Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks

Chunqiang Tang, Zhichen Xu, and Sandhya Dwarkadas

SIGCOMM 2003

Presented by Keith Tayloe

Peer-to-Peer Information Retrieval

- Distributed Hash Table (DHT)
  - CAN, Chord, Pastry, Tapestry, etc.
  - Scalable, fault tolerant, self-organizing
  - Only support exact key match
    - K_d=hash ("books on computer networks")
    - K_q=hash ("computer network")
- Extend DHTs with content-based search
  - Full-text search, music/image retrieval
- Build large-scale search engines using P2P technology

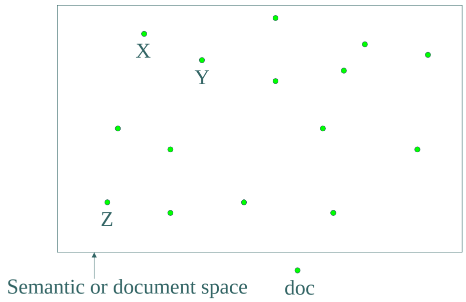
Focus and Approach in pSearch

- Efficiency
  - Search a small number of nodes
  - Transmit a small amount of data
- Efficacy
  - Search results comparable to centralized information retrieval (IR) systems
- Extend classical IR algorithms to work in DHTs, both efficiently and effectively

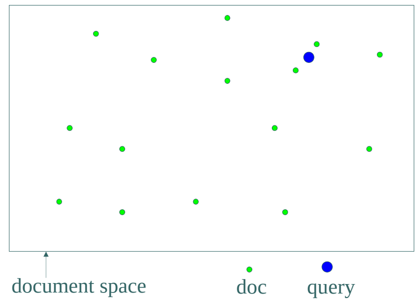
Outline

- Key idea in pSearch
- Background
  - Information Retrieval (IR)
  - Content-Addressable Network (CAN)
- Our P2P IR algorithm
- Experimental results
- Open issues and ongoing work
- Conclusions

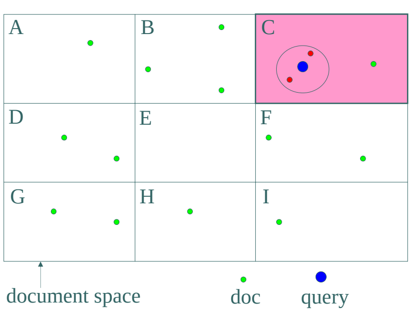
pSearch Key Idea



pSearch Key Idea



pSearch Key Idea



Outline

- Key idea in pSearch
- Background
  - Information Retrieval (IR)
  - Content-Addressable Network (CAN)
- Our P2P IR algorithm
- Experimental results
- Open issues and ongoing work
- Conclusions

Background

- Statistical IR algorithms
  - Vector Space Model (VSM) [Salton et al.]
  - Latent Semantic Indexing (LSI) [Deerwester et al.]
- Distributed Hash Table (DHT)
  - Content-Addressable Network (CAN) [Ratnasamy et al.]

9

Background: Vector Space Model

vocabulary	Va		Vq		Vb
book	0.5		0		0
computer	0.5	0.89	0.5	0.72	0
network	0.8		0.8		0.9
routing	0		0		0.6

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

10

Background: Vector Space Model

D1: How to Bake Bread Without Recipes
D2: The Classic Art of Viennese Pastry
D3: Numerical Recipes: The Art of Scientific Computing
D4: Breads, Pastries, Pies and Cakes: Quantity Baking Recipes
D5: Pastry: A Book of Best French Recipes

6 X 5 term-by-Document matrix

T1: bak(e,ing)
T2: recipes
T3: bread
T4: cake
T5: pastr(y,ies)
T6: pie

$$A = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 0.5774 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.5774 & 0.0000 & 1.0000 & 0.4082 & 0.7071 \\ 0.5774 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.4082 & 0.7071 \\ 0.0000 & 0.0000 & 0.0000 & 0.4082 & 0.0000 \end{pmatrix}$$

With Unit Columns

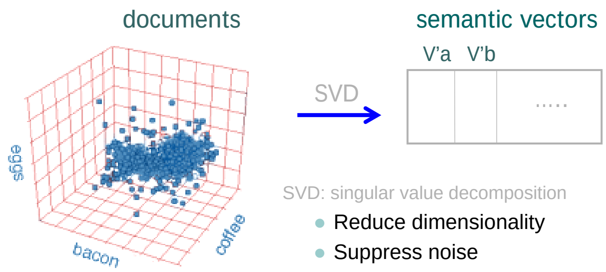
11

Background: Vector Space Model

- Example query = *baking*
  $q^{(1)} = (1\ 0\ 0\ 0\ 0\ 0)^T$
- Search for relevant documents is carried out by computing the cosines of the angles θ_j between the query vector $q^{(1)}$ and the document vectors a_j
- Results: only nonzero cosines are $\cos\theta_1 = 0.5774$ and $\cos\theta_4 = 0.4082$

