

Query Workload Driven Summarization for P2P Query Routing

Linh T. Nguyen, Wai Gen Yee and Ophir Frieder
Illinois Institute of Technology

2008 IEEE Conf. on Peer-to-Peer Computing,
Aachen, Germany

Goal

- Improve query routing accuracy in peer-to-peer networks
 - *Content*-based routing, not *key*-based routing

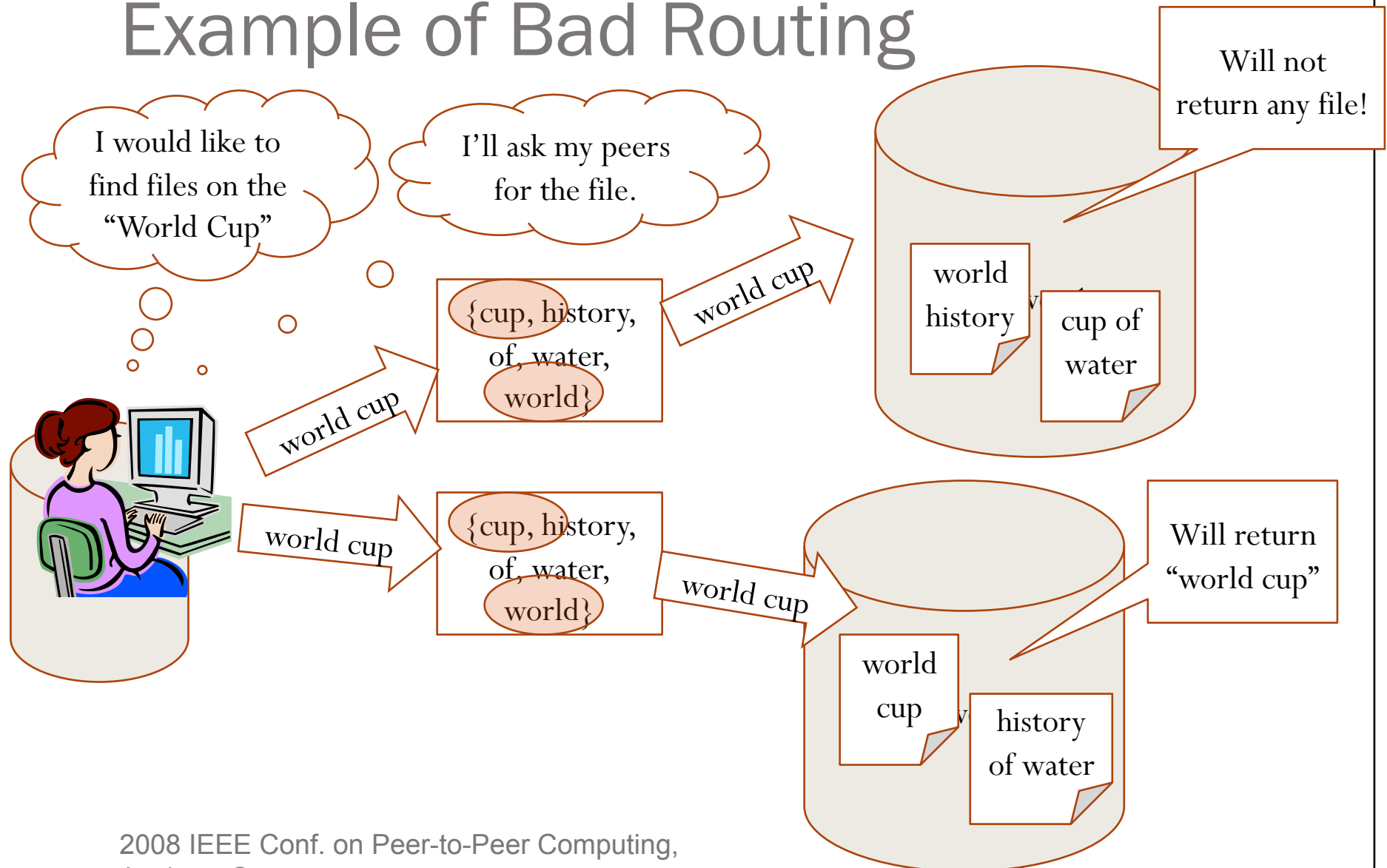
Problem

- Peer contents are too coarsely described in current P2P networks
 - Queries are sent to these peers even though they share no matching files

General Approach

- Improve the precision of the description of a peer's shared content
- Leads to fewer queries routed to peers with no matching files

Example of Bad Routing



“Co-occurrence Error” and “Routing Error”

- Summary of peer 1 implies that “world” and “cup” *co-occur* in some file
 - Summary: {cup, history, of, water, world}
 - File collection: {world history}, {cup of water}
 - → they do not
- Co-occurrence error led to a “routing error”

