Topics in Database Systems: Data Management in Peer-to-Peer Systems

Peer-to-Peer Systems:

Semantic Clustering (Recup)

Γιατί θα μιλήσουμε σήμερα ..

Clustering

- περίληψη των 3 papers του προηγούμενο μαθήματος
- μερικά στοιχεία για το πως έχουμε «σημασιολογική ομαδοποίηση σε δομημένα p2p συστήματα

Μετά το Πάσχα ..

Database related:

advanced queries

Άσκηση για 17/5

Ένα **άρθρο επισκόπησης** ("survey") με θέμα «Συστήματα Ομότιμων Κόμβων»

- *Αυστηρά Ατομική* εργασία (αντιγραφή ⇒ μηδέν στο μάθημα)
- Θα περιλαμβάνει (τουλάχιστον) τα papers που διαβάσαμε μέχρι τώρα
- Θα ανανεωθεί στο τέλος του μαθήματος (με προσθήκη νέων άρθρων)
- *35% ή 40%* του βαθμού σας (15% το πρώτο μέρος – 20% ή 25% το δεύτερο και τελικό μετά τις διορθώσεις)

  έως και 50% αν δε δοθεί τελικό διαγώνισμα

Άσκηση για 17/5

Κάποιες οδηγίες (περισσότερα στη σελίδα μέχρι και 25/4)

- Μέγεθος έως 3000 λέξεις (πρώτη έκδοση)
- Δομή κανονικού άρθρου

  δηλαδή,

  Περίληψη (abstract)

  Εισαγωγή,

  Ενότητες x-u,

  ...

  Συμπεράσματα
- Στα αγγλικά ή στα ελληνικά

Άσκηση για 17/5

Κάποιες οδηγίες (συνέχεια)

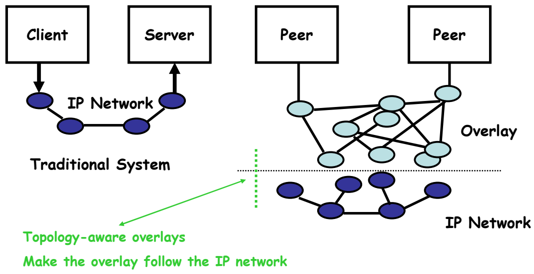
- Όχι μια ενότητα ανά paper – το άρθρο σας πρέπει να είναι ενοποιημένο, να διαβάζεται όπως ένα κεφάλαιο σε διδακτικό βιβλίο
- Συγκεντρωτικοί πίνακες, ταξινομήσεις κλπ θα βαθμολογηθούν θετικά
- Απαραίτητη η χρήση *κοινής* ορολογίας
- Χρήση «τμημάτων» από άλλες ερευνητικές εργασίες ή άρθρα επισκόπησης πρέπει να αναφέρεται άμεσα

  (π.χ. bla bla [xx] ή

  όπως αναφέρεται στο [xx], bla bla ..
- Αντιγραφή (μέρους ή όλου) από άλλες ερευνητικές εργασίες ή άρθρα επισκόπησης ΑΠΑΓΟΡΕΥΕΤΑΙ ΑΥΣΤΗΡΑ (⇒ μηδέν στο μάθημα)

Semantic Clustering of Peers

P2P Overlays



Semantic Overlay Networks

Unstructured networks: each node connects to some random nodes – what if we *cluster* nodes based on their *content, interests, previous queries*?

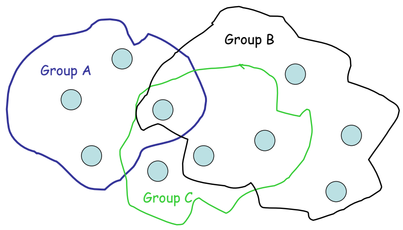
IDEA:

Build "topic" groups or sub-networks

Two step routing procedure:

- Identify the appropriate group
- Routing inside the group

Semantic P2P Overlays



- Intra-group routing
- Inter-group routing

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

- Non DHT-based (unstructured)
- Clustering on content
- Supports content hierarchies (classification) and layered SONS

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Cluster nodes and not content

That is, groups (clusters) of nodes

Content is not moved

Each node n_i maintains a set of documents D_i

Based on their documents nodes join specific SONs

Note, two types of queries

Exhaustive queries (return all documents matching a query)

Partial queries (return a minimum number of results)

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Builds a **number of overlays** (not just one)

a link between two nodes n_i and n_j has a *label* l indicating the overlay

Goal:

Define this set of overlay networks such that, given a query, we can select a small number of overlay networks whose nodes have a high number of hits

