Searching over Many Sites in OverCite

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

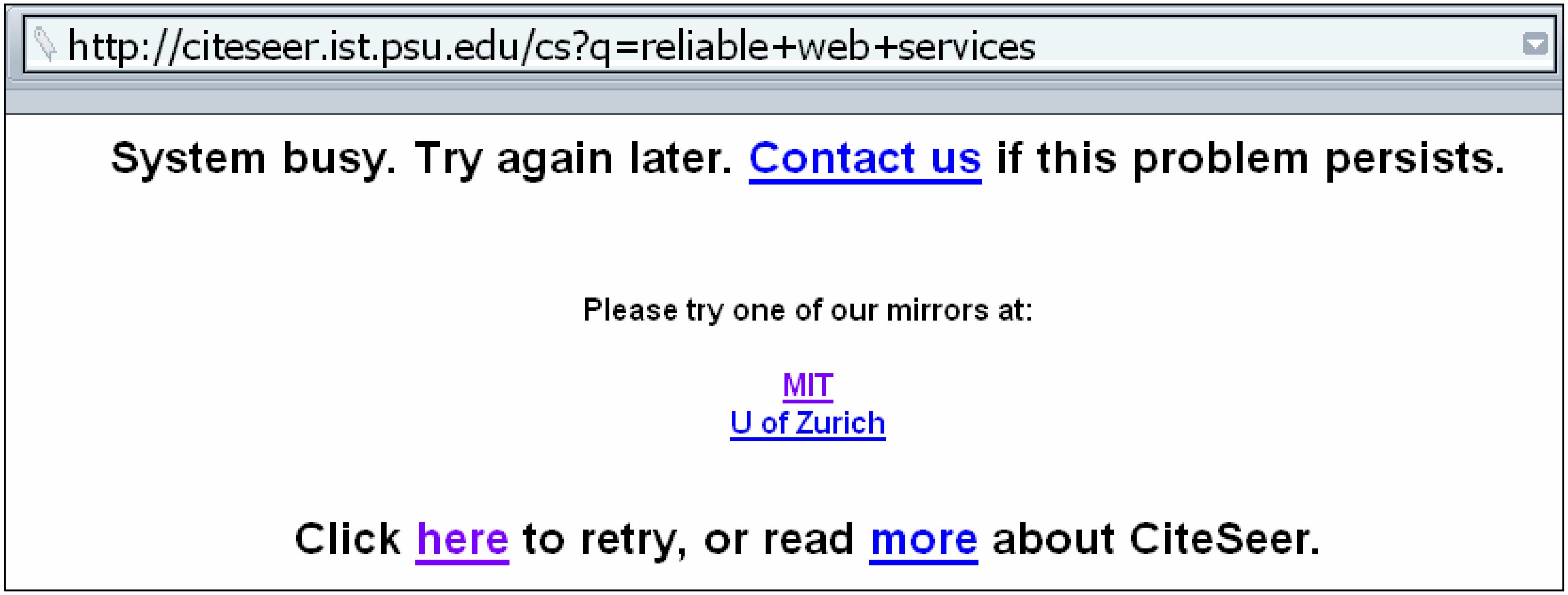
  (The Performance of Distributed Search)
- Future (and related) work

People Love CiteSeer

- Online repository of academic papers
- Crawls, indexes, links, and ranks papers
- Important resource for CS community

CiteSeer
Scientific Literature Digital Library
Find:
reliable web services
typical unification of access points and re
Documents

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.

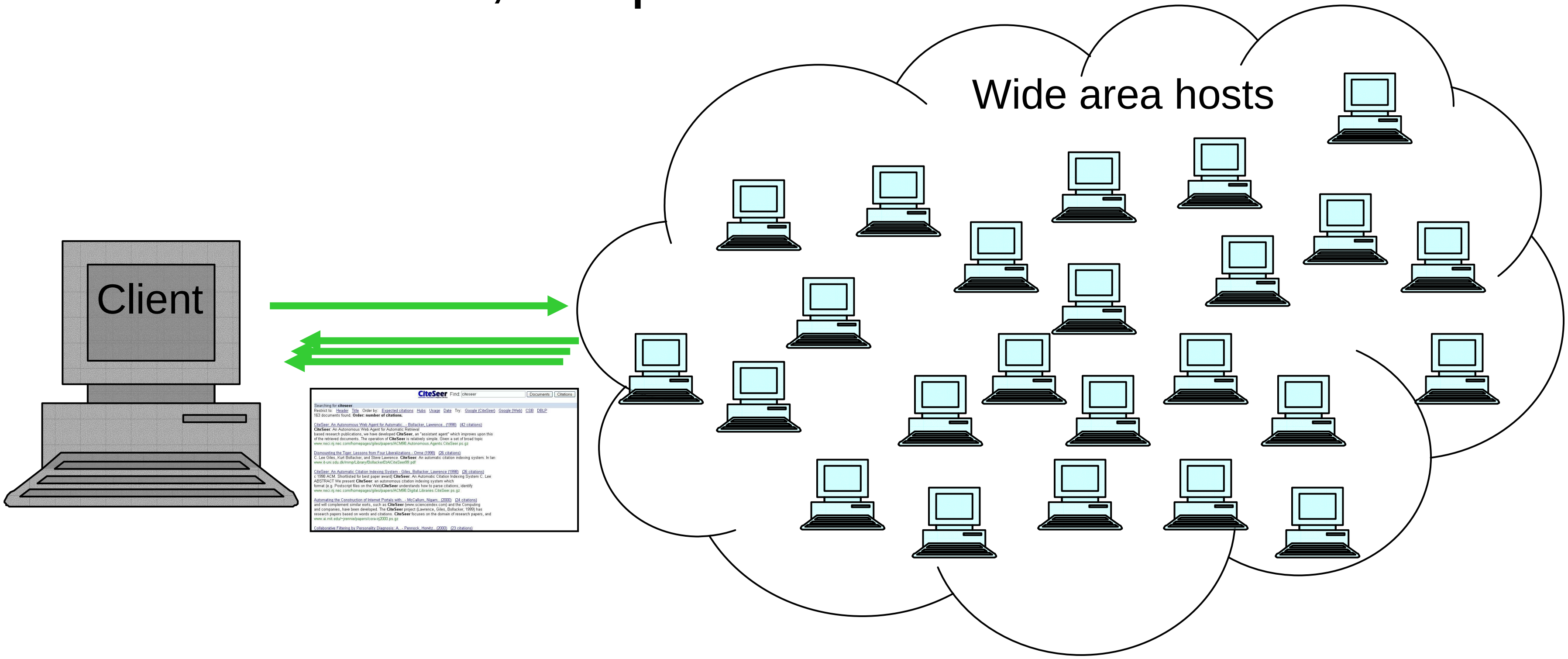
- Burden of running the system forced on one site
- Scalability to large document sets uncertain
- Adding new resources is difficult

