Intro to Peer-to-Peer Search

(COSC 416)

Nazli Goharian
nazli@cs.georgetown.edu

1

Outline

- Peer-to-peer historical perspective
- Problem definition
- Local client data processing
  - Ranking functions
  - Metadata copying
- Improving Retrieval accuracy
  - Probing queries
    - Relevance Feedback
    - Automatic descriptor enhancements
  - Descriptor enhancement using association rules on query logs

2

P2P Historical Perspective

File sharing systems:

- **Centralized**
  - **Napster --**First commercial application
  - Centralized index → guarantied results but vulnerable to attacks
  - Troubled by intellectual property infringements
- **Truly Distributed**
  - Gnutella protocol V 0.4
  - Query broadcasting by flooding, with a limited horizon
- **Hierarchical**
  - Gnutella protocol V 0.6
  - Superpeers as hubs for resource selection and results merging
  - Using a time-to-live (TTL) counter
- **Distributed Hash Table (DHT)**
  - Each node maintains the hashed key of some data
  - Examples: CAN, Chord, Pastry, Tapestry, BitTorrent

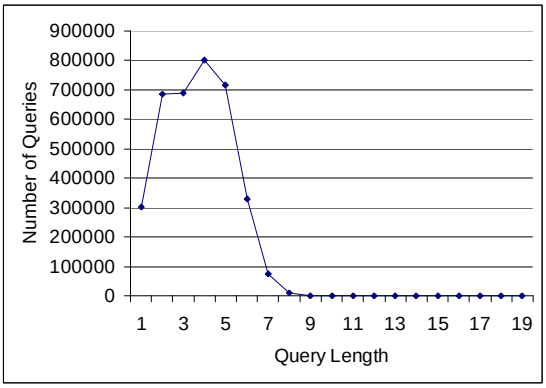
3

Peer to Peer "Key Features"

- Peer autonomy
- Self organization
- High scalability
- High robustness
- No global information

4

Query Length Distribution



P2P

- Average query length: 3.63
- 80% of queries contain 2 to 5 terms
- Only 8% of queries contain a single term

WWW

- Average query length: 2.34/2.86 (AOL)
- 76% of queries contain 1 to 3 terms
- ~ 28% of queries contain a single term

5

P2P Search Problem Definition

- Files are binary
  - Files are not self describing
  - Descriptors used
    - Often short and non-descriptive
- Conjunctive matching:
  - Match occurs when **all** query terms are in the descriptor D
    - If $Q \subseteq D$ then return (D, H_f, serverid) to client
    - Client uses D to decide if it wants to download the associated file
- No centralized directory of content
- Each node is autonomous
  - Maintains own content directory
  - Client and server of information

6

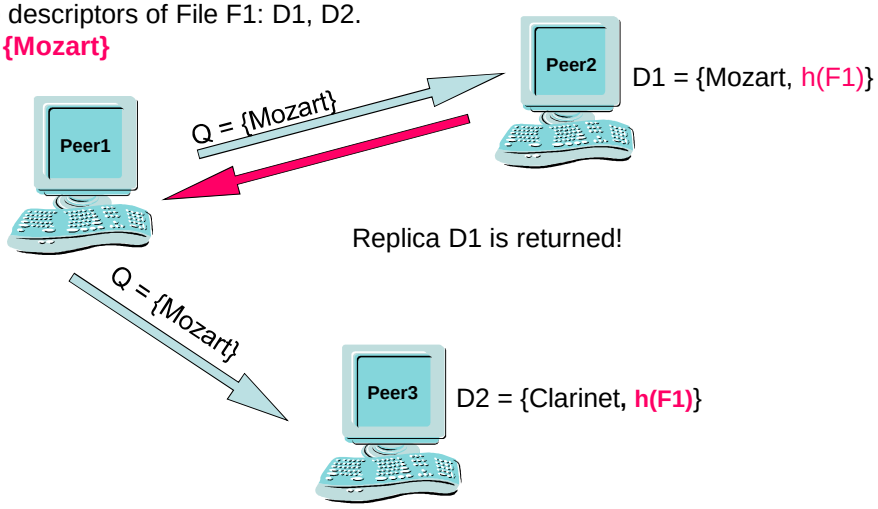
Files & Replicas

- Files (content) & replicas
- File F1 – Mozart Clarinet Concerto
  - { {Mozart}, h(F1) }
  - { {Mozart, Clarinet}, h(F1) }
  - { {Mozart, Concerto}, h(F1) }
  - { {Mozart, Clarinet, Clarinet}, h(F1) }
- File F2 – Beethoven Symphony No. 9
  - { {Beethoven}, h(F2) }
  - { {**Mozart**, Beethoven}, h(F2) }
  - { {Beethoven, Symphony}, h(F2) }
  - { {Beethoven, Symphony, 9}, h(F2) }

