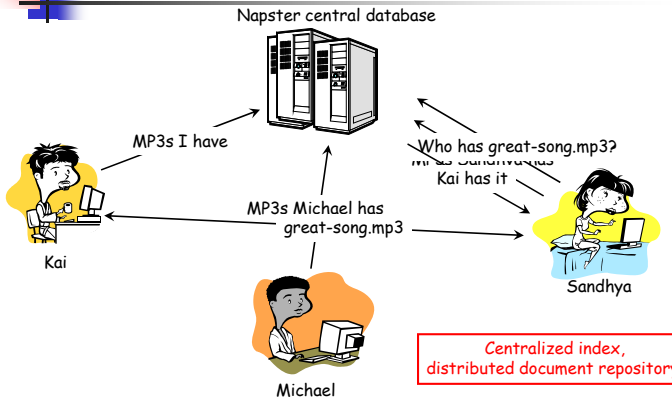
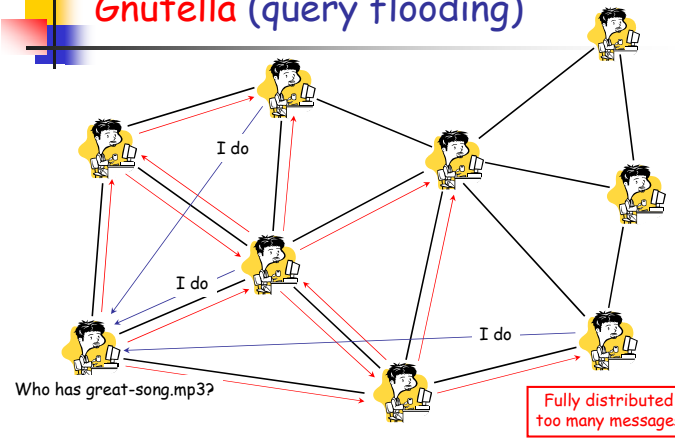
Peer-to-Peer Networks

Kai Shen

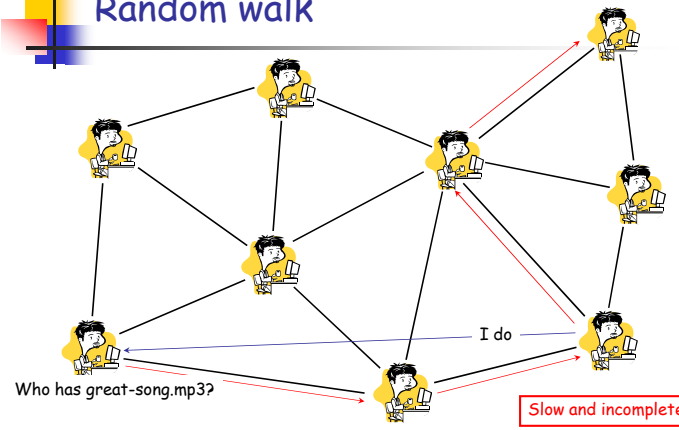
Distributed Search I: Napster (central index)



Distributed Search II: Gnutella (query flooding)

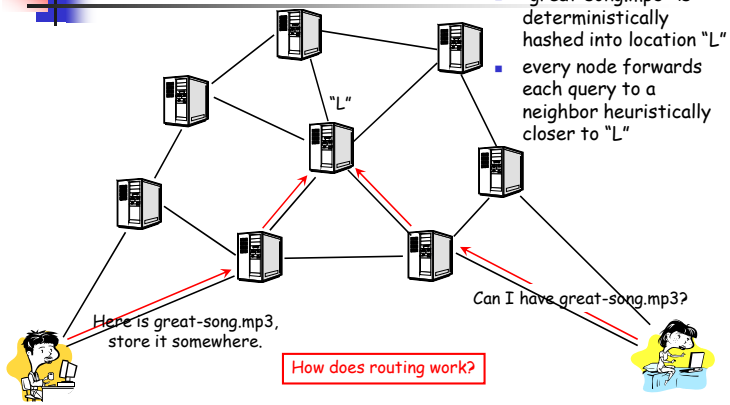


Distributed Search III: Random walk



Distributed Search IV: Object-addressable network

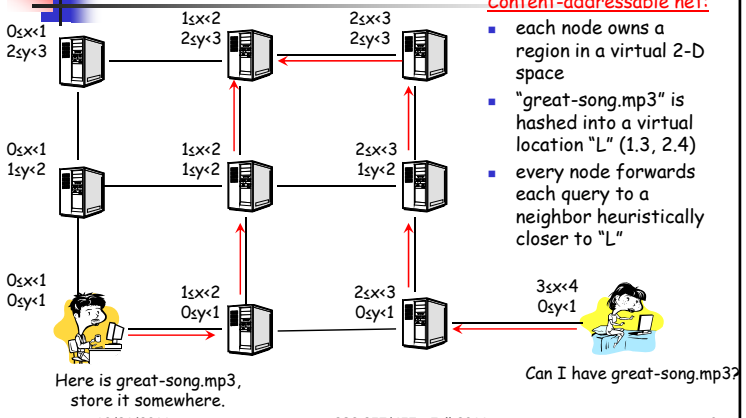
- "great-song.mp3" is deterministically hashed into location "L"
- every node forwards each query to a neighbor heuristically closer to "L"