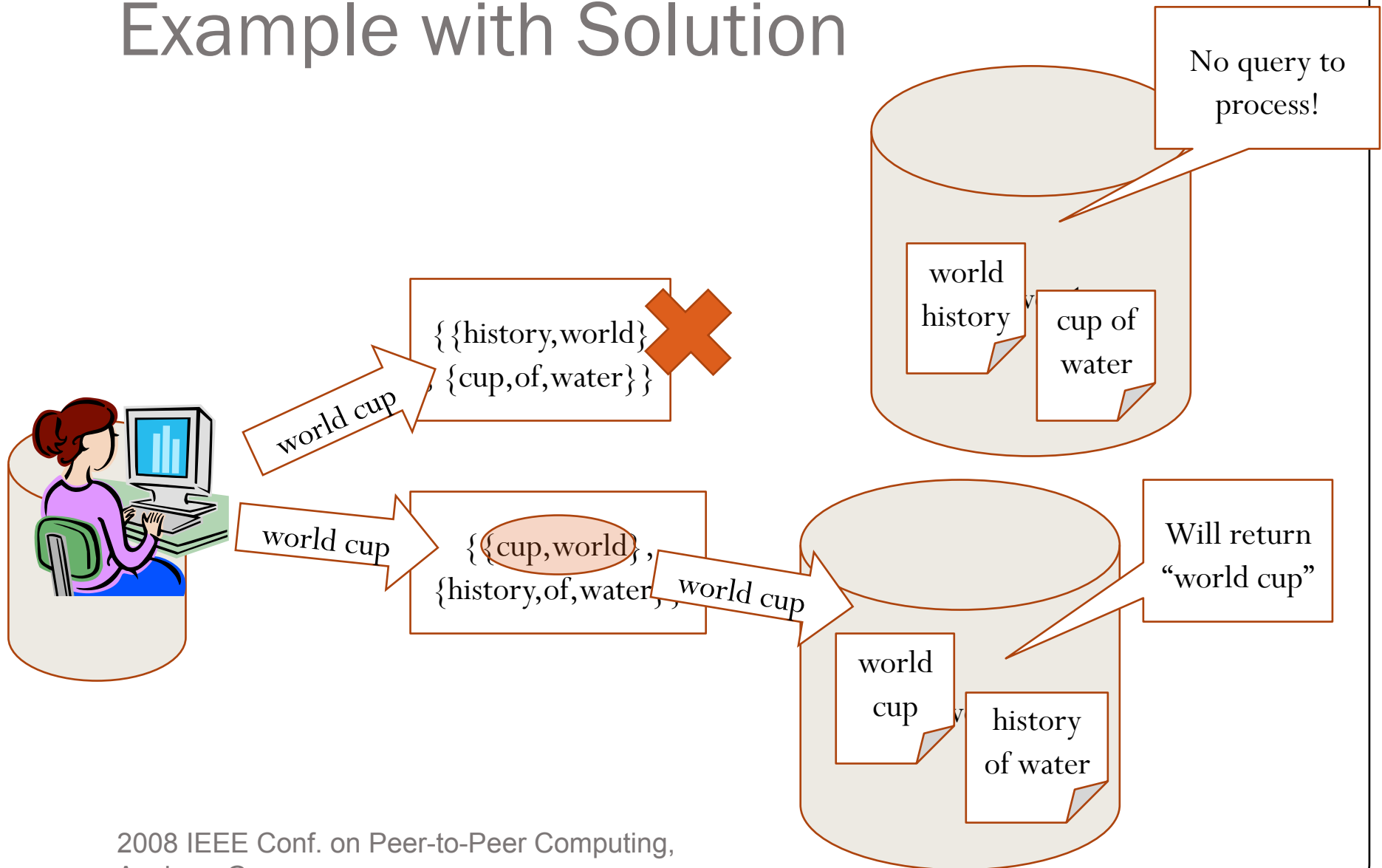
Impact of Problem

- In our analyses of Gnutella data, 70% of queries routed to a peer return no results

Solution

- For each collection:
 - Create groups of files
 - Create a description for each group
 - Query checks each group for match

Example with Solution



Challenges

- Given that every peer has a unique collection
 - How to create the groups
 - How many groups to create
 - How to use user behavior to tune groups

Related Work

- Most P2P routing techniques assume either:
 - Peer summaries are bags of words
 - Or use full indexing (e.g., index at the document level)

Creating Groups

- Straightforward solution: text clustering techniques

Why Grouping Works

- Each peer summary, T , matches $2^{|T|}$ queries
- Assume you create a summary with the disjoint groups, T_1, T_2, \dots, T_K
- These groups match $2^{|T_1|} + 2^{|T_2|} + \dots + 2^{|T_K|} < 2^{|T|}$ queries, respectively
- The number of matching queries is non-increasing in K

Good Group Characteristics

- Each group should be small
- The variance in group sizes should be small

- Example worst case: all terms in one group
- Example best case: all groups same size, no overlap
 - (Maybe)