7

Conjunctive Queries

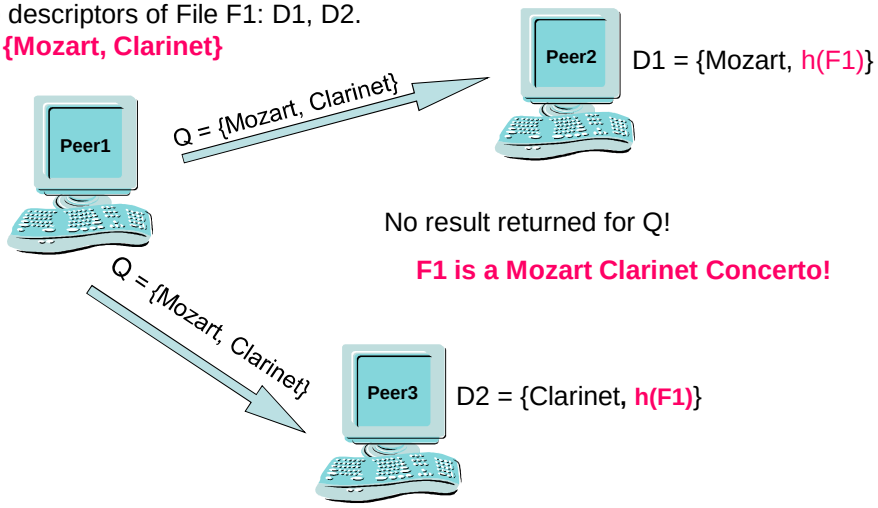
Two descriptors of File F1: D1, D2.
Q = {Mozart}



8

Conjunctive Queries

Two descriptors of File F1: D1, D2.
Q = {Mozart, Clarinet}



9

Effect of Conjunctive Queries (eDonkey)

Mozart	Clarinet	A	Major	Zukovsky	# Results
X	X				80
X	X	X	X		54
X	X			X	2
X	X	X	X	X	0

Both "Mozart Clarinet Zukovsky" recordings were also in A major!

10

Metric

- Mean reciprocal rank

$$MRR = \frac{\sum_{i=1}^{N_q} \frac{1}{rank_i}}{N_q}$$

  - N_q=number of total queries issued
  - $rank_i$=rank of desired result in result set
    - If the desired result is not in the result set, $rank_i=\infty$
  - Measures query *accuracy*
  - Appropriate when searching for a specific file (known item search)

11

Pitfall Summary

Poor file description
+
Conjunctive query constraint
=
Poor retrieval accuracy

Poor accuracy → Repetitive queries → Wasted resources (energy)!

12

Ranking Functions

- **Order of Arrival (Naïve)**
- **Group Size**
  - Number of replicas returned per file
- **Term Frequency**
  - Number of query terms in group descriptors
- **Fraction (Jaccard Coefficient)**
  - Percentage of query terms in group descriptors
- **Cosine Similarity**
  - Cosine of the angle formed by the query and group descriptor vectors

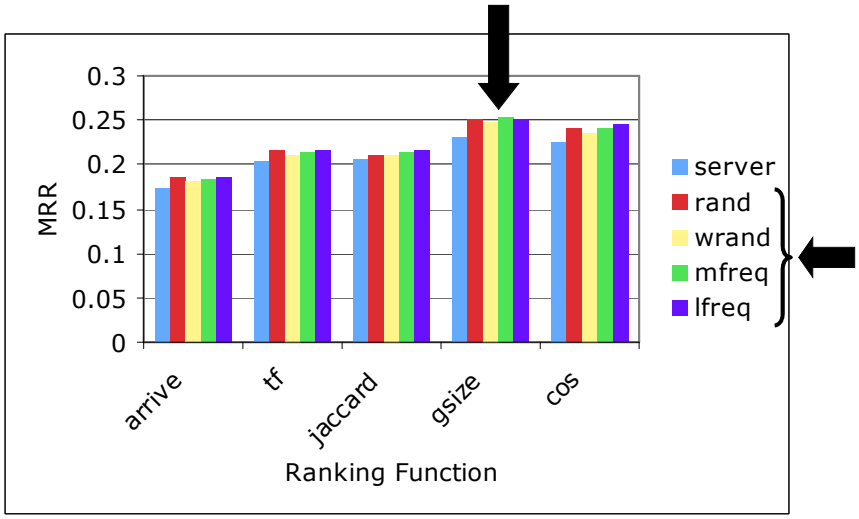
13

Metadata Copying

- **Server**
  - Replicate the descriptor of a particular server
- **Random**
  - Randomly fill new descriptor with terms from group descriptor until descriptor is full
- **Weighted Random**
  - Like random, but biased by relative frequency of term appearance in group descriptor
- **Most Frequent**
  - Most frequent terms in group descriptor selected until descriptor is full or all terms selected
- **Least Frequent**
  - Least frequent terms in group descriptor selected until descriptor is full or all terms selected