(how routing inside each overlay is performed is not discussed)

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Classification hierarchies: a tree of concepts

Example of three classification hierarchies for music documents

- One SON per concept of the hierarchy (e.g, 9 for the one in the left)
- Each query and document is classified into one or mode *leaf* concepts in the hierarchy

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Document and Query Classification

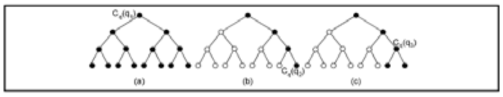
- May be **imprecise**: returns a non-leaf node A: the document (or the query) belongs to one or mode descendant of A, but the classifier cannot determine which one
- May make **mistakes**: return the wrong concept

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Document Classification

- differential assignment: place the document only in the concept that it belongs
- total assignment: in addition, place the document in all ancestors of the concept and all its descendants

Differential assignments makes query assignment more complicated, why?



Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Node Classification

- based on the classification of its documents
- conservative (place a node in the SON for concept c, if at least one document in concept c) – less conservative (a significant number of documents in c)
  - reduces number of nodes per SON
  - but, may loose results

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

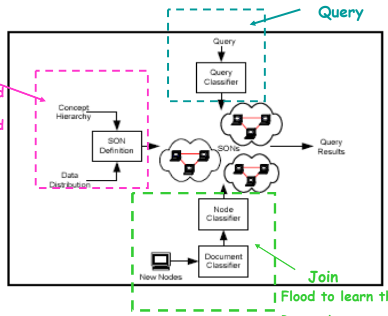
"Global procedure"

Find a good classification hierarchy and store it

Query

Run a query classifier

Sent it to the appropriate SONs



Join

Flood to learn the hierarchies

Run a document classifier

Join each SON

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Issues

Query vs documents classifiers

query classifiers must be fast and maybe imprecise, document classifiers many not be so fast but need to be more precise (in addition they are "bursty"

What is a "good" classification hierarchy

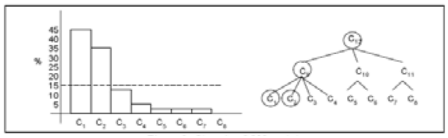
(i) produces buckets of documents that belong to a small number of nodes

(ii) nodes have documents in a small number of buckets

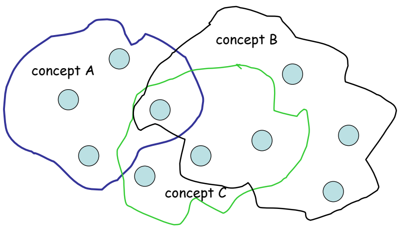
(iii) there exist efficient classifiers

Semantic Overlay Networks (SONs) for P2P [Crespo&Garcia-Molina03]

Layered SONs



Semantic P2P Overlays



Based on concepts from a *predefined* concept hierarchy

Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems [Sripanidkulchai et al, Infocom03]

- Non DHT-based, but can also be applied to DHT-based (*Does this hold for SONs? How?*)
- Clustering on previous results (interests)
- On top of Gnutella, additional connections among nodes

Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems [Sripanidkulchai et al, Infocom03]

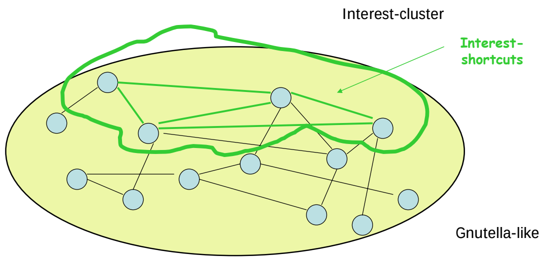
Each node, creates a short-cut list:

One of the nodes with matching results is selected at random and added in the short-cut list

Replacement based on perceived utility

Interest-based P2P Overlays

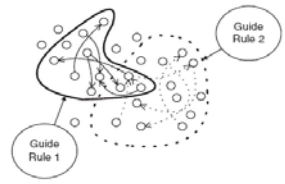
Results in clusters in the shortcut graph that correspond to clusters of interests



Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

- Non DHT-based
- Clustering based on content (Guide/Possession Rules)

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]



Guide Rule: set of peers that satisfy some predicate

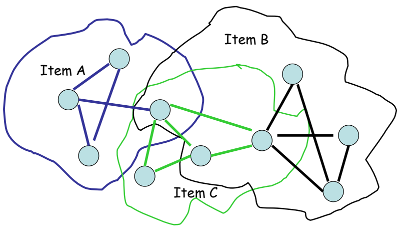
In the paper, a special form of guide rules based on the content of nodes:

Possession Rule: each associated with a data item – the predicate is the presence of the item in the node

Eg Rule(A)

Node n has item A

Possession-Rules P2P Overlays



One cluster per item

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

Two step routing procedure:

- STEP 1: The originating peer decides *which* guiding rules among those it belongs to, to use
- STEP 2: Routing inside each routing rule is blind (Gnutella-like)

A search strategy defines a search process as a sequence of guide rules and extent of search within each rule

Many propagation rules may be needed

E.g. search 100 peers that have item A and 200 paper peers that have item B, if this is unsuccessful, then search 400 ….

