Consensus Algorithms and Paxos

The Consensus Problem

- Simplified definition: The problem of having multiple instances of a system come to agreement on some piece of data in a fault tolerant manner
- Fault tolerance is where the difficulty comes from
- Scenarios
  - Deciding whether to commit a transaction to a database
  - Keeping state consistent between machines by forming consensus on all changes to that state, known as *state machine replication*.

Paxos

- Solves consensus for nodes which may fail but not arbitrarily misbehave
- Works so long as a majority of nodes are working
- Developed by Leslie Lamport
- Original paper submitted in 1990 - “The Part-Time Parliament” entirely in the form of a metaphor to a fictional ancient Greek parliament on the island of Paxos
- Finally accepted by the ACM in 1998
- Eventually got tired of having to explain what things meant and published a second paper in 2001, “Paxos Made Simple”, which boringly dropped the metaphor but is much easier to understand

How does Paxos work?

- Let there be a set of proposers which generate possible values, and a set of acceptors, which receive and may accept proposals. A proposal is chosen if a majority of acceptors accept it.
- For consistency, we need to ensure that only one value can be chosen by a majority of acceptors.
- To achieve this, Paxos uniquely numbers all proposals, and bars all acceptors from accepting any proposal with a higher number than one it already accepted, and a different value from the lowest one it accepted.
- Additionally, to ensure that no earlier proposal that could ever be accepted by a majority of acceptors exists, before making a proposal, a proposer extracts a promise to reject all earlier proposal numbers from a majority of acceptors.

Paxos All Together: The Full Four Step Process

- Let there be a set of learners, which have an interest in the value.
- Proposer B -> All Acceptors: *Prepare* message, containing a proposal number N.
- All Acceptors -> Proposer B: *Promise* message, confirming that no proposal with number less than N will ever be accepted.
- Proposer B -> All Acceptors: *Accept request* message, containing the proposal number and proposed value.
- All Acceptors -> Learners: *Accept* message, containing the proposal number and proposed value.

Paxos Variants

- Making it so the system can choose multiple values in an ordered series is a necessary step to build most useful systems.
- Multi-Paxos: You can speed things up by using the full four-step process to elect a leader, and then using assumptions about being the sole leader to permit a two-step process.
- Byzantine Paxos: A variant which can tolerate a minority of nodes not merely going offline but generating incorrect or malicious messages.

raft

- A Paxos derived algorithm designed to be simpler to deploy and less error-prone to implement in practice, implemented in Go

- https://github.com/hashicorp/raft

Other Consensus Algorithms

- CAP Theorem: Consensus, Availability, and Partition-Tolerance: Pick two
- Paxos and variants choose CP; in event of a partition in which a majority of nodes can no longer communicate, all nodes become unable to choose values
- There are situations where availability is more important than consistency, so it is preferred to keep on working and resolve inconsistencies later, and other consensus algorithms exist to permit this