14

Accuracy Comparison



15

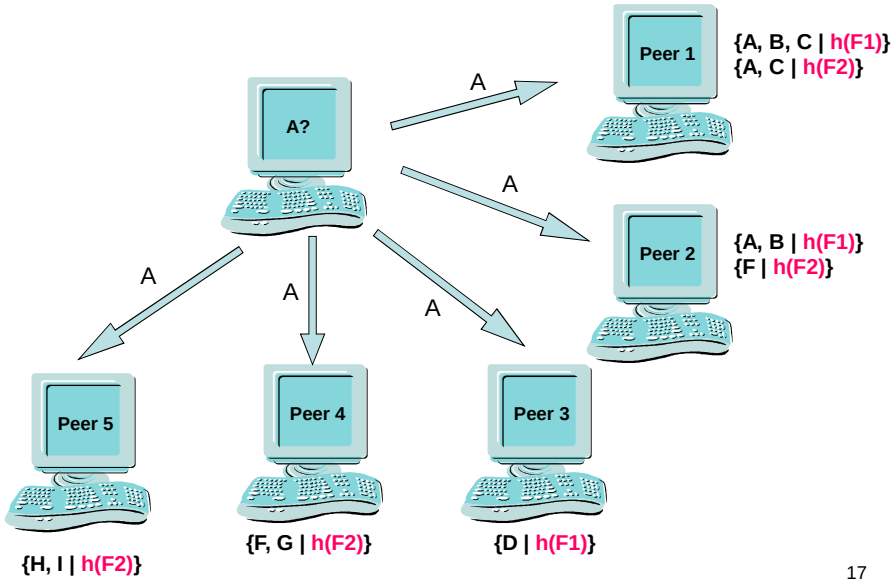
Probe Queries for Relevance Feedback

[W. Yee, L. Nguyen, O. Frieder, NCA 2006]

- Query (traditional)
  - A set of **user-specified terms** used to match against terms in server descriptors
- Probe Query *(proposed by IIT IR Lab)*
  - A query where the set of terms is replaced by a **single hash key** representing the file of interest
- Goal: Improving accuracy