Base Case Grouping Techniques

- K-means-based clustering
 - Some distance metric for files (e.g., cosine similarity)
- Random grouping
 - Just randomly assign files to K different groups

Proposed Grouping Technique: ΔM

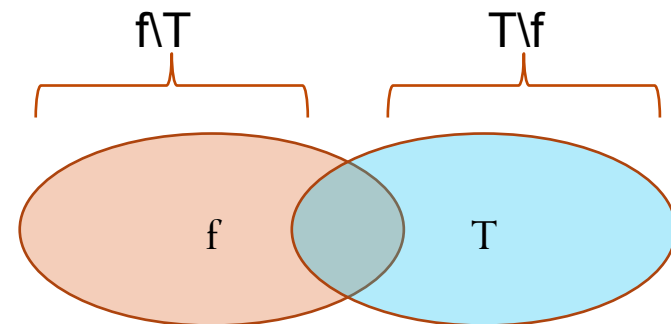
- Initialize K groups, T_1, T_2, \dots , with a random file $f \in F$
- For each file $f \in F$:
 - Assign f to D_i where $i = \operatorname{argmin}_i \Delta M(f, T_i)$

Idea Behind ΔM

- When adding a file to a group, the increase in query-matching ability is a function of the increase in group size
 - $\{\text{world, cup}\}$ matches $\{\text{world}\}$, $\{\text{cup}\}$, $\{\text{world cup}\}$, $\{\}$
 - $\{\text{world, cup}\} + \{\text{cup}\} \rightarrow$ no change (4 queries match)
 - $\Delta M(\{\text{world, cup}\}, \{\text{cup}\}) = 0$
 - $\{\text{world, cup}\} + \{\text{history}\} \rightarrow$ increase by 4 queries
 - $\{\text{world history}\}$, $\{\text{cup history}\}$, $\{\text{world cup history}\}$, $\{\text{history}\}$
 - $\Delta M(\{\text{world, cup}\}, \{\text{history}\}) = 4$

Defining ΔM

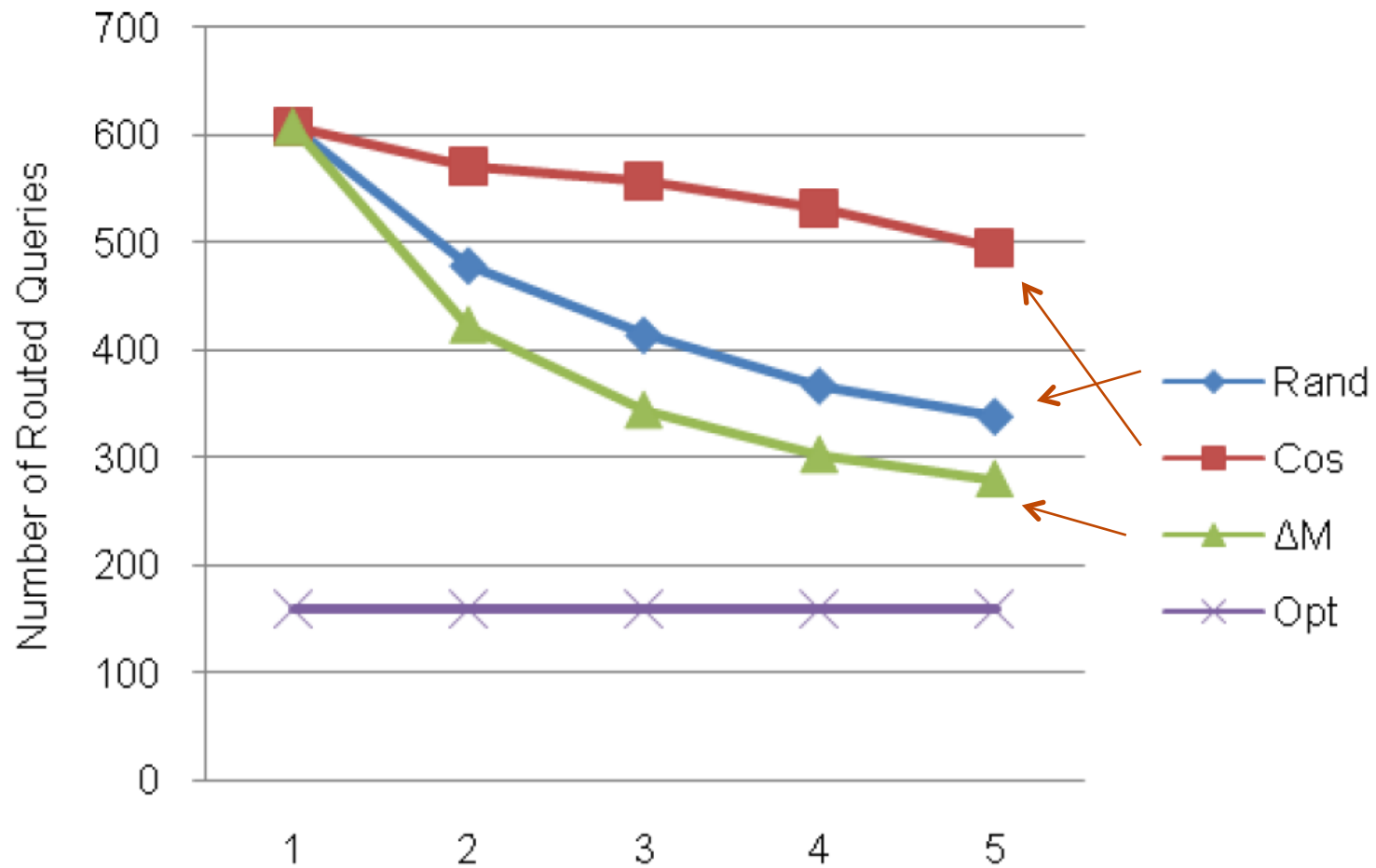
- Assume we have a file f and a summary T
- The increase in the number of queries that match T (by adding f to T) is a function of
 - $f \setminus T$: The terms in f that are not in T
 - $T \setminus f$: The terms in T that are not in f
- $\Delta M = (f - T) * (T - f)$



