Searching over Many Sites in

Jeremy Stribling

Joint work with:
Jinyang Li, M. Frans Kaashoek, Robert Morris

MIT Computer Science and Artificial Intelligence Laboratory
Talk Outline

• Background: CiteSeer
• OverCite’s Design
  *(The Search for Distributed Performance)*
• Evaluation
  *(The Performance of Distributed Search)*
• Future (and related) work
People Love CiteSeer

- Online repository of academic papers
- Crawls, indexes, links, and ranks papers
- Important resource for CS community
People Love CiteSeer Too Much

http://citeseer.ist.psu.edu/cs?q=reliable+web+services

System busy. Try again later. Contact us if this problem persists.

Please try one of our mirrors at:

MIT
U of Zurich

Click here to retry, or read more about CiteSeer.

• Burden of running the system forced on one site
• Scalability to large document sets uncertain
• Adding new resources is difficult
What Can We Do?

• Solution #1: Let a big search engine solve it
• Solution #2: All your © are belong to ACM
• Solution #3: Donate money to PSU
• Solution #4: Run your own mirror
• Solution #5: Aggregate donated resources
Solution: OverCite

- A distributed, cooperative version of CiteSeer

→ Implementation/performance of wide-area search

CiteSeer Today: Hardware

- Two 2.8-GHz servers at PSU
CiteSeer Today: Search

Search keywords

Results meta-data

Context


CiteSeer: An Autonomous Web Agent for Automatic Retrieval of research publications, we have developed CiteSeer, an "assistant agent" which improves upon this. The operation of CiteSeer is relatively simple. Given a set of broadly topics, we retrieve the most relevant documents from the Web.


CiteSeer: An Automatic Citation Indexing System - Giles, Bollacker, Lawrence (1998) (26 citations)


Automating the Construction of Internet Portals with... - McCallum, Nigam, (2000) (24 citations)

and will complement similar sorts, such as CiteSeer (www.scienceindex.com) and the Computing Research Portal, have been developed. The CiteSeer project (Lawrence, Giles, Bollacker, 1999) has research papers based on words and citations. CiteSeer focuses on the domain of research papers, and www.ai.mit.edu/~jrennie/papers/cora-arXiv2000.ps.gz

Collaborative Filtering by Personality Diagnosis: A... - Pennock, Horvitz, (2000) (23 citations)
CiteSeer: Local Resources

<table>
<thead>
<tr>
<th># documents</th>
<th>675,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Index size</td>
<td>22 GB</td>
</tr>
<tr>
<td>Index coverage</td>
<td>5%</td>
</tr>
<tr>
<td>Searches</td>
<td>250,000/day</td>
</tr>
<tr>
<td>Document traffic</td>
<td>21 GB/day</td>
</tr>
<tr>
<td>Total traffic</td>
<td>34.4 GB/day</td>
</tr>
</tbody>
</table>

- Current CiteSeer capacity: 4.8 queries/s
- Users issue 8.3 queries/doc → 404 KB/s
  - Search is the bottleneck
Talk Outline

• Background: CiteSeer
• OverCite’s Design
  (The Search for Distributed Performance)
• Evaluation
  (The Performance of Distributed Search)
• Future (and related) work
Search Goals for OverCite

• Distributed search goals:
  – Parallel speedup
  – Lower burden per site
• Challenge: Distribute work over wide-area nodes
Search Approach

• Approach:
  – Divide docs into partitions, hosts into groups
  – Less search work per host
• Same as in cluster solutions, but wide-area
• Doesn’t sacrifice search quality for performance
• Not explicitly designed for the scale of the Web
The Life of a Query

Client

Web-based front end

Query Results Page

Keywords Hits w/ meta-data, rank and context

Meta-data req/resp

Group 1

Group 2

Group 3

Group 4

Index

DHT storage (Documents and meta-data)

Web-based front end

Client

Results Page

Query

Index

DHT storage (Documents and meta-data)

Group 1

Group 2

Group 3

Group 4

Keywords Hits w/ meta-data, rank and context

Meta-data req/resp
Local Queries

- Inverted index: words → posting lists
  <4-byte doc ID, 2-byte offset>
- DB: words → position in index
- Text file: full ASCII text for all documents

Query: “peer hash”

Result: Doc #1 w/ context
Parallelizing Queries

- Partition by document
- Divide the index into $k$ partitions
- Each query sent to only $k$ nodes
Considerations for $k$

- If $k$ is small
  - Fewer hosts $\rightarrow$ less network latency
  - Less opportunity for parallelism

- If $k$ is big
  - More parallelism
  - Smaller index partitions $\rightarrow$ faster searches
  - More hosts $\rightarrow$ some node likely to be slow
Talk Outline

• Background: CiteSeer
• OverCite’s Design
  *(The Search for Distributed Performance)*
• Evaluation
  *(The Performance of Distributed Search)*
• Future (and related) work
Deployment

• 27 nodes across North America
  – 9 RON/IRIS nodes + private machines
  – 47 physical disks

Map source: http://www.coralcdn.org/oasis/servers
Evaluation Questions

• What are the bottlenecks for local queries?
• Is wide-area search distribution worthwhile?
• Do more machines mean more throughput?
Local Configuration

• Index first 64K chars/document (78% coverage)
• 20 results per query
• One keyword context per query
• Total of 523,000 unique CiteSeer documents
• Average over 1000s of CiteSeer queries
Local: Index Size vs. Latency

- Context bottleneck: Disk seeks
- Search bottleneck: Disk tput and cache hit ratio
Local: Memory Performance

- Smaller index → better memory performance
Distributed Configuration

- 1 client at MIT
- 128 queries in parallel
- Average over 1000 CiteSeer queries
- Vary $k$ (number of machines used)
- Each machine has local index over $1/k$ docs
Distributed: Index Size vs. Tput

• Throughput improves, despite network latencies
Talk Outline

• Background: CiteSeer
• OverCite’s Design  
  \textit{(The Search for Distributed Performance)}
• Evaluation  
  \textit{(The Performance of Distributed Search)}
• Future (and related) work
Future Work

• Will throughput level off or drop as $k$ increases?
• How would many more nodes affect approach?
• Push to have a more “real” system
Related Work

• Search on unstructured P2P
  – [Gia SIGCOMM ’03, FASD ’02, Yang et al. ’02]

• Search on DHTs
  – [Loo et al. IPTPS ’04, eSearch NSDI ’04, Rooter WMSCI ’05]

• Distributed Web search
  [Google IEEE Micro ’03, Li et al. IPTPS ’03, Distributed PageRank VLDB ‘04 & ’06]

• Other paper repositories
  [arXiv.org (Physics), ACM and Google Scholar (CS), Inspec (general science)]
Summary

- Distributed search on a wide-area scale
- Large indexes (> memory) should be distributed
- Implementation and performance of a prototype

http://overcite.org