12

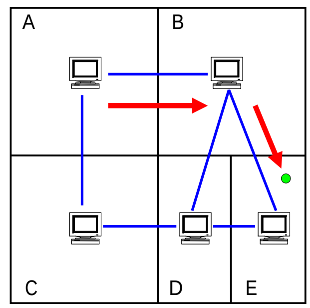
Background: Latent Semantic Indexing



SVD: singular value decomposition

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

Background: Content-Addressable Network



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

14

Outline

- Key idea in pSearch
- Background
  - Information Retrieval (IR)
  - Content-Addressable Network (CAN)
- Our P2P IR algorithm
- Experimental results
- Open issues and ongoing work
- Conclusions

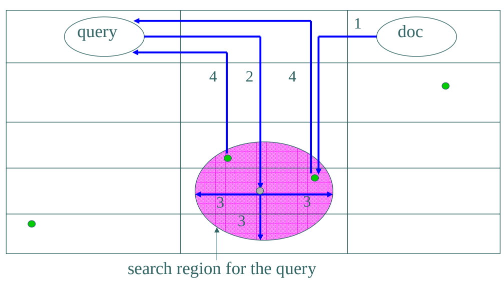
15

pLSI Basic Idea

- Use a CAN to organize nodes into an overlay
- Use semantic vectors generated by LSI as object key to store doc indices in the CAN
  - Index locality: indices stored close in the overlay are also close in semantics
- Two types of operations
  - Publish document indices
  - Process queries

16

pLSI Illustration



Major Challenges

- Dimensionality mismatch between CAN and LSI
  - Large search space
- The curse of dimensionality
  - Inefficient searching
- Uneven distribution of document indices
  - Inefficient routing and unbalanced load

pLSI Enhancements

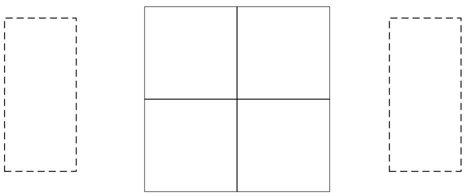
- Further reduce nodes visited during a search
  - Multi-plane (Rolling-index)
  - Content-directed search
- Balance index distribution
  - Content-aware node bootstrapping

Multi-plane (rolling index)

- 4-d semantic vectors

Multi-plane (rolling index)

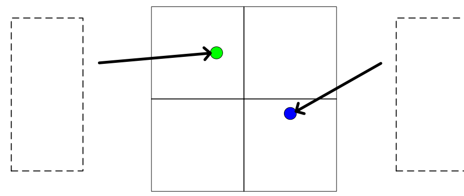
- 4-d semantic vectors
- 2-d CAN



21

Multi-plane (rolling index)

- 4-d semantic vectors
- 2-d CAN



22

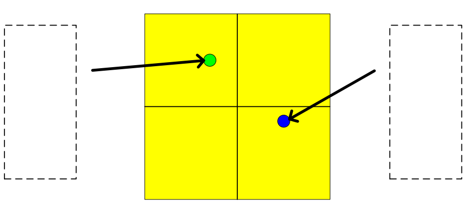
doc

1

query

Multi-plane (rolling index)

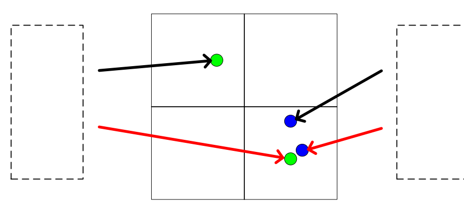
- 4-d semantic vectors
- 2-d CAN



23

Multi-plane (rolling index)

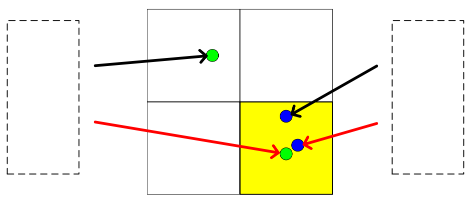
- 4-d semantic vectors
- 2-d CAN



24

Multi-plane (rolling index)

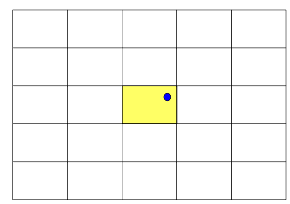
- 4-d semantic vectors
- 2-d CAN



25

Content-directed Search

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



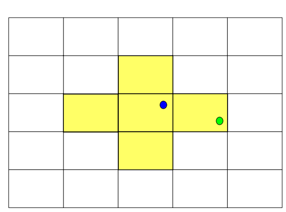
26

doc 1

query

Content-directed Search

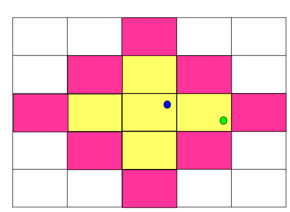
- Search direct (1-hop) neighbors of query center