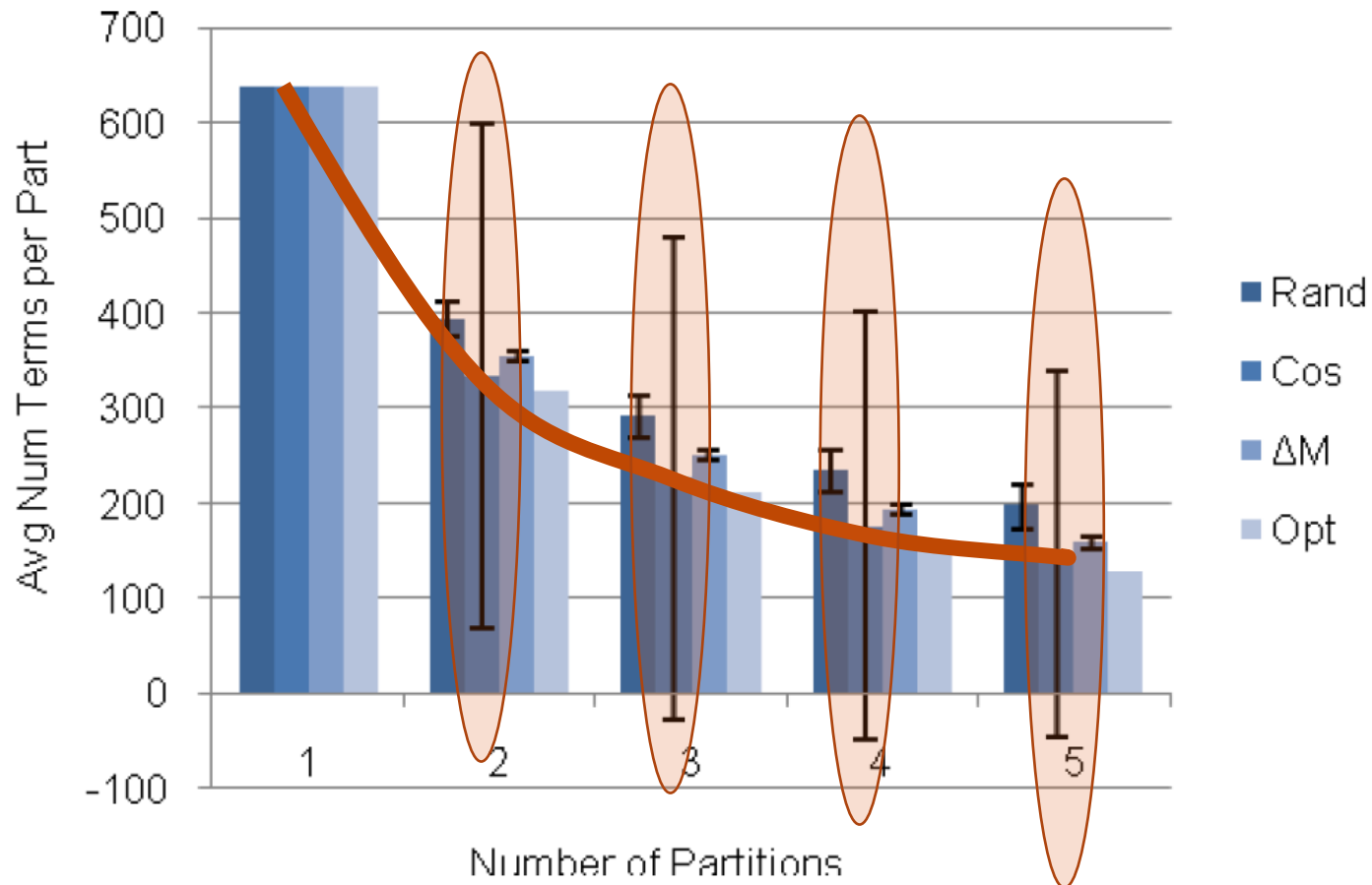
Experimental Setup

- Crawled the Gnutella network in 2007 with our tool (ICDE07)
 - 65,000 queries from the Gnutella network
 - 50 random peers, sharing from 100 and 500 files
- Metric: average number of queries that match each peer's summary