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



Searching for **citeseer**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLP
163 documents found. **Order: number of citations.**

CiteSeer: An Autonomous Web Agent for Automatic.. - Bollacker, Lawrence.. (1998) (42 citations)
CiteSeer: An Autonomous Web Agent for Automatic Retrieval
based research publications, we have developed **CiteSeer**, an "assistant agent" which improves upon this
of the retrieved documents. The operation of **CiteSeer** is relatively simple. Given a set of broad topic
www.neci.nj.nec.com/homepages/giles/papers/ACM98.Autonomous.Agents.CiteSeer.ps.gz

Dismounting the Tiger: Lessons from Four Liberalizations - Orme (1998) (26 citations)
C. Lee Giles, Kurt Bollacker, and Steve Lawrence. **CiteSeer**: An automatic citation indexing system. In Ian
www.it-uni.sdu.dk/mmp/Library/BollackerEtAlCiteSeer99.pdf

CiteSeer: An Automatic Citation Indexing System - Giles, Bollacker, Lawrence (1998) (26 citations)
c 1998 ACM. Shortlisted for best paper award] **CiteSeer**: An Automatic Citation Indexing System C. Lee
ABSTRACT We present **CiteSeer**: an autonomous citation indexing system which
format (e.g. Postscript files on the Web)**CiteSeer** understands how to parse citations, identify
www.neci.nj.nec.com/homepages/giles/papers/ACM98.Digital.Libraries.CiteSeer.ps.gz

Automating the Construction of Internet Portals with.. - McCallum, Nigam.. (2000) (24 citations)
and will complement similar eorts, such as **CiteSeer** (www.scienceindex.com) and the Computing
and companies, 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-irj2000.ps.gz

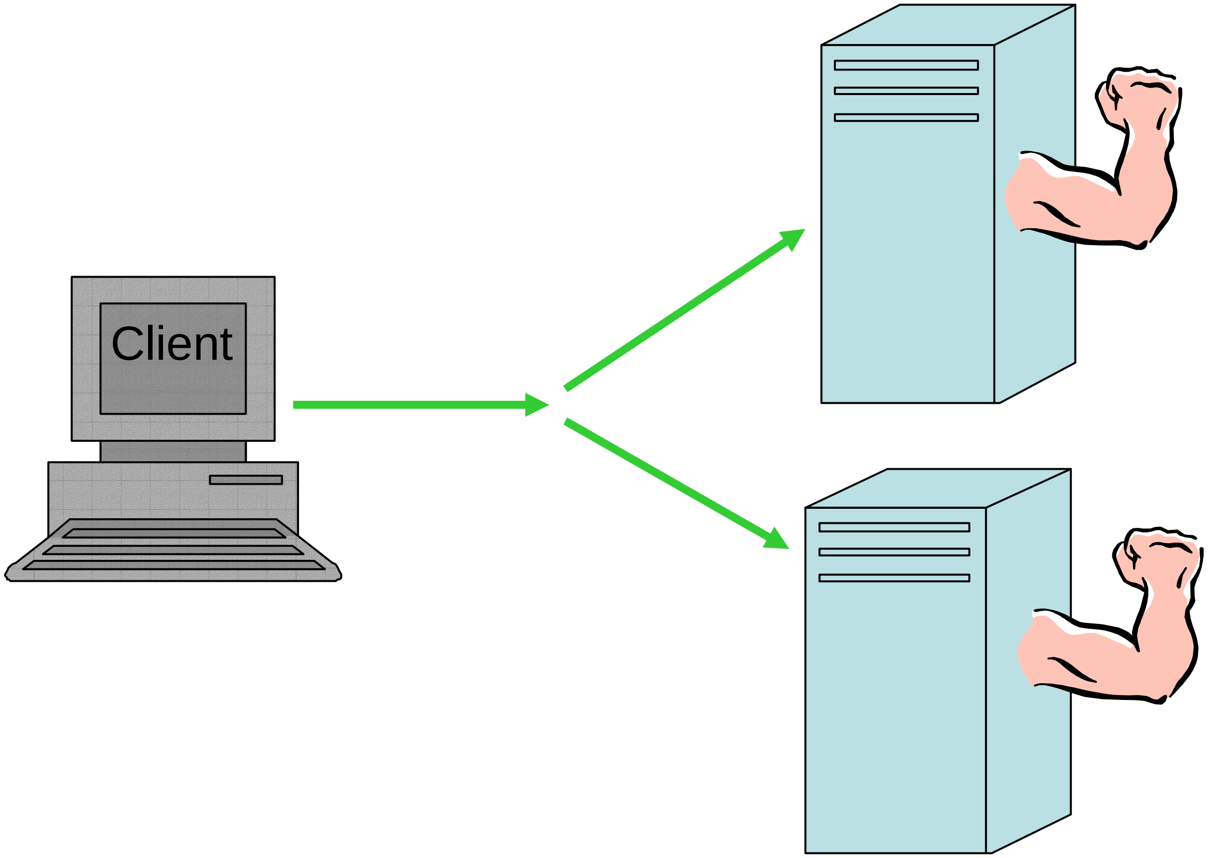
Collaborative Filtering by Personality Diagnosis: A.. - Pennock, Horvitz.. (2000) (23 citations)

