Leader Election

Assume subscription based services such as Amazon Prime.

All the subscription related data will be stored in the Database. The payments will be carried out using third party services such as Paypal, Stripe.

The third party services should not be given access to our database. Thus a server sits between the database and the third party service. Inorder to make our system more redundant two or more servers sits between the DB and the third party service. This is where the problem arises. If server-1 and server-2 both deducts money from the user then it becomes a problem. This is where leader election comes into play.

Leader Election

Leader Election guarantees that all the nodes in the cluster knows which one is the leader amongst them at any given time and can elect a new leader if the leader dies for whatever reason. The leader is responsible for all primary operations of that service.

This leader election is done using the distributed consensus algorithm. Two popular consensus algorithms are Paxos (more complicated) and Raft. Instead of using the algorithm directly services such as Etcd (pronounced as Et-c-d), Zookeeper are used, which under the hood uses a consensus algorithm.

Eg: Raft

http://thesecretlivesofdata.com/raft/

- Election timeout (150ms - 300ms)
- Follower, Candidate, Leader (default)
- heart beat

↳ Log Replication . Leader Election

↳ Split votes