Performance with Increasing K



Size of Each Group



Determining the Number of Groups

- Effectiveness of grouping increases with increasing K
 - K = number of files eliminates co-occurrence errors
- But, cost of transmitting group information to neighbors also increases
- Solution (ACM/ICST Infocast, 2008):
 - Encode groups in a fixed size Bloom filter
 - Model increase in Bloom filter collisions with K
 - Model decrease in “co-occurrence” errors with K
 - Find tradeoff

Problem with Basic ΔM Solution

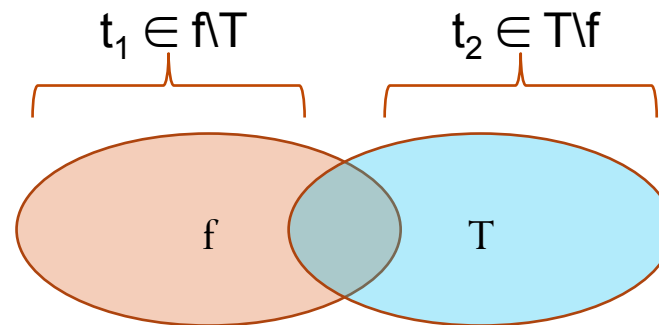
- Basic ΔM tries to eliminate all co-occurrence errors equally
- Problem: Not all co-occurrence errors are equally likely to result in routing errors
- Hypothesis:
- Summaries based on co-occurrence errors likely to result in routing errors are more precise

Query Driven Solution

- Estimate likelihood that a co-occurrence error will result in a routing error by studying queries in query log Q
 - If they occur in Q , then eliminate them
- Example:
 - Assume the files $\{\text{world}\}$, $\{\text{cup}\}$, $\{\text{water}\}$
 - ... With the summary: $\{\text{world, cup, water}\}$
 - Better to split “world” and “cup” rather than “world” and “water.”
 - “world water” likely does not exist in Q , so who cares?

Query Driven Cost Function

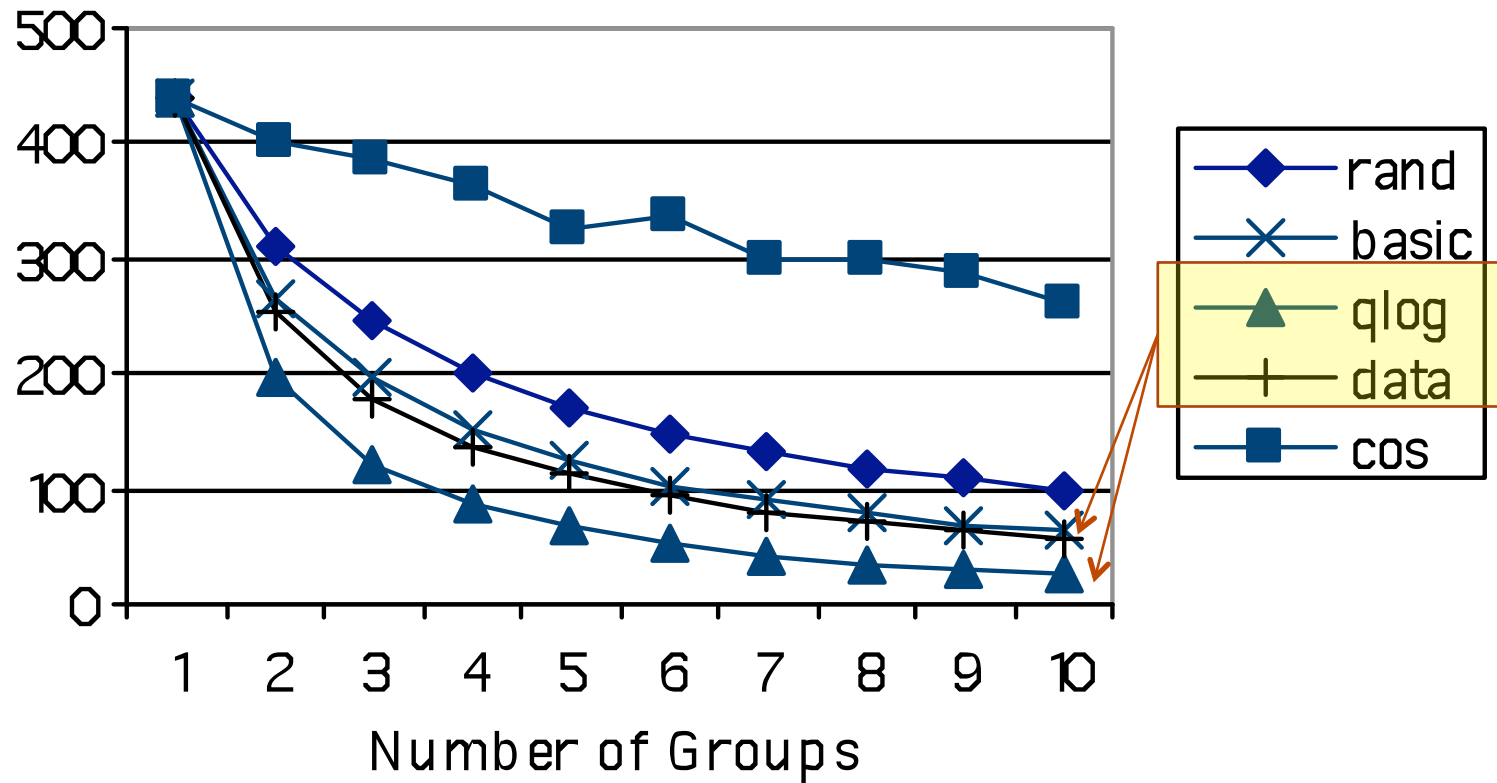
- Assume we have a file f and a group T
- Query driven cost of assigning f to T is
 - 0 if f contains T or T contains f
 - Else, frequency of all q in query log Q , where $q = (t_1, t_2)$ is a co-occurrence error created by adding f to T
 - Original model did not consider the frequency of q in Q