→ Implementation/performance of wide-area search

OverCite: A Distributed, Cooperative CiteSeer. Jeremy Stribling, Jinyang Li, Isaac G. Councill, M. Frans Kaashoek, Robert Morris. *NSDI*, May 2006.

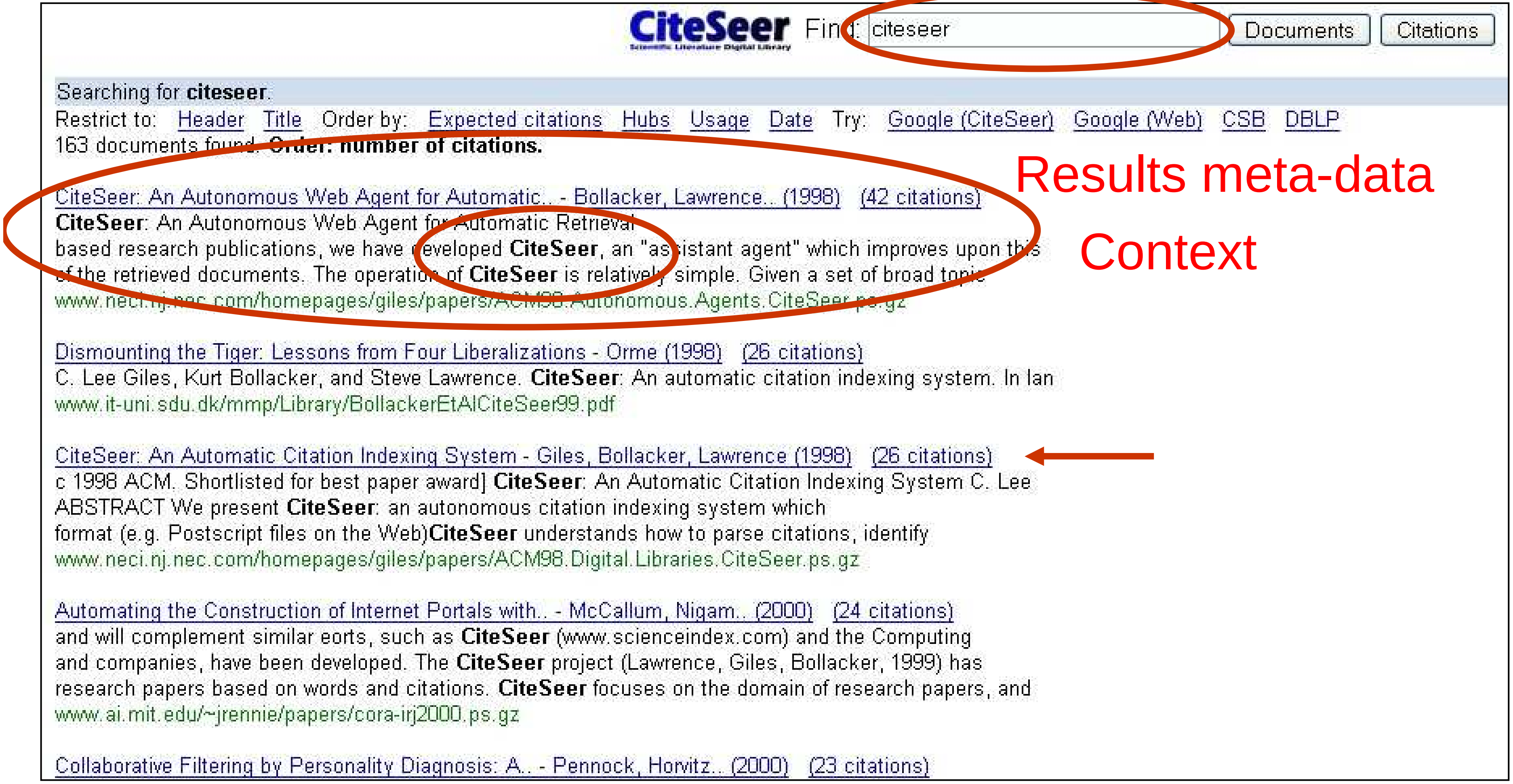
CiteSeer Today: Hardware

- Two 2.8-GHz servers at PSU



CiteSeer Today: Search

Search keywords



Searching for **citeseer**.

Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) CSB DBLP

163 documents found. **Order: number of citations.**

CiteSeer: An Autonomous Web Agent for Automatic.. - Bollacker, Lawrence.. (1998) (42 citations)
CiteSeer: An Autonomous Web Agent for Automatic Retrieval
based research publications, we have developed **CiteSeer**, an "assistant agent" which improves upon this
of the retrieved documents. The operation of **CiteSeer** is relatively simple. Given a set of broad topic
www.neci.nj.nec.com/homepages/giles/papers/ACM98.Autonomous.Agents.CiteSeer.ps.gz

Dismounting the Tiger: Lessons from Four Liberalizations - Orme (1998) (26 citations)
C. Lee Giles, Kurt Bollacker, and Steve Lawrence. **CiteSeer**: An automatic citation indexing system. In lan
www.it-uni.sdu.dk/mmp/Library/BollackerEtAlCiteSeer99.pdf

CiteSeer: An Automatic Citation Indexing System - Giles, Bollacker, Lawrence (1998) (26 citations)
c 1998 ACM. Shortlisted for best paper award] **CiteSeer**: An Automatic Citation Indexing System C. Lee
ABSTRACT We present **CiteSeer**: an autonomous citation indexing system which
format (e.g. Postscript files on the Web)**CiteSeer** understands how to parse citations, identify
www.neci.nj.nec.com/homepages/giles/papers/ACM98.Digital.Libraries.CiteSeer.ps.gz

Automating the Construction of Internet Portals with.. - McCallum, Nigam.. (2000) (24 citations)
and will complement similar eorts, such as **CiteSeer** (www.scienceindex.com) and the Computing
and companies, 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-irj2000.ps.gz

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

Results meta-data

Context

CiteSeer: Local Resources

# documents	675,000 ←
Index size	22 GB ←
Index coverage	5% ←
Searches	250,000/day ←
Document traffic	21 GB/day
Total traffic	34.4 GB/day ←

- 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

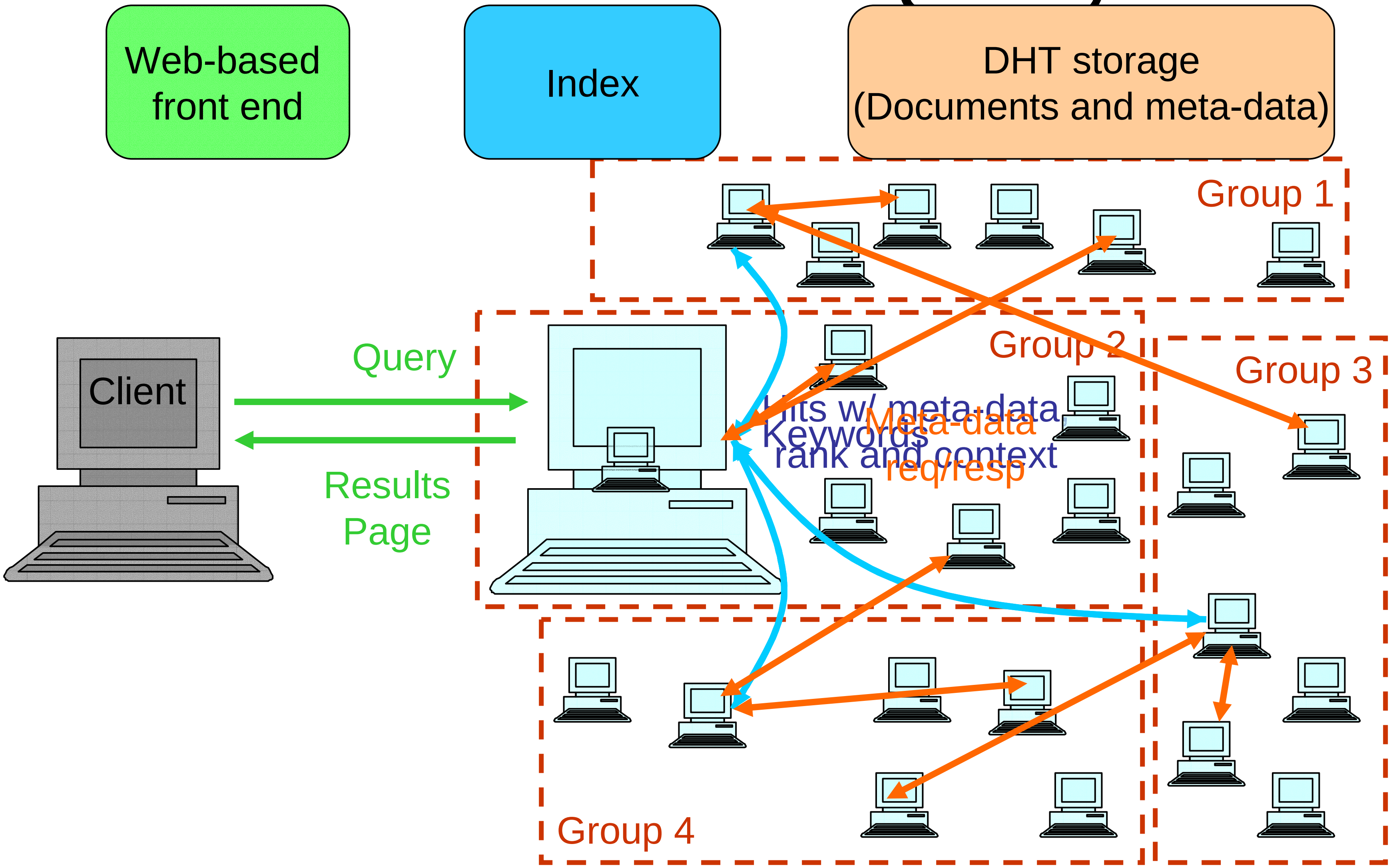
Search Goals for OverCite

- 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



Web-based front end

Index

DHT storage (Documents and meta-data)