Unclear how they are specified

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

- Expectation: Large number of guide rules, but each peer uses a bounded number (?)
- Each guide rule corresponds to a large connected component
- Each peer may keep track of many other peers, proportional to the guide rules it belongs to
  - a neighbor list of the (item, peer) pairs for most items in its index
  - how it creates it?
    - Iteratively searches for the items it has

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

Peer26

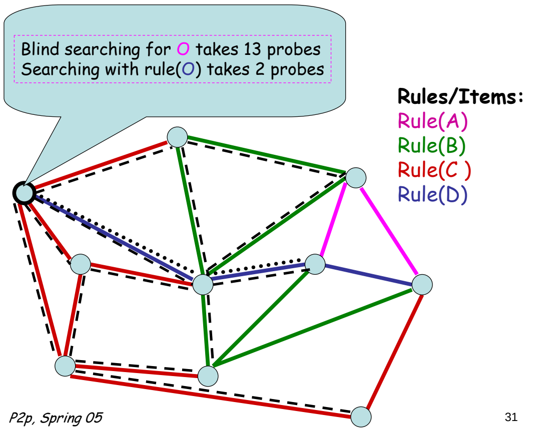
item	Rule(item) neighbors
A	p11,p7,p3
B	p2,p6,p9
C	p13,p15,p1
D	p4,p5,p10

Index of P26 Rules/Items:
Rule(A)
Rule(B)
Rule(C)
Rule(D)

Example Search Strategy of P26:

2 hops in rule(A)
4 hops in rule(B)
6 hops in rule(C)

4 hops in rule(A)
3 hops in rule(D)



Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

RAPIER

- STEP 1: *(The originating peer decides which guiding rules among those it belongs to, to use)*

Choose a *random* item from its index (i.e. a guiding rule uniformly at random)

- STEP 2: *(Routing inside each routing rule is blind - Gnutella-like)*

Perform a blind search on the possession-rule for the item to some *predefined depth*

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

Goal: compare RAPIER with

URAND: blind search, all peers equally liked to be probed

PRAND: the likelihood that a peer is probed is proportional to the size of its index – WHY?

RAPIER is biased towards searching in peers with many items (i.e many guide rules). Is that enough? Is it OK if we just choose nodes with many items (no guide rules)?

Caveat: comparing apples and oranges

- When searching by possession rules we have bias towards peers that participate in more rules/ have more items.
- But, with this bias, a strategy has better chance of finding what it is looking for! So…
- We show that the likelihood of being probed is proportional to number of rules you participate in.
- Prand "blind search" strategy has same bias.
- Thus, it is "fair" to compare Prand search with possession-rule based RAPIER

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

ANALYSIS Itemsets Model

Items belong to "topics." There are very many topics; but each peer can only select items from a fixed set of topics. Topic popularities can highly vary; but each peer has equal interest in each of "its" topics.

Show that

- RAPIER is at least as good as PRAND
- RAPIER is better than PRAND when peers have fewer topics
- Simple model that hints on what is going on…

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

ESS (Expected Search Size)

1/(success probability in each probe)

(when probes are "independent")

Probe success probability:

- URAND: fraction of peers that have the item in their index
- PRAND: the *weight* of each peer is its index size divided by sum of index sizes of all peers.
  - Success prob: (weight of peers with item) / (weight of peers without item)
- RAPIER: the average, over possession rules peer participates in, of fraction of peers in rule that have the item.