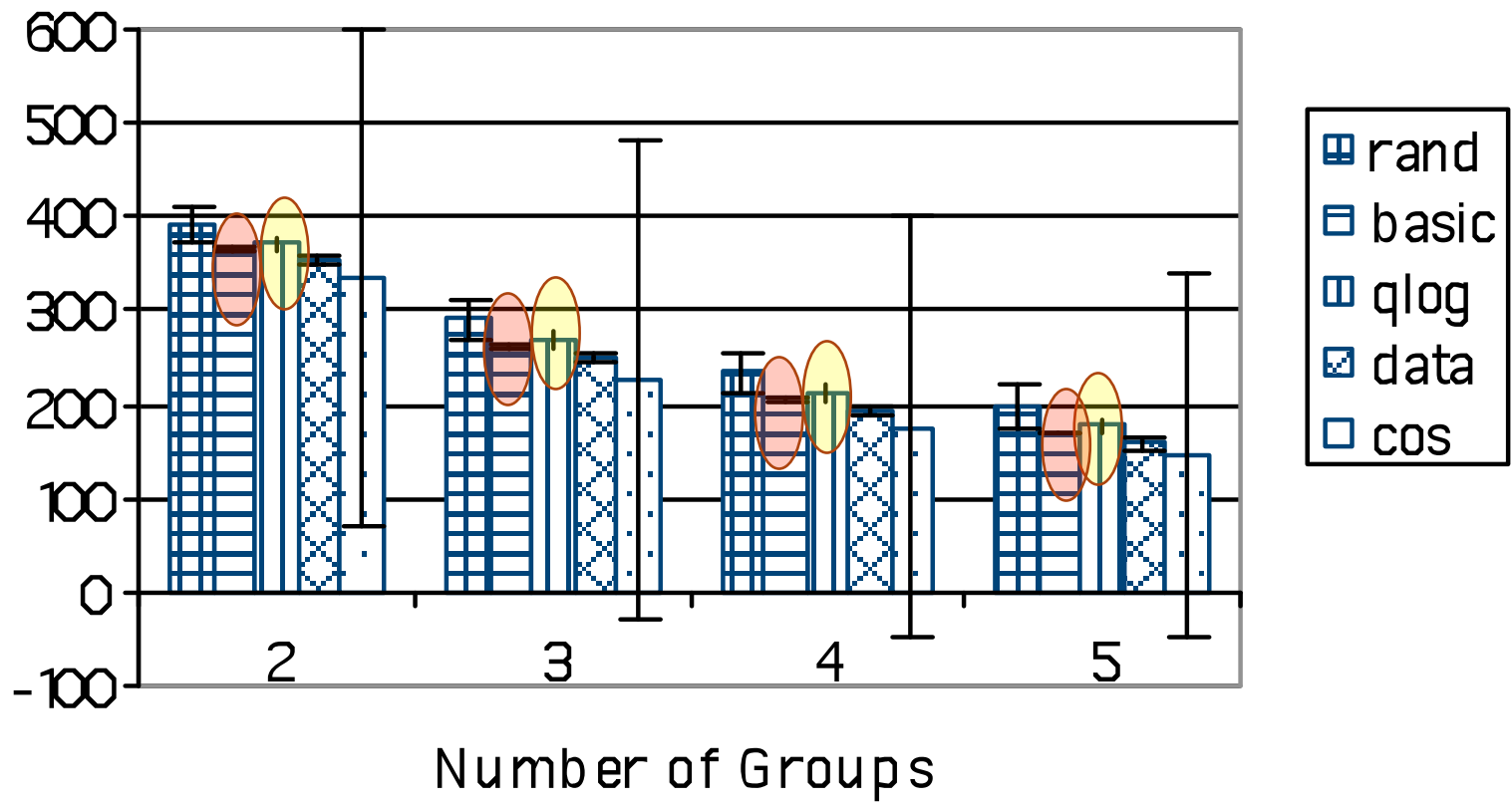
Local Data Driven Solution

- If a newly created co-occurrence error exists in some other file, do not consider its cost
- Example:
 - {world history} + {cup of water} creates {history of water} (as well as other co-occurrence errors)
