Simulation of the Nomification of Independent Chord Rings

Andrew Rosen Brendan Benshoof

Outline

File Lookup

- Most important function of a decentralized Peer-to-Peer system.
- Need to discover which node in the network has a certain file.
- Need to do it efficiently.

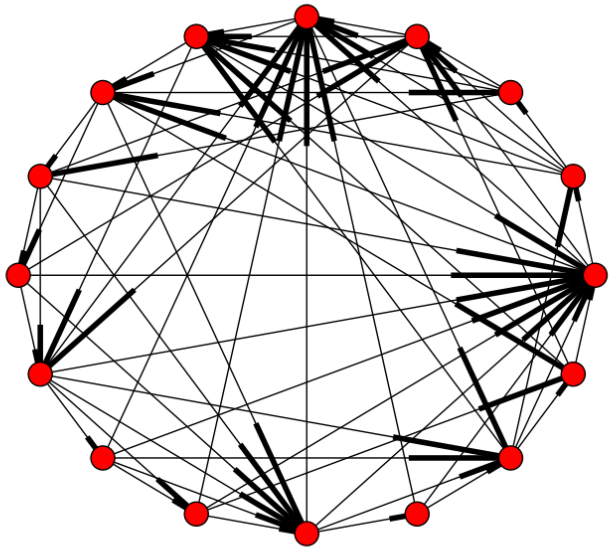
Hashing Protocols

- One of the best solutions for this task.
- Map nodes and files identifiers to keys.
- Nodes are responsible for files with keys that match some criteria,.
- Imposes an structure on the network.

What is Chord

- Arranges a network into a ring.
- Maximum 2^m nodes in the network.
- Nodes and filenames are hashed to create an m-bit key.

How Are Files Stored



- Node responsible for for key κ is the *sucessor*(κ).
- *sucessor*(κ) is the node with key κ or first following key.

How Are Files Retrieved

- To find a file with key κ, we find *sucessor*(κ).
- We could just ask our successor, who would do the same.
- We use a *finger-table* make this quicker
  - *m* entries in the table.
  - Stores location of *successor*$(n + 2^{i-1})$ *mod* 2^m.
  - If κ is between us and an entry, skip to that entry.
- Once we know *sucessor*(κ), we can directly connect.
- This reduces the lookup time to $O(\log n)$

How Is Churn Handled

- Join a network by asking some node in it to find your successor.
- Periodically update the entries in the finger table.
- Periodically find your successor's predecessor.

What happens:

Goals

- If there is more than one initial ring?
- If our range of sight is limited?
- ANOTHER THING
- We ignore file lookup; identical to node lookup.

Agents

- Nodes for nodes, Links for links.
- Seekers.
- Updates.

Setup

- Generate nodes in random locations (ring visualization optional)
- Specified number of nodes for independent rings.
- Set the hash size.

Stuff