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

![Diagram showing overlapping collections]
Why Overlap Awareness?

**naïve routing strategy:**

<table>
<thead>
<tr>
<th>Recall</th>
<th>#peers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

\{A\} \quad \{A,B\} \quad \{A,B,C\} \quad \{A,\ldots,D\}

**overlap aware routing strategy:**

<table>
<thead>
<tr>
<th>Recall</th>
<th>#peers</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

\{A\} \quad \{A,E\}
Structured P2P Systems

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

\[ \text{lookup}(\text{key}) \rightarrow \text{in } O(\log n) \text{ routing hops!} \]

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

+ robustness to load skew, failures, dynamics

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 $k \leq p$ and there is no node $p'$ with $k \leq p'$ 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_{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} |

| 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} |

Chord Ring

Lookup(54)
Design Fundamentals

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

• Peers share metadata 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)

Step 0:
Post per-term summaries of local indexes

Step 1:
Retrieve list of peers for each query term

Step 2:
Retrieve and combine local query results from peers

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

Distributed Directory
Term → List of Peers

Query Routing

Max-Planck-Institut
Informatik

Sebastian Michel
Improving Collection Selection with Overlap Awareness in P2P Search Engines

University of Patras
NetCInS Lab

ACM SIGIR 2005, Salvador, Brazil
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
  \[ \text{usefulness} := \alpha \times \text{quality} + (1 - \alpha) \times \text{novelty} \]
  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!)
  \[ \rightarrow \text{allows parallel query execution, no additional latency} \]
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

already seen docs
already seen docs
Peer Y
Bloom Filters [Bloom70]

- bit array of size $m$
- $k$ hash functions $h_i: docId_{space} \rightarrow \{1,...,m\}$
- 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)

\[
\begin{array}{cccccc}
0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
\end{array}
\]
BF for term a

\[
\begin{array}{cccccc}
& & & & & & & \\
\end{array}
\]
BF for term b

\[
\begin{array}{cccccc}
0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\
\end{array}
\]
BF for “a and b”

• Estimating Peer i’s novelty w.r.t. to known documents

\[
\left| \{ k \mid b f_{comb}[k] = 0 \land b f_i[k] = 1 \} \right|
\]

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

![Graph showing recall vs. number of queried peers]

- **P1**
- **P2**
- **P3**
- **P4**
- ...

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?
Sebastian Michel
Improving Collection Selection with Overlap Awareness in P2P Search Engines

Max-Planck-Institut Informatik
University of Patras
NetCInS Lab