Scalable Object-addressable Net

Content-addressable net:

- each node owns a region in a virtual 2-D space
- "great-song.mp3" is hashed into a virtual location "L" (1.3, 2.4)
- every node forwards each query to a neighbor heuristically closer to "L"



Analysis

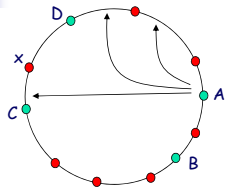
Content-addressable network [Ratnasamy et al. 2001]:

- each node owns a region in a virtual 2-D space
- each object is hashed into a virtual location "L"
- every node forwards each query to a neighbor heuristically closer to "L"

Questions:

- Space consumption? Lookup cost for a network of N nodes?
  - 4 links per node; $O(N^{1/2})$
- Can the lookup cost be improved?
- Can we take into account the actual link latency?

Chord [Stoica et al. 2001]



- hash code of a data object
- hash code of a network node

- Each object is mapped to the first node in the clockwise direction on the ring.
- Every node maintains links to
  - the node half-way in the circular ID space
  - the node a quarter-way across the circular ID space
  - ...
- Space consumption? Lookup cost for a network of N nodes?

Distributed Hashtables

- Content-addressable network (CAN) [Ratnasamy et al. 2001] and Chord [Stoica et al. 2001] are also called scalable distributed hash tables (DHTs)
- There are other scalable DHT protocols.

Geographic Location-based Routing

- Bunch of wireless sensors (not mobile)
- Each is identified by its geographic location (longitude, latitude)
- Routing – a packet is routed to a neighbor with shortest geographic distance to the destination
- Pro and Con?

Peer-to-peer Networks

- Peer-to-peer networks: distributed systems of no hierarchy – each node is a peer of any other in terms of equal functionality
  - DNS? Napster? Gnutella? Random walk? DHTs (CAN and Chord)? GLR?
- Fundamental advantage of p2p networks
  - better scalability?
    - no performance bottleneck
  - better robustness?
    - no single point of failures; more tolerant to intentional attacks

More on Scalability & Robustness

- Scalability: able to support large number of nodes
  - cost of each operation is below linear-scaling – goes up slowly when the system size increases: $O(n)$ is terrible, $O(n^{1/2})$ is OK, $O(\log n)$ is good.
  - space requirement at each node is below linear-scaling.
  - Gnutella? Random walk? CAN? Chord?
- Robustness: complexity of repair mechanism is a very important issue
  - Gnutella? Random walk? CAN? Chord?

Where are we now?

- Several approaches for searching large, distributed systems:
  - central index
  - query flooding
  - random walk
  - hash + heuristics-based routing (DHT)
- Concept of peer-to-peer networks.
- Additional p2p services, built on distributed search
  - DNS
  - p2p file sharing
  - p2p keyword search

P2P DNS

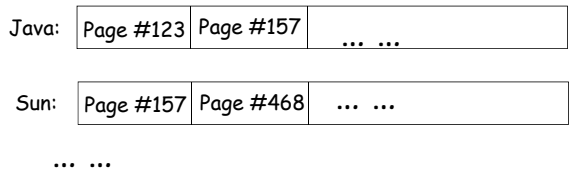
- DNS running on scalable DHT like Chord or CAN
  - Scalable and robust
  - Accesses aren't always local
- A hybrid approach
  - Retain local DNS server and caching
  - Support search of destination authoritative server using scalable DHT ⇨ no need for root servers

P2P Sharing of Large Files – BitTorrent

- Downloading large files is slow and unreliable
  - Chop a large file into pieces and try to download each piece
- How to find who has each piece of the file?
  - Distributed search: Gnutella, Chord, CAN, …
- Downloading approach:
  - Rarest-first ⇨ high chance of completion
  - Multiple TCP connections for downloading

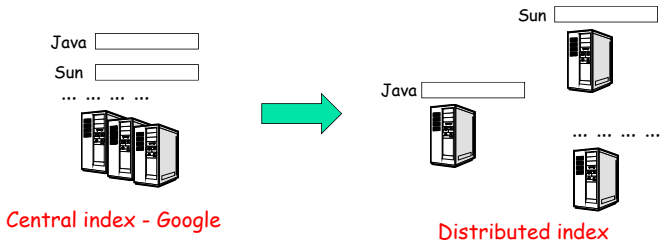
Keyword Search

- 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
Solution 1: Split based on keywords

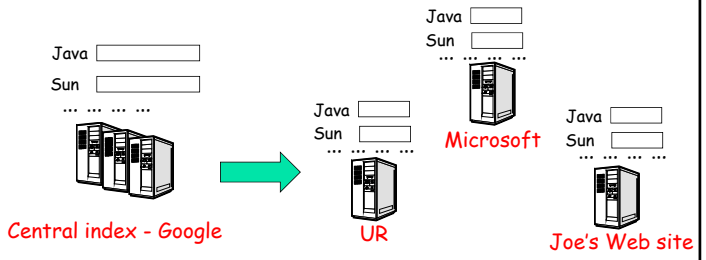
Split the index database to many pieces based on keywords and distribute them to many nodes in the network.



Weakness: transferring large index for multiple-word queries.

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.



Weakness: too many sites to visit for each query.