P2P Networks
- DHTs - lookup for a key routed to site responsible for storing key
- Search query more complex than simple key lookup
  - Example query: "The Neptunes"

Distributed Information Retrieval
- Mediator-based
  - Architectures
  - Models
  - Wrappers (query rewriting)
  - Repository Selection
- Relatively Static
- 1000s of repositories

Solution: SETS
- Topic-segmentation based network providing efficient search
- Partition sites into topic segments
  - Sites with similar documents belong to same segment
  - Short distance links to sites within segment
  - Long distance links to sites outside of segment

SETS Architecture
- Topic Segments: Sites with similar content
  - Local routing network
    - Links within Topic Segments
  - Global routing network
    - Links to other Topic Segments
- Query propagation
  - Probe only relevant Topics

Quality of Topic Segments & Efficiency of topic-driven routing
- Topic Segment Construction
- Topic Segment Maintenance
- Topic Segment Selection
- Global Routing
- Local Routing
**Topic Segment Construction**

- **Document-vector**
  - Normalized to unit-length
  - No stop-words, No frequent-words
  - Each term weighted by log tf

- **Site vector**
  - Normalized sum of all its document vectors

- **Administration site A**
  - Passive support, e.g., code-maintenance.
  - Responsible for clustering vectors into topic segments
  - k-means clustering to generate C topic centroids

**Topic Segment Maintenance**

- How often?
  - Every T time-units

- "Leases" => Sites belong to
  - at most two different
  - sets of topic centroids

**Topic Segment Selection**

- **Query Vector**
  - Similarity between query vector and each topic centroid computed
  - Use cosine between vectors

- Top R <= C segments selected
- Query is routed to these R segments in parallel

**Global and Local Routing Networks**

- **My segment?**
  - Centroid closest to my site-vector

- **Local Network**
  - Inter-Topic-Segment
  - Application specific

- **Global Network**
  - Chord, etc.

- Dominant cost: Local
  - Experimental Results:
    - Broadcast to all segment members

**% Recall vs. C (#clusters)**

- Graph showing recall percentage vs. number of sites probed
- As C (#clusters) increases, recall decreases
- Cineser dataset (478K docs 84K sites)

**Ordering of Segments to be Probed**

- Graph showing recall percentage vs. number of sites probed
- AS-HClusters increased
- COSINE ordering
- RANDOM ordering (Baseline)

TREC-12-AP
- C = 128 segments
Choice of Clustering
Site-vector clustering
outperforms
Document-vector clustering

Why?
Heterogeneous sites
Doc-vector clustering => site classified into topic that dominates
Site-vector clustering => site classified by mixtures of topics

Load on A (Admin site)
t = 1000 terms per vector
T = 15 minutes (lease interval)
C = 256 clusters
N = #sites

20% TCP/IP overhead -- No compression of vectors
Inbound-traffic = 2 x 4t x N Bps
Outbound-traffic = 2 x 4t x C Bps

TI line (1.54 Mbps) can support 8800 sites
T3 line (44 Mbps) can support 256K sites
k-means for 84K sites on Pentium-II in 15 min

pSearch
- 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

pSearch Key Idea

Background:
Vector Space Model

<table>
<thead>
<tr>
<th>vocabulary</th>
<th>Va</th>
<th>Vq</th>
<th>Vb</th>
</tr>
</thead>
<tbody>
<tr>
<td>book</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>computer</td>
<td>0</td>
<td>0.5</td>
<td>0.72</td>
</tr>
<tr>
<td>network</td>
<td>0.8</td>
<td>0</td>
<td>0.9</td>
</tr>
<tr>
<td>routing</td>
<td>0</td>
<td>0</td>
<td>0.6</td>
</tr>
</tbody>
</table>

A: "books on computer networks"
B: "network routing in P2P networks"
Q: "computer network"
**Background: Latent Semantic Indexing**

documents | semantic vectors
---|---
\(V_a\) | \(V_b\)
\(\ldots\) | \(\ldots\)

SVD: singular value decomposition
- Reduce dimensionality
- Suppress noise
- Discover word semantics
- Car \(\leftrightarrow\) 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

**pLSI Illustration**

- Search region for the query

**Content-directed Search**

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

**Multi-plane (rolling index)**

- 4-d semantic vectors
- 2-d CAN
**Multi-plane (rolling index)**

- 4-d semantic vectors
- 2-d CAN

![Diagram of multi-plane indexing](image)

**Content-Aware Node Bootstrapping**

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

![Diagram of content-aware bootstrapping](image)

**Performance w.r.t. System Size**

- Accuracy = 90%
- Search < 0.2% nodes
- Transmit 72kB data

![Graph showing performance](image)

**Thoughts/Discussion**

- SETS
  - Distinguished node required (ask Tyson for solutions)
- pSearch
  - LSI computationally expensive
  - "Maintenance" - nodes joining/leaving?
- General
  - Dynamic content?

**Many of the slides borrowed from:**

- SETS
- pSearch