 - Check if a local file contains {history of water}
 - If yes, do not count the cost of {history of water}
 - Else, add 1 to the cost of this combination

Performance – Number of Routing Errors with K



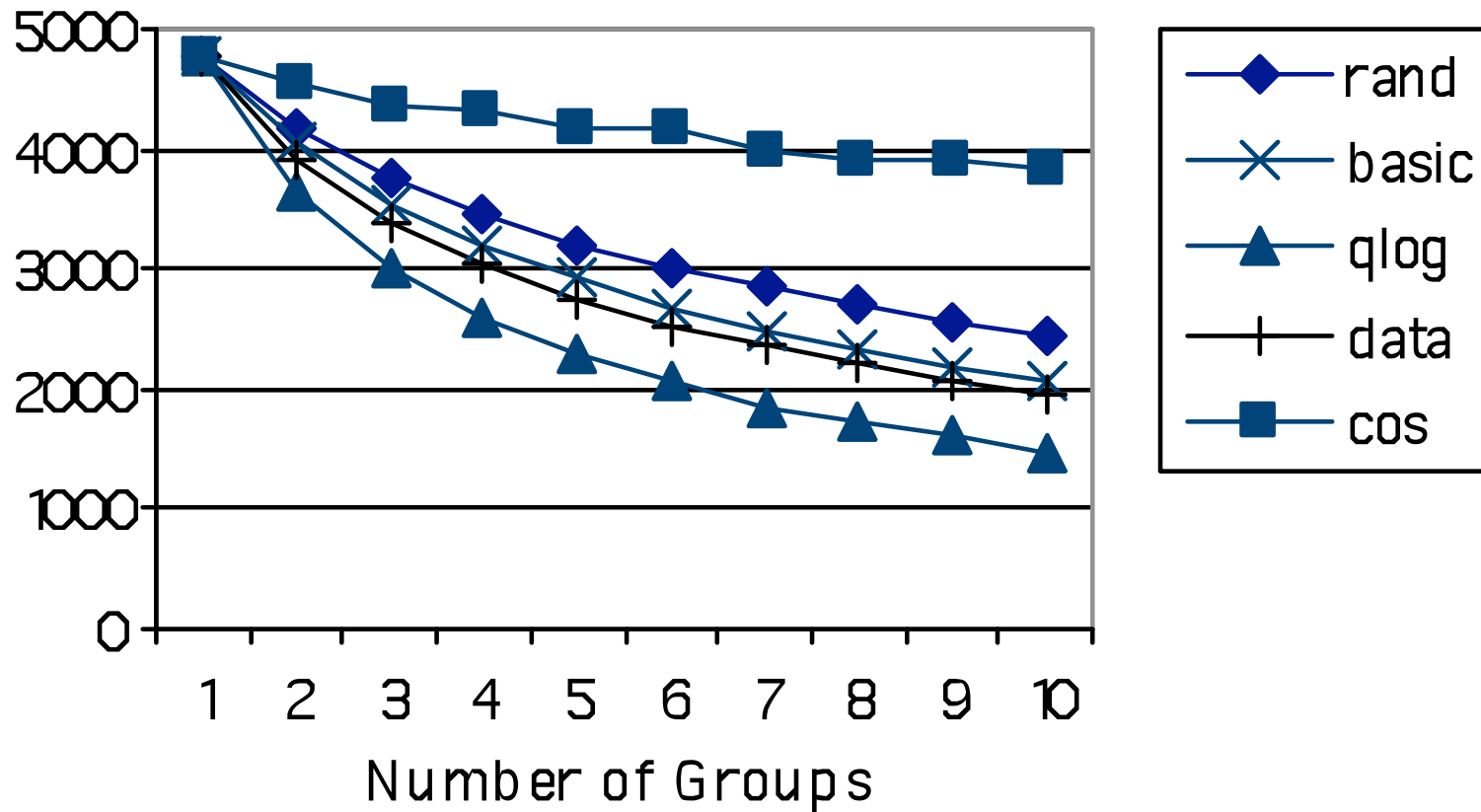
Explanation of Effect of Query Driven Summarization