27

Content-directed Search

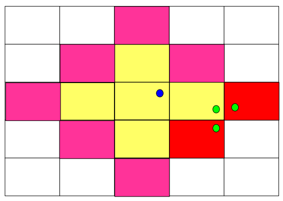
- How about 2-hop neighbors of query center?



28

Content-directed Search

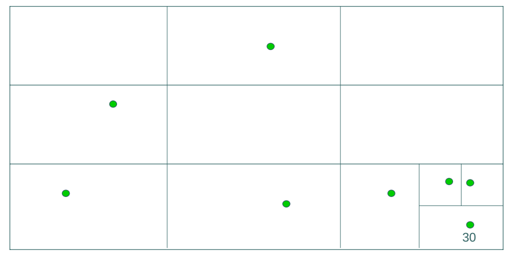
- Search direct (1-hop) neighbors; Selectively search some 2-hop neighbors
  - Focusing on "promising" regions suggested by samples



29

Content-Aware Node Bootstrapping

- pSearch randomly picks the semantic vector of an existing document for node bootstrapping



30

Outline

- Key idea in pSearch
- Background
  - Information Retrieval (IR)
  - Content-Addressable Network (CAN)
- Our P2P IR algorithm
- Experimental results
- Open issues and ongoing work
- Conclusions

31

Experiment Setup

- pSearch Prototype
  - Cornell's SMART system implements VSM
  - We extended it with implementations of LSI, CAN, and our pLSI algorithms
- Corpus: Text Retrieval Conference (TREC)
  - 528,543 documents from various sources
  - total size about 2GB
  - 100 queries, topic 351-450

32

Evaluation Metrics

- Efficiency: nodes visited and data transmitted during a search
- Efficacy: compare search results
  - pLSI vs. LSI
  - pLSI vs. best known IR algorithms

33

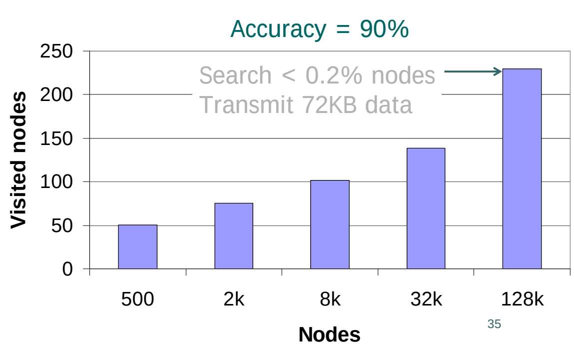
pLSI vs. LSI

$$\text{Accuracy} = \frac{|A \cap B|}{|A|} \times 100\%$$

- Retrieve top 15 documents
- A: documents retrieved by LSI
- B: documents retrieved by pLSI

34

Performance w.r.t. System Size



35

Performance & System Size

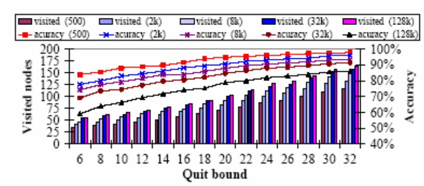


Figure 10: The effect of varying system size.

- Relaxing Quit Bound improves accuracy slowly with No. of nodes visited.
- Suggest results can be returned w/o waiting for query to reach final bound

36

Performance & Replication

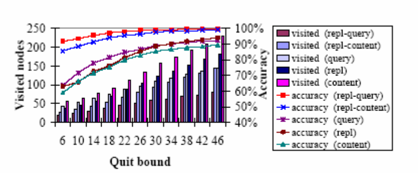


Figure 15: Performance of a 128k-node system.

- Accuracy of Content can approach 90% @ .2% of nodes
- W/replication and query heuristics can achieve 91.7% @ 19 nodes or 98% at 45 nodes.

37

Open Issues & Ongoing Work

- Larger corpora, other docs or queries
- Efficient variants of LSI/SVD: 1 hour->1min
- Evolution of global statistics
- Incorporate other IR techniques
  - Relevance feedback, Google's PageRank, Music and image retrieval
- Compare with other alternatives
  - pVSM [Tang et al., HotNets-I]

38

Conclusion

- We 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
- We introduced techniques to
  - Further reduce visited nodes: content-directed search & rolling index
  - Balance index distribution: content-aware node bootstrapping

39