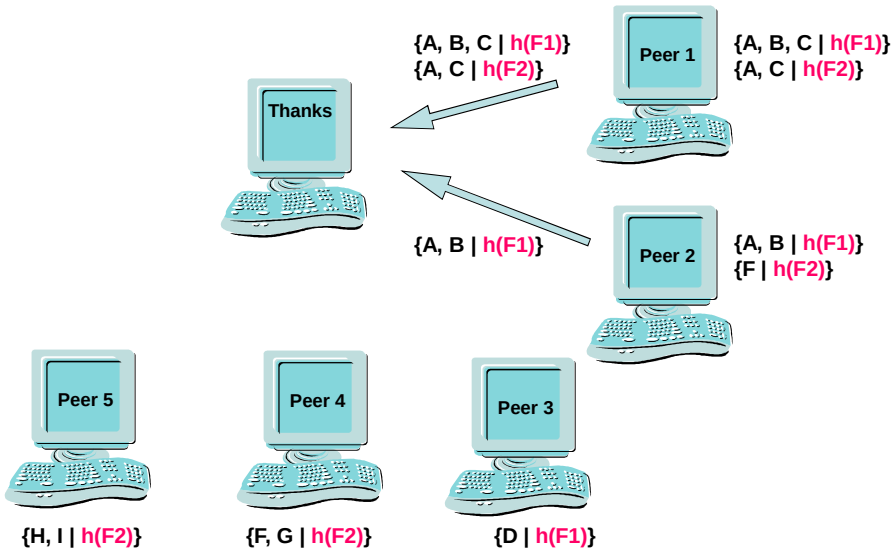
16

Information Search



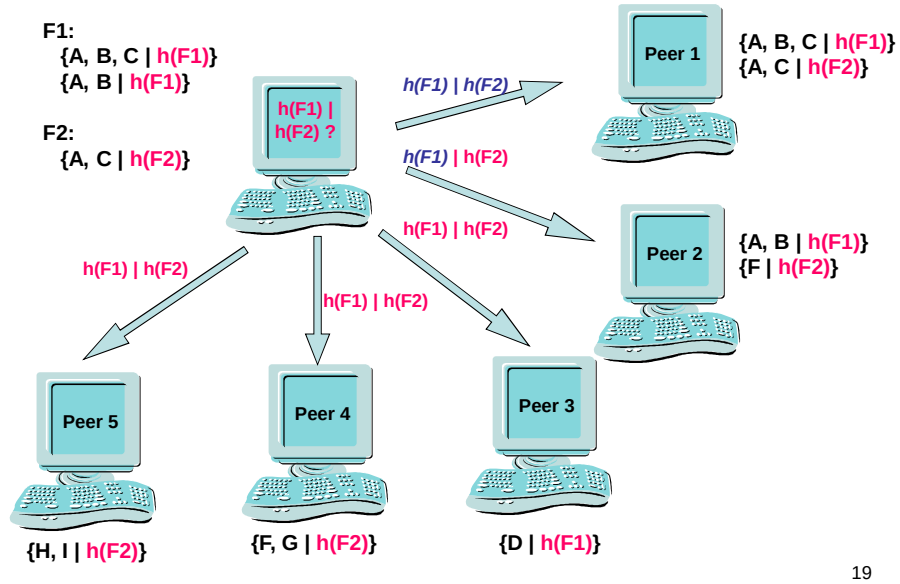
17

Search Response



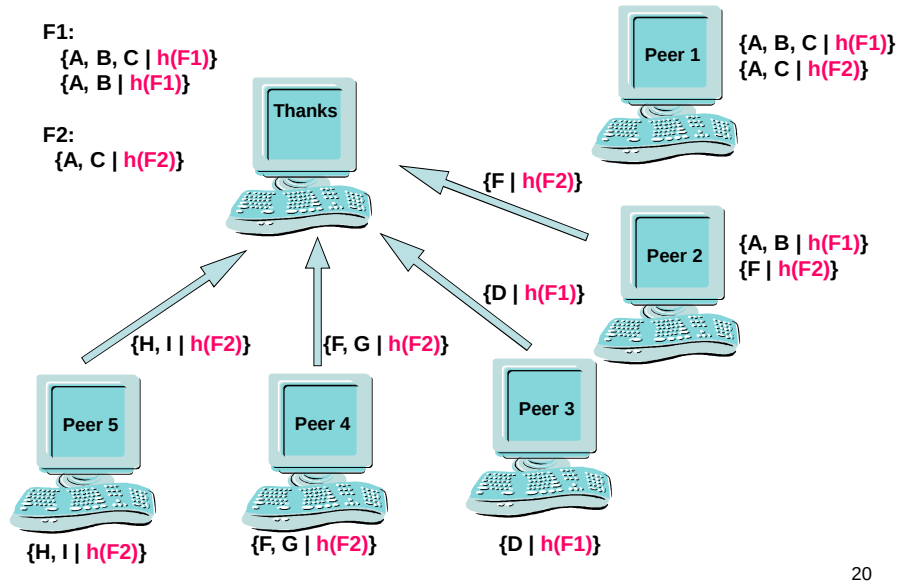
18

Additional Context Probe



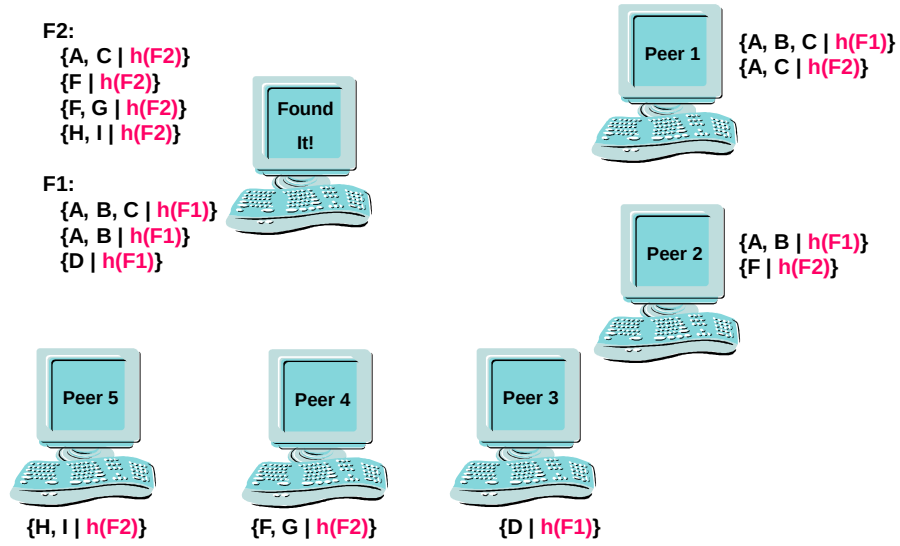
19

Additional Context Response



20

Final Ranking



Example (1/2)