Group 1

Group 2

Group 3

Group 4

Client

Query

Results Page

Hits w/ meta-data, rank and context

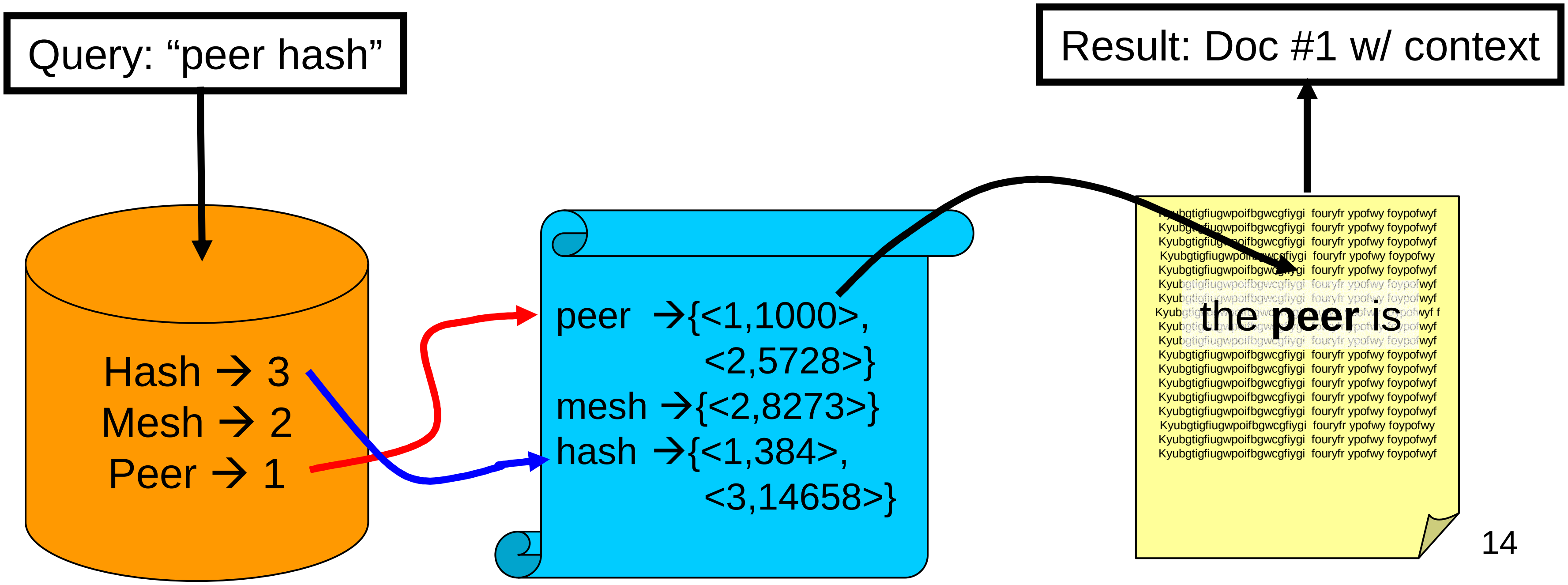
Keywords

Meta-data req/resp

Local Queries

- Inverted index: words → posting lists

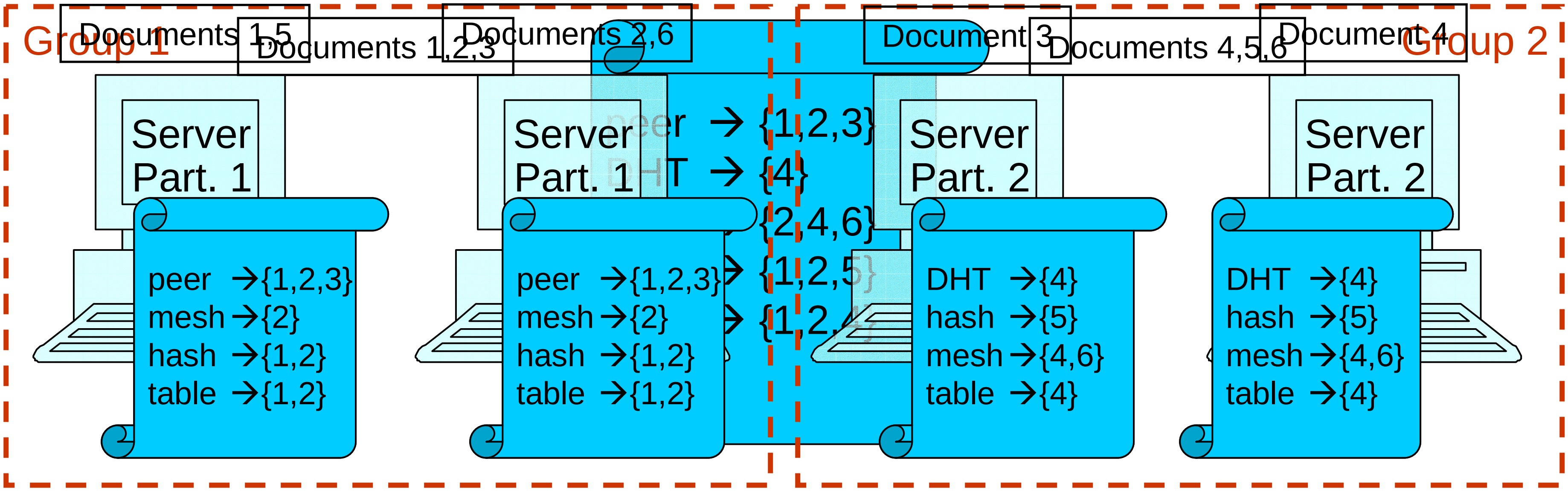
  <4-byte doc ID, 2-byte offset>
- DB: words → position in index
- Text file: full ASCII text for *all* documents