Associative Search in Peer-to-Peer Networks: Harnessing Latent Semantics [CohenFiatKaplan, Infocom03]

Items

Peers

0	0	?	?	?	0	0	0	0	0
0	0	0	0	0	?	0	0	?	?
?	?	0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0	1	0
0	0	0	0	0	0	1	1	1	0
1	1	0	0	0	0	0	0	1	0
0	0	0	1	1	0	0	1	1	1
0	0	1	1	0	0	0	0	1	0
1	1	0	0	0	1	0	0	0	0
0	1	0	0	1	0	0	0	1	0

Peer-Item Matrix

URAND and PRAND

Urand Ps=3/9 ESS=3

Prand ESS=29/9

Items

Peers

1/9	0	0	1	1	1	0	0	0	0	0	3/29
1/9	0	0	0	0	0	1	0	0	1	1	3/29
	1	?	0	0	0	0	1	0	0	0	
1/9	0	0	1	0	1	0	0	0	1	0	3/29
1/9	0	0	0	0	0	0	1	1	1	0	3/29
1/9	1	1	0	0	0	0	0	0	1	0	3/29
1/9	0	0	0	1	1	0	0	1	1	1	5/29
1/9	0	0	1	1	0	0	0	0	1	0	3/29
1/9	1	1	0	0	0	1	0	0	0	0	3/29
1/9	0	1	0	0	1	0	0	0	1	0	3/29

RAPIER (Random Possession Rule)

rule 0.5 Items rule 0.5

Peers

	0	0	1	1	1	0	0	0	0	0
	0	0	0	0	0	1	0	0	1	1
	1	?	0	0	0	0	1	0	0	0
	0	0	1	0	1	0	0	0	1	0
0.5	0	0	0	0	0	0	1	1	1	0
0.25	1	1	0	0	0	0	0	0	1	0
	0	0	0	1	1	0	0	1	1	1
	0	0	1	1	0	0	0	0	1	0
0.25	1	1	0	0	0	1	0	0	0	0
	0	1	0	0	1	0	0	0	1	0

What is latent semantics?

Selections people make are dependent:
- If you buy baby formula, you are more likely to buy diapers.
- If two people loved a show, they are more likely to agree on other shows.

- Peer/Item matrix is "Market Basket" dataset. Similar to buyers/items, Document/terms, Web-pages/hyperlinks, movies/viewers.
- Applications for extracting patterns from market basket data: Information Retrieval, Collaborative Filtering, Web search, Marketing, Recommendation Systems,.... (clustering, search, association rules)

?? P2P search – direct queries to peers with interests that match yours

Remarks

- semantic proximity between peers:
  - similarity between their cache contents or download patterns
- IDEA: semantically related peers are more likely to be useful to each other
- Use a predefined classification (SONs), semantic shortcuts (peers that share interests), possession rules (peers that share documents)

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

- DHT-based
- Placement of peers in the DHT not based on their ID but on their content
- Placement of documents (or indexes (of documents) on nodes based on their content, not just their ID (keyword, title)
- **How: For each document create a vector and use this vector to place the document**

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

How to create the vector for each document: Vector Space Model (VSM)

Documents and queries are represented as Term Vectors

- Each elements of the vector corresponds to the importance of the term in the document (or the query)
- Statistical computation of vector elements
  - Term frequency * inverse document frequency

Ranking of retrieved documents

- Similarity between document vector and query vector

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Example with 4-term vectors

vocabulary	VA	VQ	VB
book	0.5	0	0
computer	0.5	0.5	0
network	0.8	0.8	0.9
routing	0	0	0.6

VA ↔ VQ: 0.89
VQ ↔ VB: 0.72

Document A: "books on computer networks"
Document B: "network routing in P2P networks"
Query Q: "computer network"

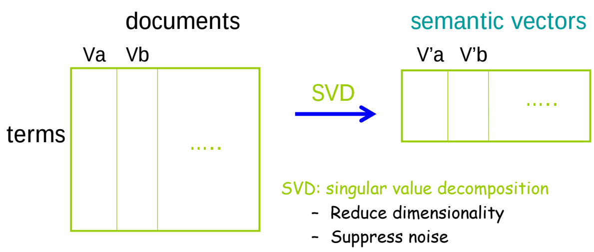
Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

VSM suffers from synonyms and noise in documents

Latent Semantics Indexing (LSI)

- Uses Singular Value Decomposition (SVD) to transform a high-dimensional term vector to a low-dimensional semantic vector (based on *abstract concepts*)
- Elements correspond to the importance of the abstract concept in document/query

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

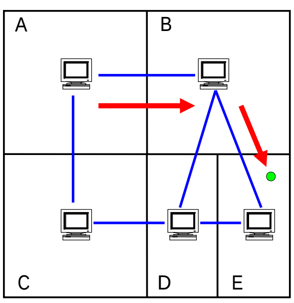


SVD: singular value decomposition

- Reduce dimensionality
- Suppress noise
- Discover word semantics
  - Car <-> Automobile

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Use CAN



CAN Overview

- Partition Cartesian space into zones
- Each peer is assigned to a zone
- Neighboring zones are routing neighbors
- An object key is a point in the space
- Object lookup is done through routing

