5. ZooKeeper

The ZooKeeper Service

- A Distributed Coordination Service for Distributed Applications
- Exposes a simple file-system like semantics with a hierarchical name space
  - The concept of a znode
  - But data is kept in memory for good performance
  - Accesses are sequentially consistent with linearized updates
- Internally, a replicated service based on the Paxos protocol
- An open source project
  - Lead by Yahoo!
  - Used by many other companies
  - Currently, the open source replication engine of choice

- Largely mimics Google’s proprietary chubby service
- Recent competitors include etcd and consul
ZooKeeper’s Architecture

- A ZooKeeper service is replicated over a collection of servers called an Ensemble.
- A client interacts directly with a single server at a time over a TCP connection. If the server fails, the client initiates a new TCP connection with another server.

ZooKeeper’s API

- A znode can both store data (typically small – up to a few KB) and have children.
- The data has a version number and an ACL:
  - The version number is incremented on each update.
  - The version number is returned with each read.
- A node can be persistent or ephemeral:
  - The latter exists only during the session that created them.
- Operations include:
  - create, delete, exists, get data, set data, get children, sync.
- Can set watchers.
ZooKeeper’s API Guarantees

- Sequential consistency
- Atomicity – operations are either fully executed or not at all
- Single system image – no matter which server a client accesses
- Reliability/persistence
- Timeliness – reads cannot return information that is older than a certain bound

- It is possible to find many recipes for various services on the Web

Example: Leader Election with ZooKeeper

- Let ELECTION be a path of choice for a znode created by the application for this instance of leader election
- To volunteer to be a leader:
  - myseq=Create(ELECTION/n_, SEQUENCE | EPHEMERAL);
  - tryLeader()

- Upon receiving a notification of znode deletion:
  - tryLeader()

- tryLeader()
  - C=getChildren(ELECTION);
  - If myseq is the smallest znode in C, then execute leader procedure;
  - Else
    - Let j be the lowest index of a znode in C; set watcher on ELECTION/n_j;
    - Endif