Peer-to-Peer Systems

CS 257/457
Dept. of Computer Science, University of Rochester

Outline

- Object location discovery in large-scale distributed systems.
- Peer-to-peer systems
- Keyword search in peer-to-peer fashion

Case Study I: Napster (central index)

Centralized index, distributed document repository.

Case Study II: Gnutella (query flooding)

Non-hierarchical, equal status.

Case Study III: "hashing + heuristics-based routing"

No coordination among users, no central index.

"hashing + heuristics-based routing" - how does it work?

Things can go wrong!
Peer-to-peer Systems: Definition

- Distributed systems of no hierarchy, equal-status. (Napster is not! DNS is not!)
- Scalability: able to support large number of nodes
  - cost of each operation is non-scaling - goes up slowly along with the system size: \( O(n) \) is terrible, \( O(\log n) \) is good.
  - space requirement at each node is non-scaling.
- Tolerance of long-term or short-term node failures
  - losing a few objects is fine, but node failures should not cripple the system.

What makes peer-to-peer different?

- Compare with hierarchical, non equal-status systems:
  - better scalability - no performance bottleneck
  - better availability - more tolerant to random failures, intentional attacks, or censorship
  - less complex - every node bears the same responsibility
- Challenges:
  - peer coordination without complete global knowledge
- Analogy:
  - planned economy vs. market economy

Where are we now?

- Concept of peer-to-peer systems.
- Three approaches for object location discovery in large-scale distributed systems:
  - central index
  - query flooding
  - hash + heuristics-based routing
- Many other problems:
  - load balancing, replication, ...
- Keyword search in peer-to-peer fashion.

Keyword Search

- user inputs a few keywords, the system returns a list of documents matching the keywords - Google
- Google maintains a central search index:
  - a search index contains a list of all searchable words, each of which contains a list of documents relevant to the word
  - intersection of document lists for multiple-word queries

Peer-to-peer Keyword Search

- What is "peer-to-peer keyword search"?
  - splitting the central search index into smaller pieces and distribute them in peer-to-peer fashion
- Why is it better?
  - better scalability?
  - better availability?
  - less complexity?
- Challenges:
  - make them work collaboratively to achieve similar speed and quality of search

Peer-to-peer Keyword Search Solution 1: Split based on keywords

Split the index database to many pieces based on keywords and distribute them to many nodes.
Peer-to-peer Keyword Search

Solution 2: Split based on documents

Split the index database to many pieces based on documents and distribute them to many nodes.

How to build a peer-to-peer keyword search engine?

- Split based on keywords
  - Weakness: transferring large data segments for multiple keyword queries.
- Split based on documents
  - Weakness: too many sites to visit for each query.
- Neither is perfect!