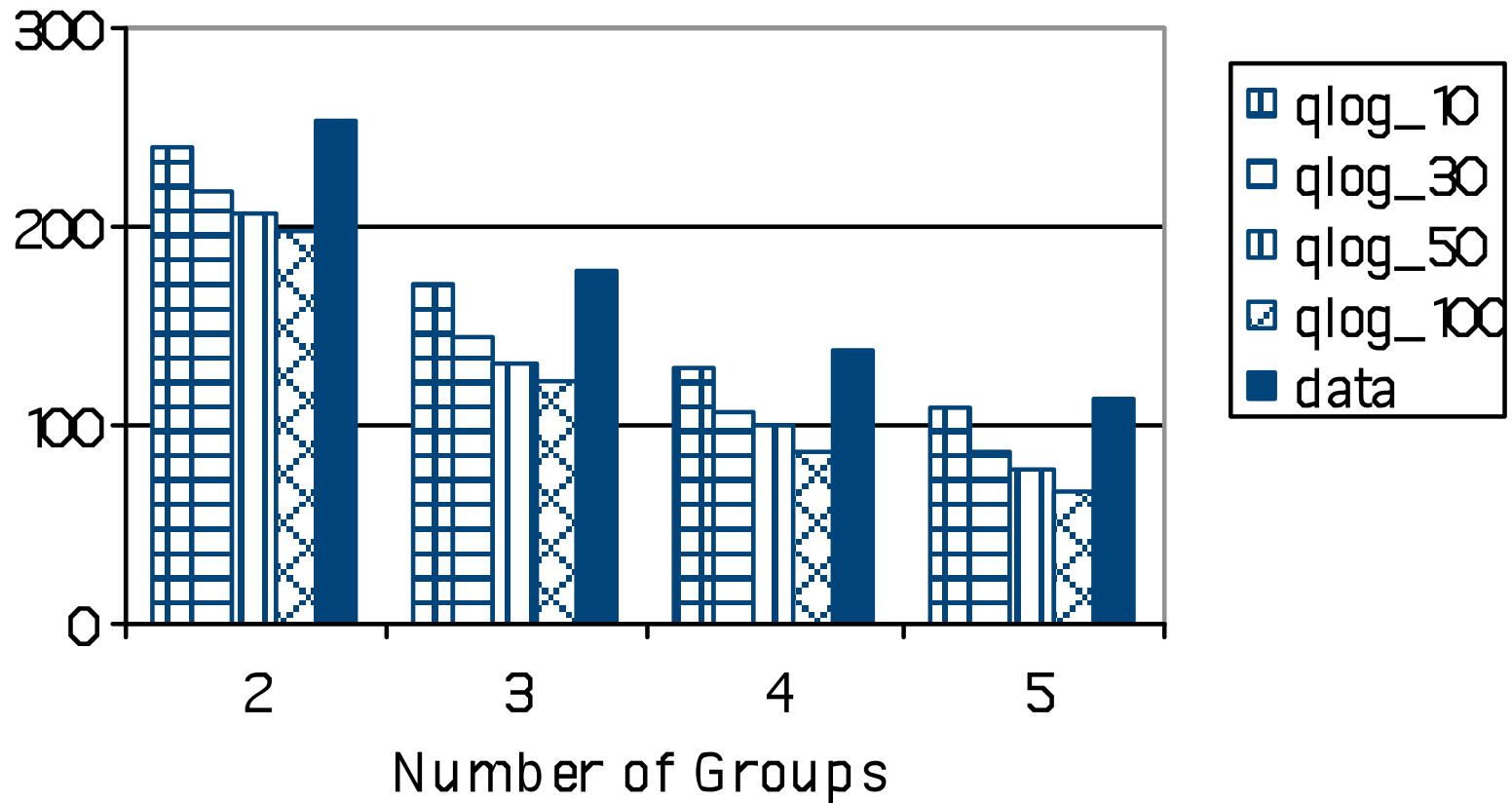
Experiments with Web Data

- Random 50 domains from the WT10G TREC collection
 - Each contains from 100 to 500 documents
 - Random 65,000 queries selected

Performance with Web Data – Number of Routing Errors with K



How Much Query Log Do We Need?



Conclusion

- The current approach to query routing – using bags of words to describe a collection – is too coarse
- Grouping allows us to make more precise the description
 - Decrease routing errors by 60%
- Grouping technique
 - Need “difference-based” grouping is effective
 - But use more data if available –
 - Can improve accuracy over basic difference-based grouping by up to 65%
 - ... And up to 75% over base case

Other Work

- Applied these concepts to content search in DHT networks
 - Reduction of search cost by over 40%
 - CIKM08

Future Work

- Model usefulness of query log driven solution with time
- Apply grouping to multi-hop routing techniques

Danke!

- Questions?

- Contact info:
 - Wai Gen Yee
 - yee@iit.edu
 - Ir.iit.edu