Query: "peer hash"
Hash → 3
Mesh → 2
Peer → 1
peer →{<1,1000>, <2,5728>}
mesh →{<2,8273>}
hash →{<1,384>, <3,14658>}
Kyubgtigfiugwpoifbgwcgfiygi fouryfr ypofwy foypofwyf
the peer is
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 → less network latency
  – Less opportunity for parallelism
- If *k* is big
  + More parallelism
  + Smaller index partitions → faster searches
  – More hosts → some node likely to be slow

Talk Outline

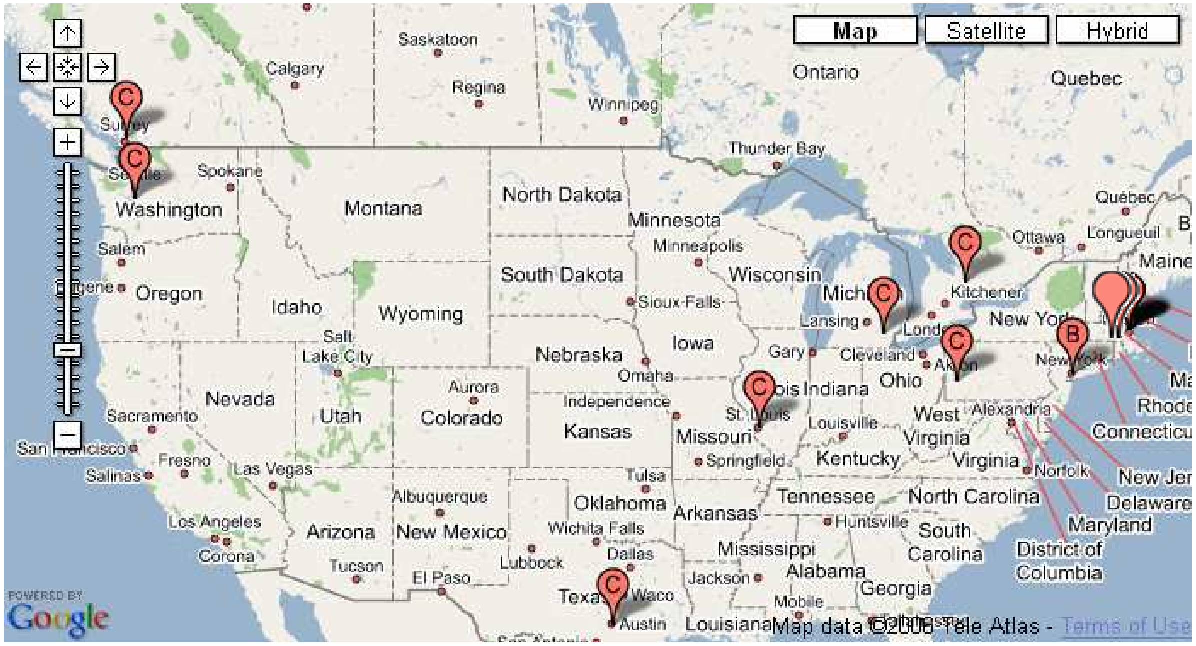
- Background: CiteSeer
- OverCite’s Design