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

pSearch Overview

- CAN: organize nodes into a semantic overlay
- LSI: generate semantic vectors
  - Used **as object key** to store doc indices in the CAN

Indices close in semantics are stored close in the overlay

- Two types of operations
  - Publish document indices (join)
  - Process queries (route)

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

pSearch Basic Algorithm: Setup

- Dimensionality of CAN = dimensionality of LSI's semantic space
- Index of documents:
  - key: document's semantic vector
  - value: reference (URL) to document

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

pSearch Basic Algorithm: Steps

Join:

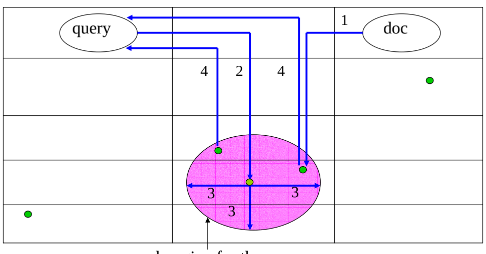
1. Receive a new document A: generate a semantic vector V_a, store the key in the index *(USE CAN)*

Route:

2. Receive a new query Q: generate a semantic vector V_q, route the query in the overlay *(USE CAN)*
3. The query is flooded to nodes within a radius *r*
   R determined by similarity threshold or number of wanted documents
4. All receiving nodes do a local search and report references to best matching document

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

pSearch Illustration



search region for the query

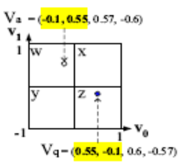
Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

Major Challenges

1. Dimensionality mismatch between CAN and LSI
   LSI: 50 – 350
   Many dimension are not partitioned: search space not reduced in these dimensions
2. Large search region
3. Uneven distribution of indices

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

Dimensionality Mismatch



We have only two dimensions – q is not similar with A in this two dimensions!

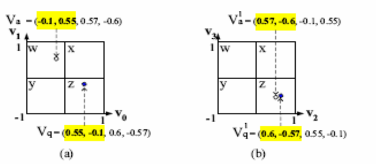
Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

Dimensionality Mismatch: **Rolling Index**

- Rotate vectors based on estimated *effective dimensionality* (number of actually partitioned dimensions) of the CAN
- Index the vector *p times*
- pLSI algorithm is executed *p* times for a query
- Does not affect *similarity* measure

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Dimensionality Mismatch: Rolling Index



We have only two dimensions – q is not similar with A in this two dimensions!

Rotate with m = 2

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Large Search Region

Curse of dimensionality:

In centralized index structures, the search space grows quickly as dimensionality of data increases.

Observations:

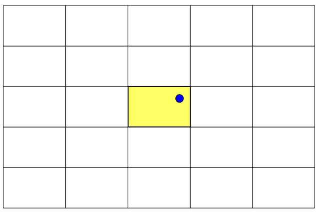
1. High-dimensional data spaces are sparsely populated
2. The distance between a query and its neighbors steadily grows with dimensionality

For a naïve nearest-neighbor search to work, a large number of nodes must be searched

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Content-directed Search

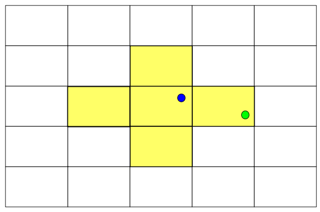
- Search the node whose zone contains the query semantic vector. (query center node)



Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Content-directed Search

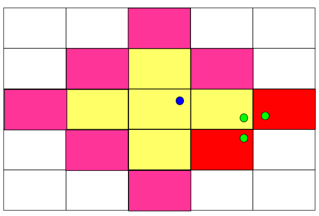
- Search direct (1-hop) neighbors of query center



Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Content-directed Search

- Selectively search some 2-hop neighbors
  - Focusing on "promising" regions suggested by samples



Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOM03]

Unbalanced Index Distribution

Solution: *content-aware node bootstrapping*

1. A new node randomly picks a document to publish
2. The node computes the semantic vector
3. The vector is rotated to a space *i*
4. The node containing the semantic vector splits in the middle giving half of the space to the new node

Effects of bootstrapping:

1. More balanced index distribution
2. Index locality (share content)
3. Query locality (share interests)

Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks [TangXuDwarkadas, SIGCOMM03]

Conclusion

- Map semantic space generated by modern IR algorithms atop overlay networks to enable efficient P2P search
  - pLSI is good at clustering documents
  - **Index locality: indices stored close in the overlay network are also close in semantics**