Files

Peer 1: {A, B, C | h(F1)} & {A, C | h(F2)}

Peer 2: {A, B | h(F1)} & {F | h(F2)}

Peer 3: {D | h(F1)}

Peer 4: {F, G | h(F2)}

Peer 5: {H, I | h(F2)}

Group by key K_i – after issuing Query {A}

Group G1: {A, B, C | h(F1)} & {A, B | h(F1)}
Group G2: {A, C | h(F2)}

Ranking (by Group Size)

Group G1
Group G2

Example (2/2)

Issue secondary queries

Query Q'1: { h(F1) }
Query Q'2: { h(F2) }

Group by key

Group G'1: {A, B, C | h(F1)}, {A, B | h(F1)}, & {D | h(F1)}
Group G'2: {A, C | h(F2)}, {F | h(F2)}, {F, G | h(F2)}, & {H, I | h(F2)}

Ranking (by group size – other ranking metric possible)

1. **Group G'2**
2. **Group G'1**

QUESTION: "Is new ranking better than original ranking?"

23

Experimental Setup

- 1,000 peers
- 10,000 queries
  - Clients search for specific files
- Max descriptor size is 20 terms
- Average descriptor size is 6 terms
- ~20 files per peer at initialization

24

Probe Queries using "Relevance Feedback"

- To improve ranking accuracy:
  - Retrieving phase
    - Use keyword-based query
    - Primary ranking function: gsize
  - Refining phase
    - Use probe (hash value based) queries
    - Secondary ranking function: tfreq or frac.
- Additional metadata from refinement phase help improve result ranking by ~15%

25

Probe Queries: Automatic Descriptor Enhancement

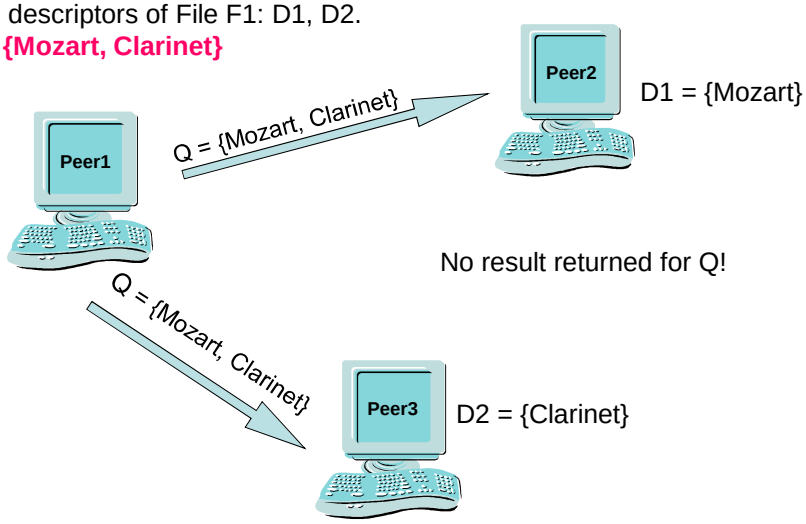
[W. Yee , D. Jia, O. Frieder, P2P 2005]

- Apply the same approach periodically for system enhancement of descriptors
- Shown to improve the MRR for ~ 20%

26

Problem: Descriptor Sparse

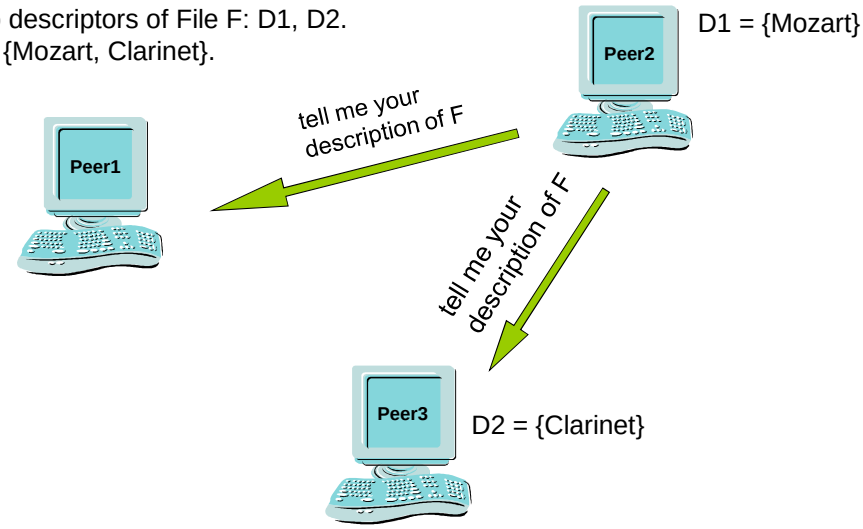
Two descriptors of File F1: D1, D2.
Q = {Mozart, Clarinet}