  (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

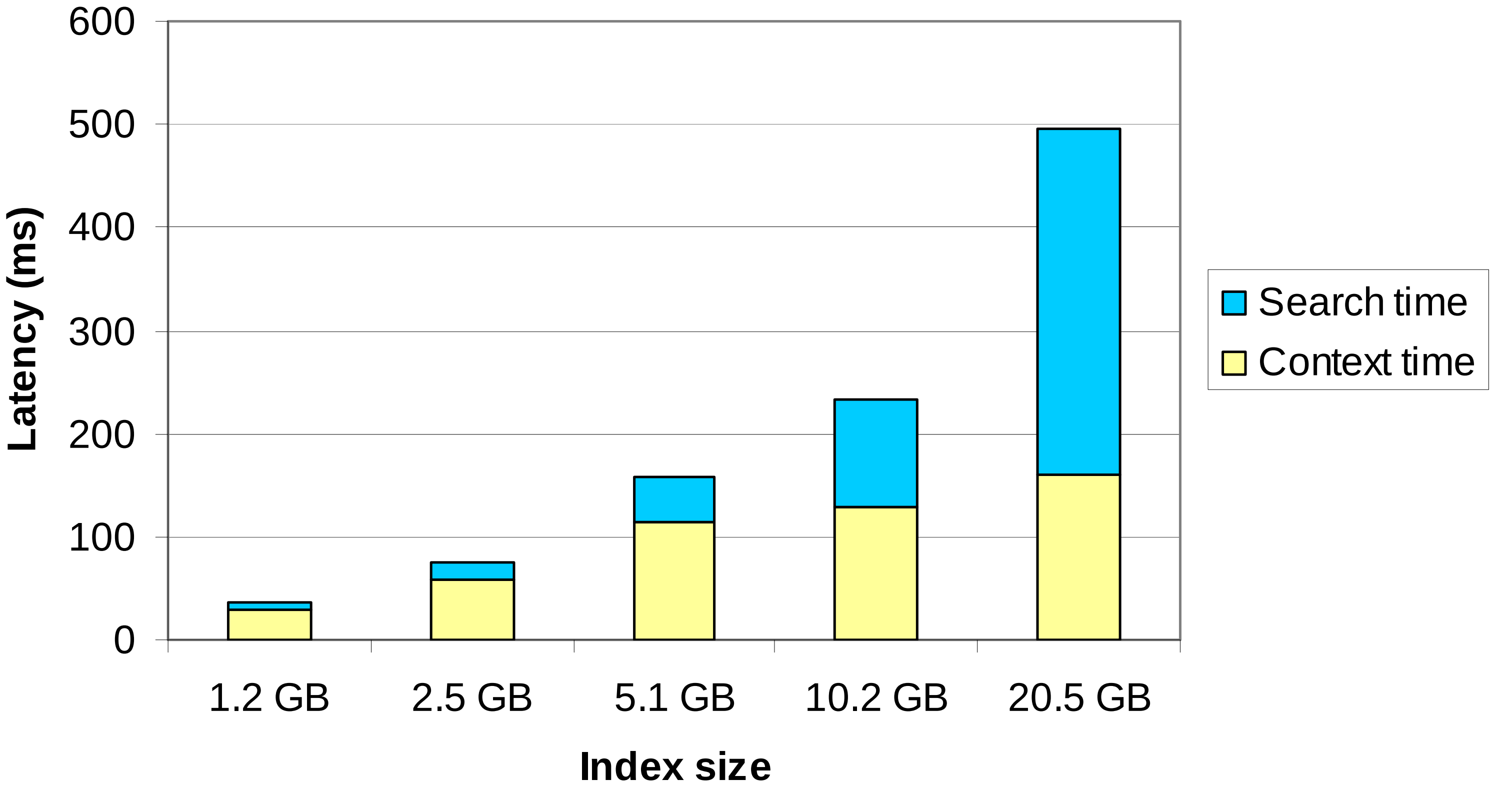
Evaluation Questions

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

Local Configuration

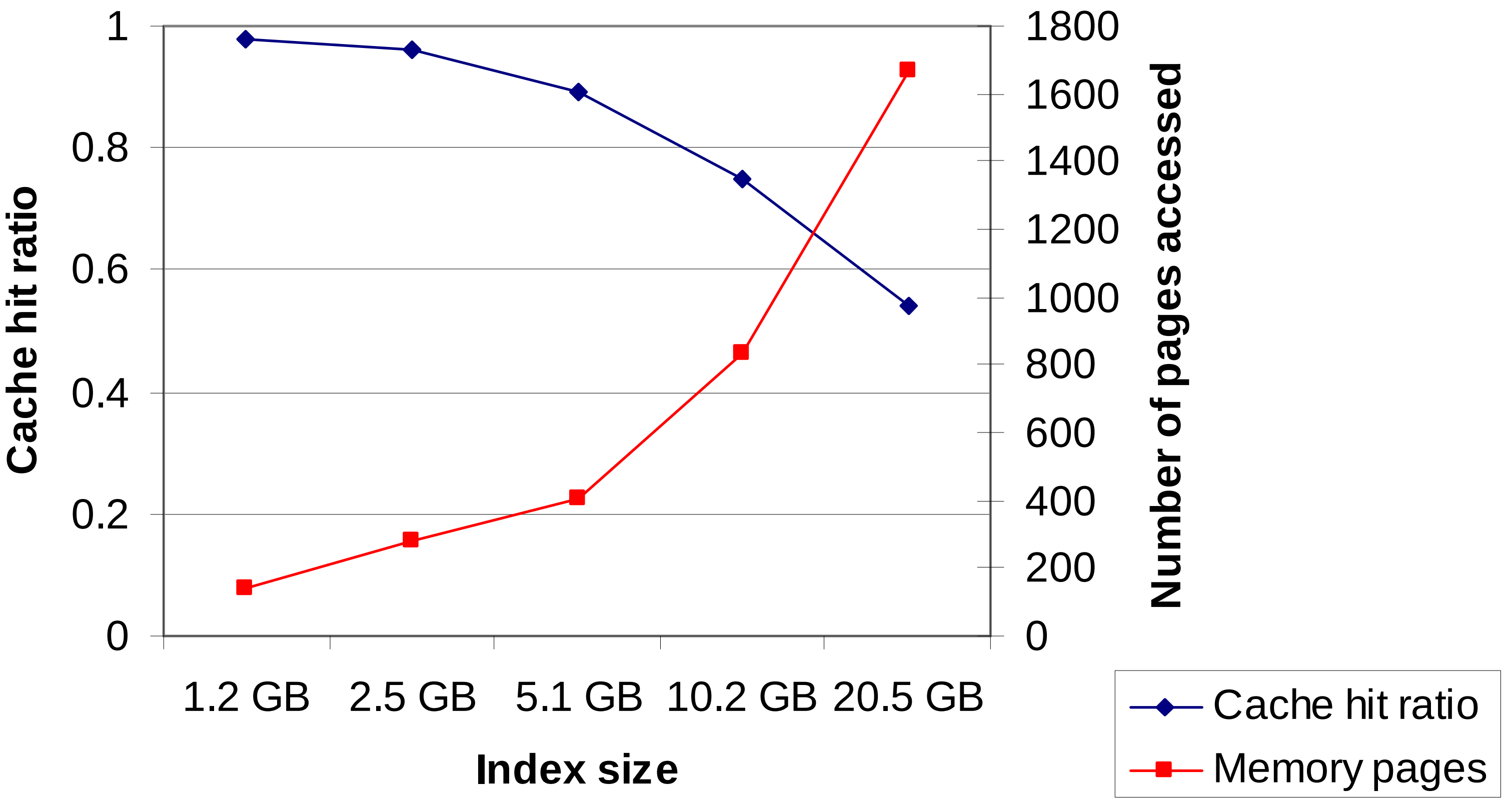
- Index first 64K chars/document (78% coverage)
- 20 results per query
- One keyword context per query
- Total of 523,000 unique CiteSeer documents
- Average over 1000s of CiteSeer queries

