Chord

Lei Yan

Slides adopted from Rishabh Iyer

26-11-20

Context

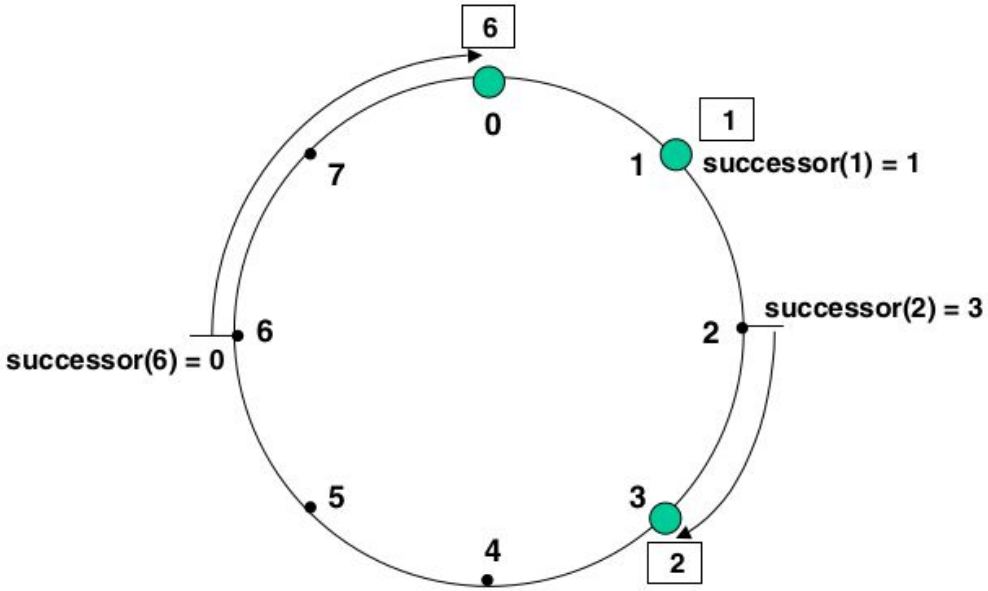
- P2P:
  - Peer-to-peer (P2P) computing or networking is a distributed application architecture that **partitions tasks or workloads between peers.** Peers are **equally privileged**, equipotent participants in the application.

Chord

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

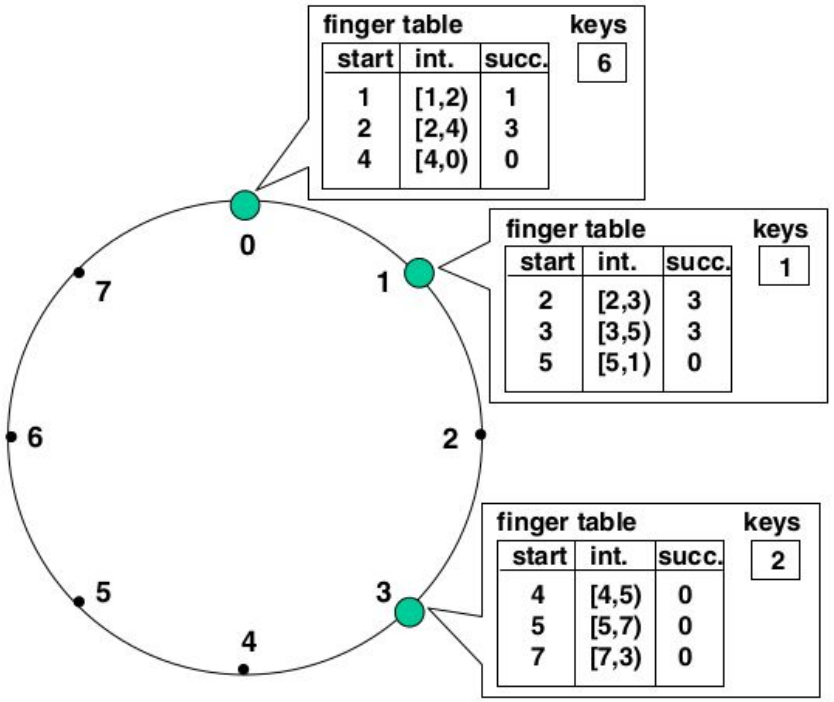
Consistent Hashing

- 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?
    - What's the problem of doing this as the last step? How to solve this?

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:
  - Find successor immediately
  - Find predecessor, finger table entries later
  - Update everything periodically

Fault Tolerance

- Failure detection
- Maintaining routing invariants
- Avoiding data loss

Any thoughts on Chord? Any POCS principles you find?

Design Project

- Design a distributed, incrementally-scalable, fault-tolerant, highly available, eventually consistent key-value store