27

Periodic Probe Query

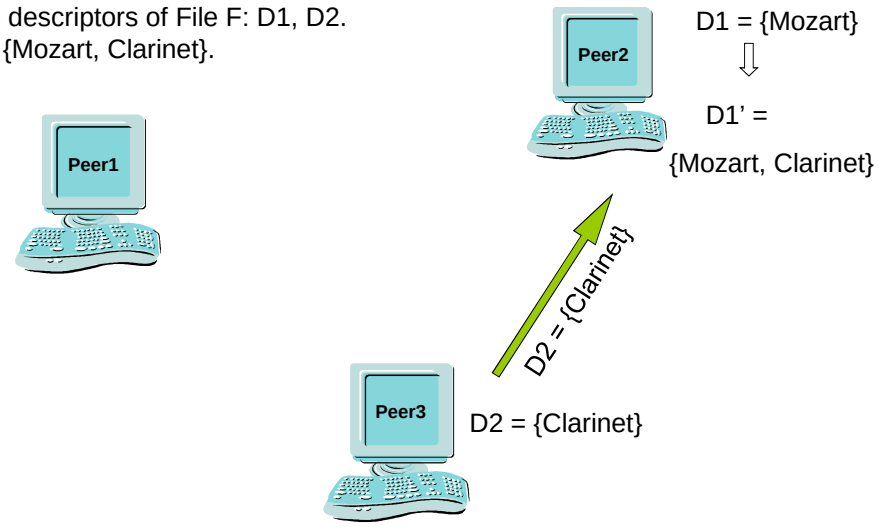
Two descriptors of File F: D1, D2.
Q = {Mozart, Clarinet}.



28

Updating Descriptors

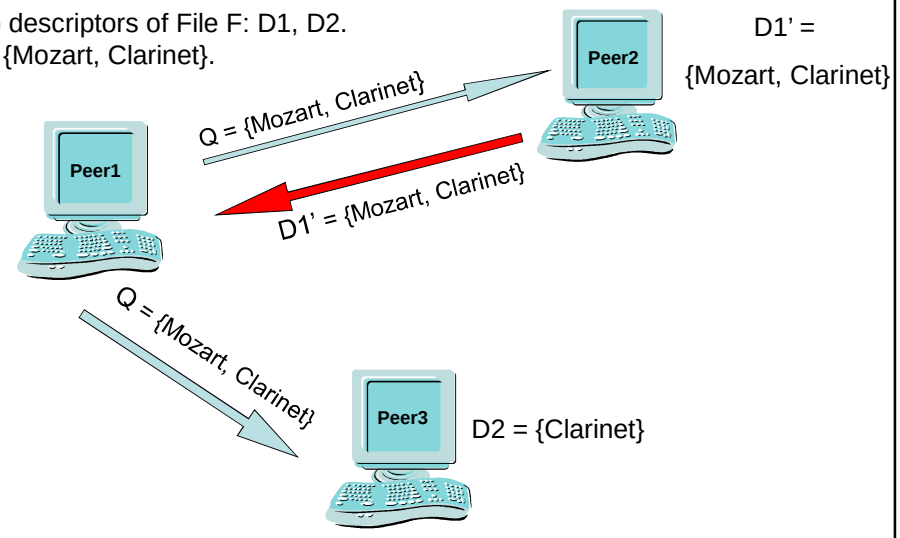
Two descriptors of File F: D1, D2.
Q = {Mozart, Clarinet}.