Local: Index Size vs. Latency



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

Local: Memory Performance

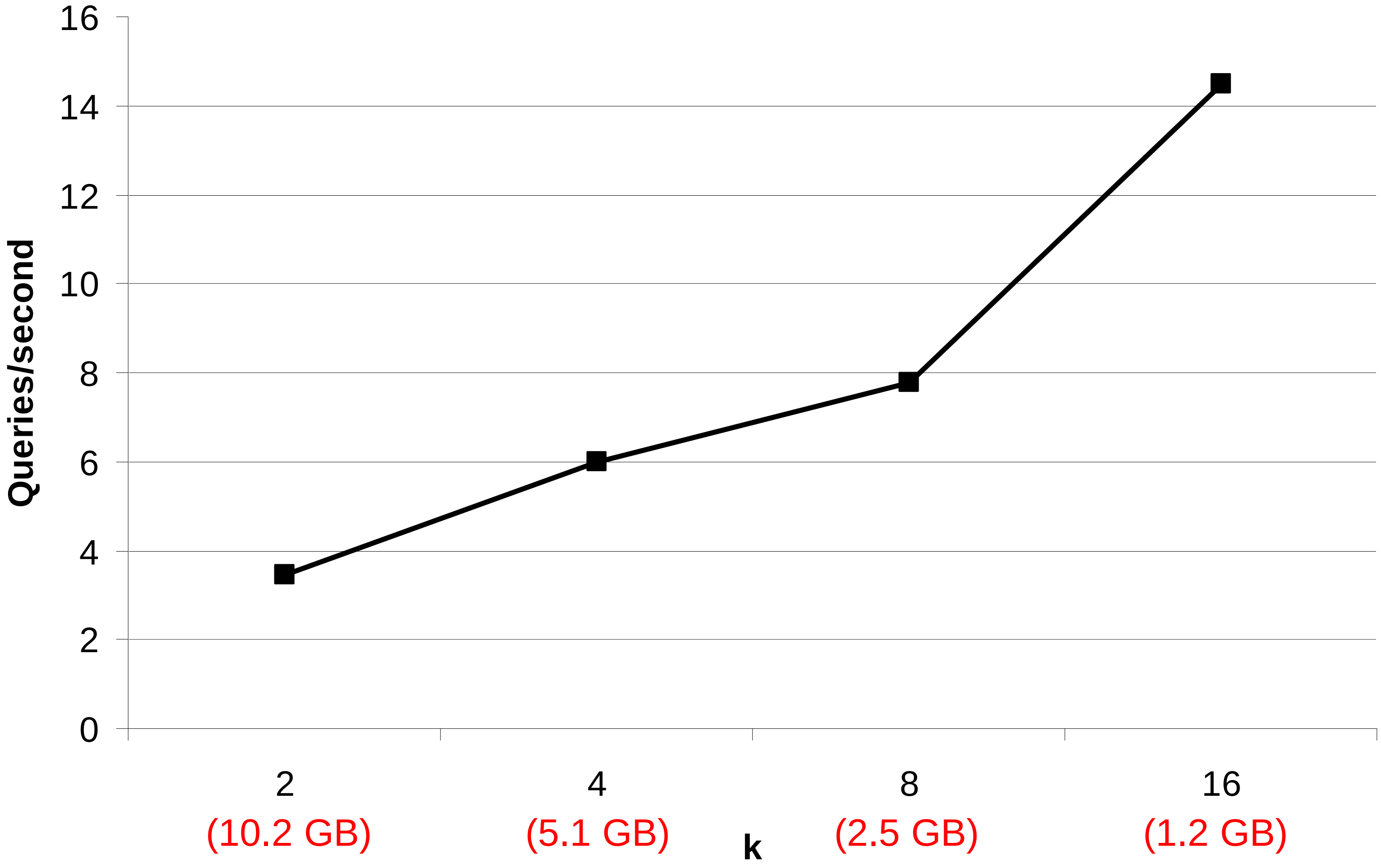


- Smaller index → better memory performance

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

  (The Search for Distributed Performance)
- Evaluation

  (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