Chord

Lei Yan

Slides adopted from Rishabh Iyer

Context

- P2P:
  - Peer-to-peer (P2P) computing or networking is a distributed application architecture that **partitions tasks or workloads between peers.** Peers are **equally privileged, equipotent participants** in the application.

Chord

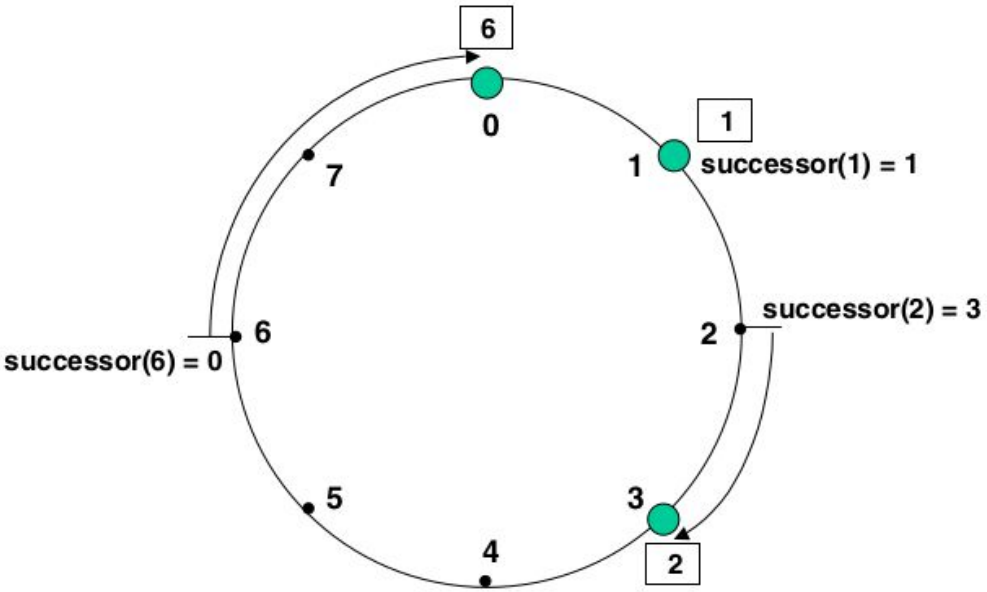
- Load balance
- Scalable key location
- Correct and fast lookup upon nodes joining/leaving

Chord

- Consistent Hashing
- Key location
- Node joins, stabilization
- Fault tolerance

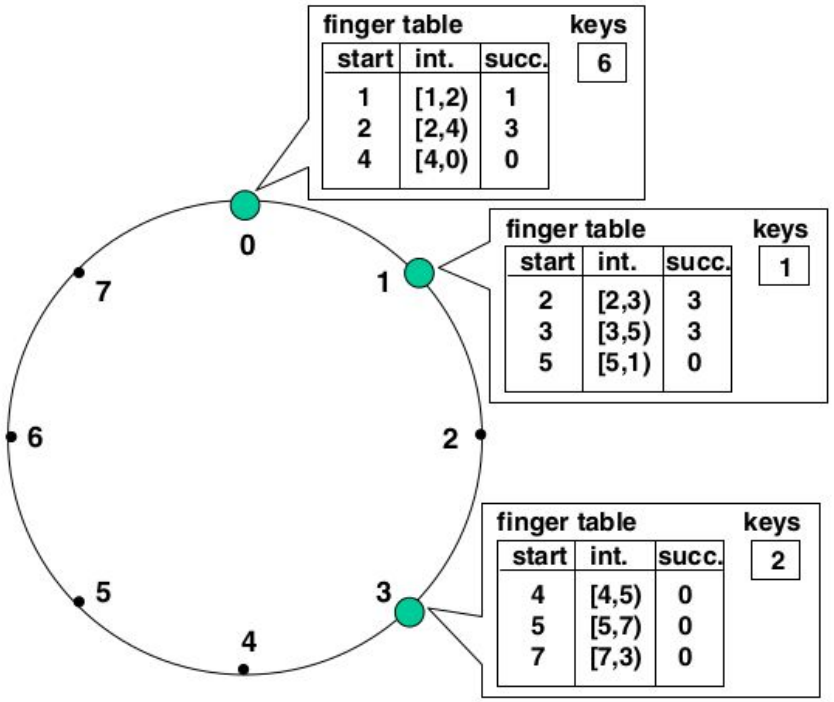
Consistent Hashing

- What's different from regular hashing?
  - What are the invariants that it maintains?



Scalable Key Location

- Minimum requirements for key location?
- Why is lookup latency O(log(n))? What state must be maintained for this to be the case?



Node joins

- Three steps:
  - Initializing new node
  - Updating existing nodes' finger tables
  - Transferring ownership of keys
    - How would you design chord-based storage system if BW was the critical resource?

Stabilization Protocol

- Is updating finger tables really needed for correct key lookups?
- Have finger tables to be up-to-date to guarantee fast lookups?
- Stabilization:
  - Runs periodically on each node
  - Update successor quickly
  - Update finger table entries gradually

Fault Tolerance

- Failure detection
- Maintaining routing invariants
- Avoiding doing key lookup through failed node
- Avoiding data loss

Any thoughts on Chord? Any POCS principles you find?