29

Improved Retrieval

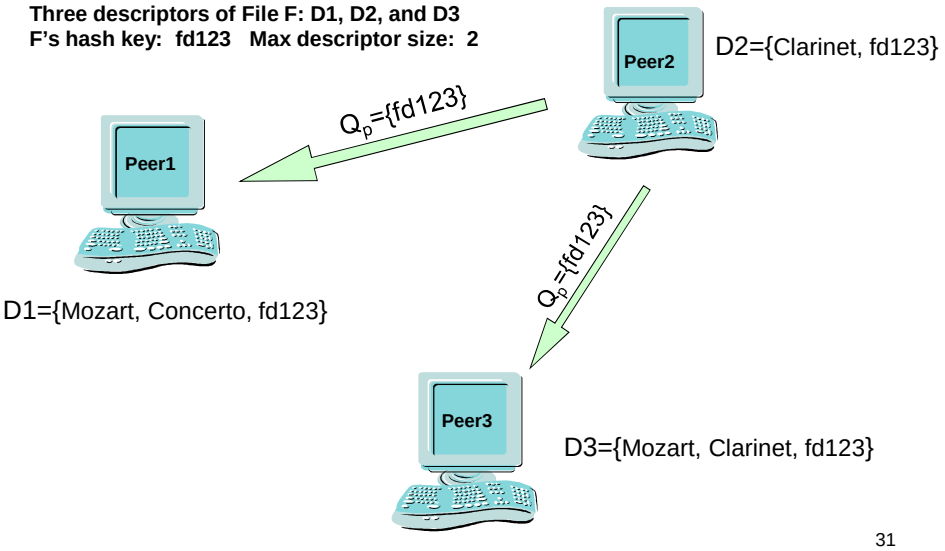
Two descriptors of File F: D1, D2.
Q = {Mozart, Clarinet}.



30

Implementation

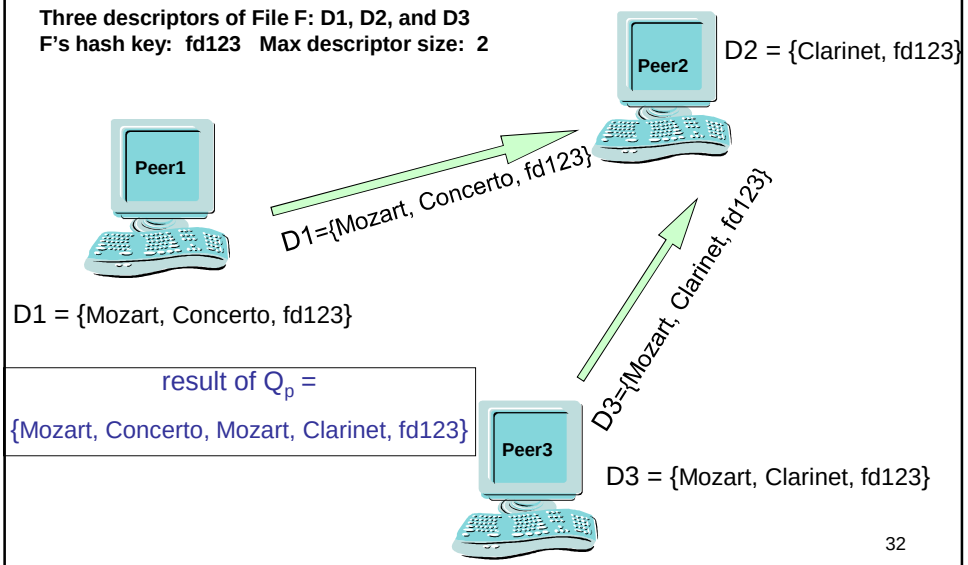
Three descriptors of File F: D1, D2, and D3
F's hash key: fd123 Max descriptor size: 2



31

Implementation (cont)

Three descriptors of File F: D1, D2, and D3
F's hash key: fd123 Max descriptor size: 2



32

Implementation (cont)

Three descriptors of File F: D1, D2, and D3
F's hash key: fd123 Max descriptor size: 2

D1={Mozart, Concerto, fd123}

Most frequent term in probe query results is added to D2.

D2 = {Clarinet, fd123}

D2' = {Clarinet, Mozart, fd123}

result of Q_p =

{**Mozart**, Concerto, **Mozart**, Clarinet, fd123}

D3={Mozart, Clarinet, fd123}

33

Design Challenges

- When to probe?
- What file to probe?
- What to do with probe results?

34

When to Probe?

- When a peer is not busy and under-utilized.
  - Measured by number of responses returned N_r.
- When a peer has a high desire to participate.
  - Measured by number of files published N_f.
- When the system is active.
  - Measured by number of queries received N_q.

35

What File to Probe?

- A poorly/sparsely described file should be probed.
  - *Criterion 1*: a local replica that has matched the fewest queries.
    - A lower number of matches may indicate poor description.
  - *Criterion 2*: a local replica that has a smallest descriptor.
    - Smaller descriptor indicates it is hard to match.

36

What File to Probe? (Cont'd)

- Potential problem
  - A same file is probed repeatedly.
  - Example: file is unpopular, always matches the fewest queries.
- Tentative solution
  - *Criterion 1'*: after every probe of a local replica, double its query match count.

