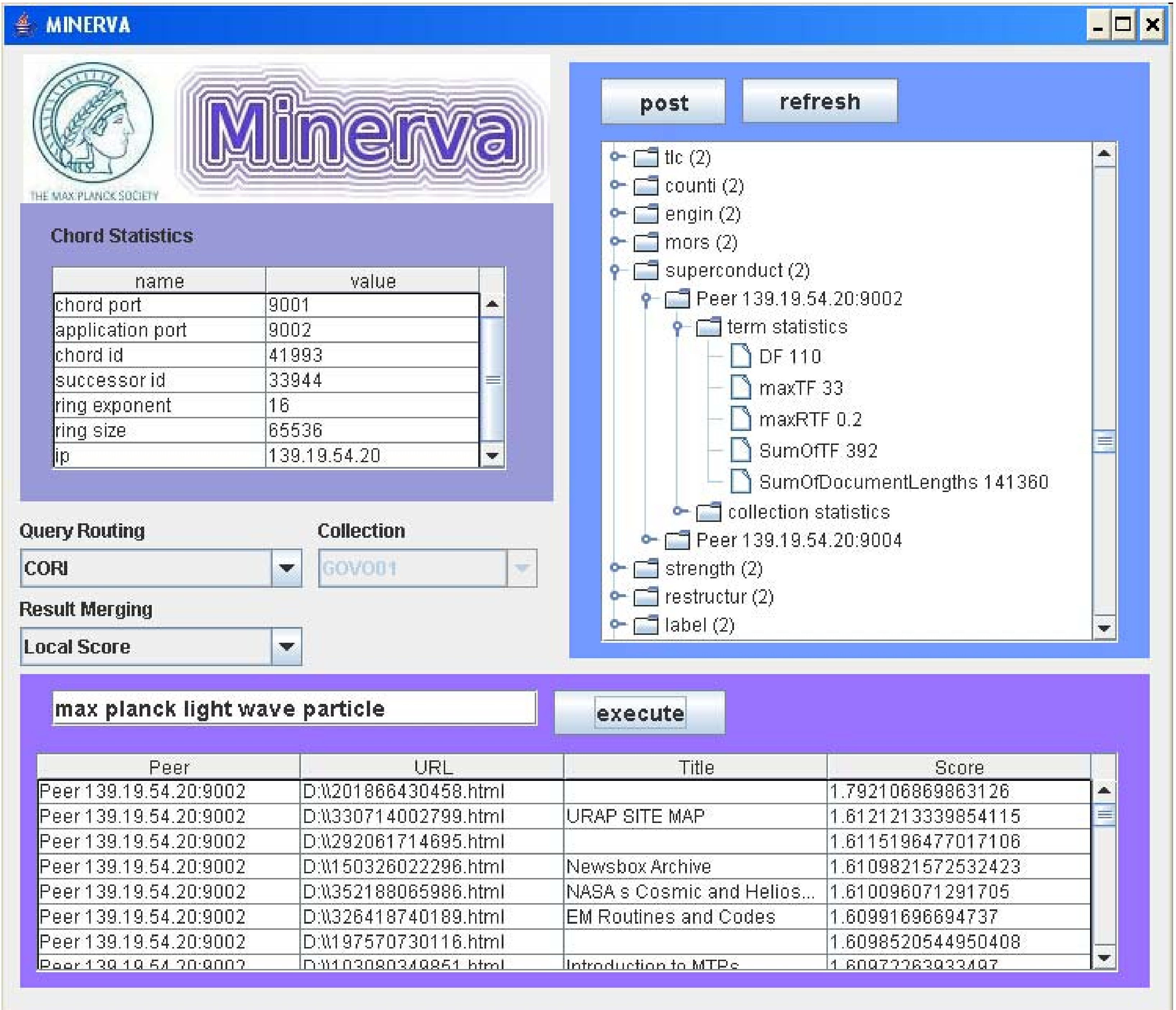
Future/Ongoing Work

- There are other techniques for describing sets….
- Benefit/cost ratio
- put only subset of docs in BF to (i) enhance quality of representation and (ii) avoid the size-related limitations….
  - bookmarks?
  - top-*k* per index list?
  - random sample?
  - leave it to the peers?
- Experiments with larger datasets & larger number of peers

Thanks for you attention!

Questions?

Comments?


MINERVA
THE MAX PLANCK SOCIETY
Minerva
Chord Statistics
name | value
chord port | 9001
application port | 9002
chord id | 41993
successor id | 33944
ring exponent | 16
ring size | 65536
ip | 139.19.54.20
Query Routing
CORI
Collection
GOV001
Result Merging
Local Score
post
refresh
tlc (2)
counti (2)
engin (2)
mors (2)
superconduct (2)
Peer 139.19.54.20:9002
term statistics
DF 110
maxTF 33
maxRTF 0.2
SumOfTF 392
SumOfDocumentLengths 141360
collection statistics
Peer 139.19.54.20:9004
strength (2)
restructur (2)
label (2)
max planck light wave particle
execute
Peer | URL | Title | Score
Peer 139.19.54.20:9002 | D:\\201866430458.html | | 1.792106869863126
Peer 139.19.54.20:9002 | D:\\330714002799.html | URAP SITE MAP | 1.6121213339854115
Peer 139.19.54.20:9002 | D:\\292061714695.html | | 1.6115196477017106
Peer 139.19.54.20:9002 | D:\\150326022296.html | Newsbox Archive | 1.6109821572532423
Peer 139.19.54.20:9002 | D:\\352188065986.html | NASA s Cosmic and Helios... | 1.610096071291705
Peer 139.19.54.20:9002 | D:\\326418740189.html | EM Routines and Codes | 1.60991696694737
Peer 139.19.54.20:9002 | D:\\197570730116.html | | 1.6098520544950408