37

What to do with Probe Results?

- Select terms from the result set to add to the local descriptor
  - Most frequent
  - Least frequent
  - Random
  - Weighted random
- Stop when local descriptor size limit is reached

38

Enriching P2P File Descriptors using Association Rules on Query Logs

[N. Goharian, O. Frieder, W. Yee, J. Mudrawala – ECIR 2010]

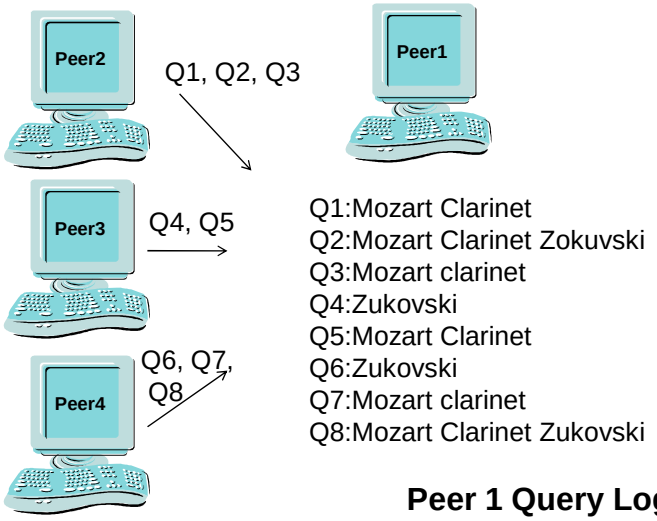
Motivation: Improving accuracy, yet without extra communication among peers!

Each peer

- Maintains its own query log
- Mines its query log to identify the co-related query terms
- Selects a subset of the co-related terms based on an empirically determined *support* and *confidence* to enrich the file descriptors

39

Enriching P2P File Descriptors using Association Rules Mining on Query Logs



40

Enriching P2P File Descriptors using Association Rules on Query Logs

Term 1	Term 2	Support	Confidence
Mozart	Clarinet	0.75	1
Clarinet	Mozart	0.75	1
Mozart	Zukovski	0.25	0.33
Zukovski	Mozart	0.25	0.5
Clarinet	Zukovski	0.25	0.33
Zukovski	Clarinet	0.25	0.5

Derive term associations from peer query log

41

Enriching P2P File Descriptors using Association Rules Mining on Query Logs

Original File Descriptors	Enriched File Descriptors
Mozart	Mozart Clarinet
Zukovski	Zukovski Mozart Clarinet
Clarinet	Clarinet Mozart

Enrich replica descriptors using terms that their correlation support and confidence meets an empirically determined threshold

42

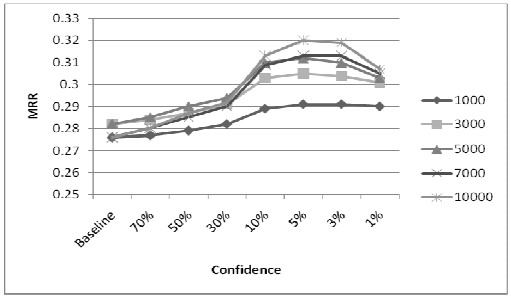
Enriching P2P File Descriptors using Association Rules Mining on Query Logs

Peers	**1,000**
Categories	**37**
Documents	**1,080**
Queries	**10,000**
Descriptor size (terms)	**20**
Initial descriptors size	**3-10**
Categories per peer	**3-5**
Files per peer at initialization	**10-30**
Trials per experiment	**10**

Experimental Data

Enriching P2P File Descriptors using Association Rules Mining on Query Logs

- MRR increases up to a point (5% confidence) and then eventually declines.
- Up to 15% increase in MRR (with 10,000 query query-log) MRR increase of 5.5% with 1000 query query-logs.

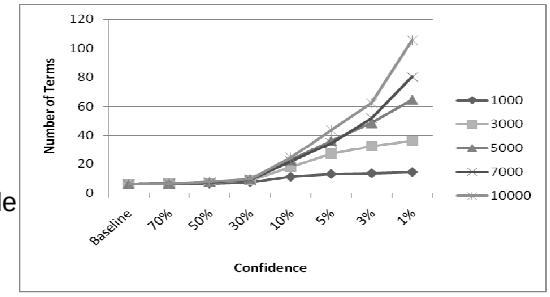


MRR versus confidence and query log size

Enriching P2P File Descriptors using Association Rules Mining on Query Logs

- Steep increase in descriptor size is observed for confidence Below 5%

- Descriptor size is manageable At 5% confidence and yields the highest MRR.



Descriptor size versus confidence and query log size

45