More

Rsms

Where we are

- State Machines & State Machine Replication

- Raft
  ↳ Overall Structure
    * Leader Synchronizes Log With Replicas
      Decides When Commands Are Committed
    * An Entry At Index I once Committed Is At Log Index I At All Future Leaders
    * Log Completeness.

⇒ Protocol

(i) Command Replication

(ii) Failure Detection

(iii) Leader Election / Failure Handling

(iv) Reconfiguration

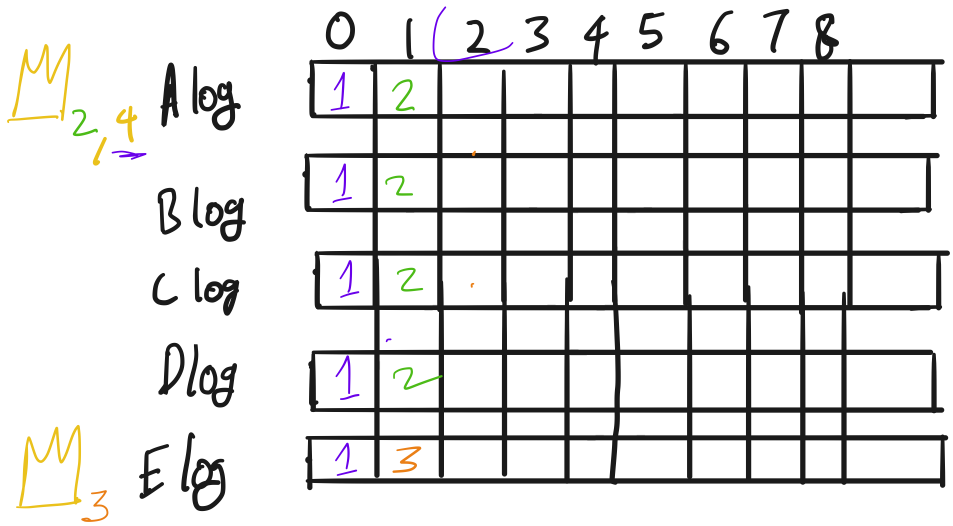
Leader Election

- Term
  - Requirements To Vote For A Candidate
  - Requirement For A Candidate → Leader

When Is An Entry Committed?

- Command Replication?

- Interactions with Leader Election



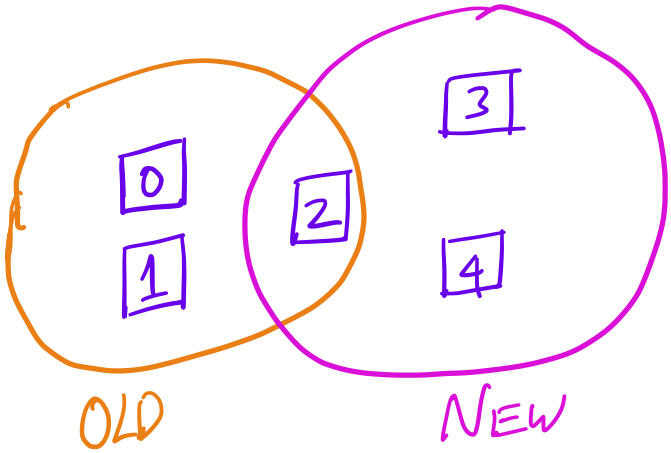
5 (LastLogTerm > Node.LastLogTerm) OR

(LastLogTerm == Node.LastLogTerm

AND LastLogIndex >= Node.LastLogIndex)

Reconfiguration

Problem: Want to Add &/or Remove Nodes



Need to make sure no commands are lost

Avoid "SPLIT BRAIN"

How?

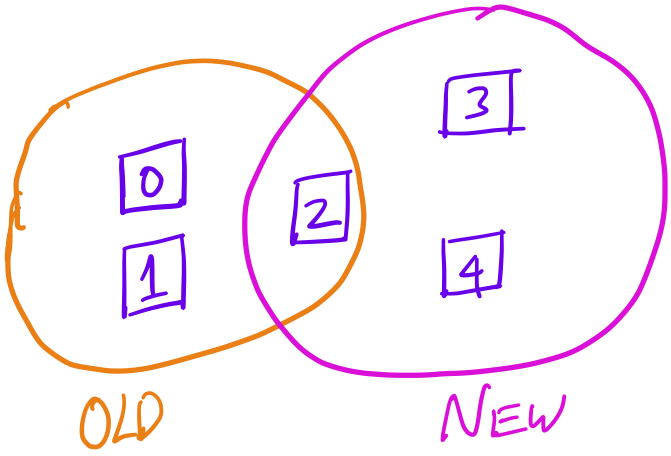
- Store configuration in log
  - ↑ Set of active nodes

- Transition

$C_{OLD} \rightarrow C_{OLD+NEW} \rightarrow C_{NEW}$

↓

Majority in OLD + Majority in New



MULTI PAXOS

- CLIENTS / REPLICAS / LEADERS / ACCEPTORS

SIGH!

- Really Not All That Different From Raft As We Will See

Replica Leader Acceptor

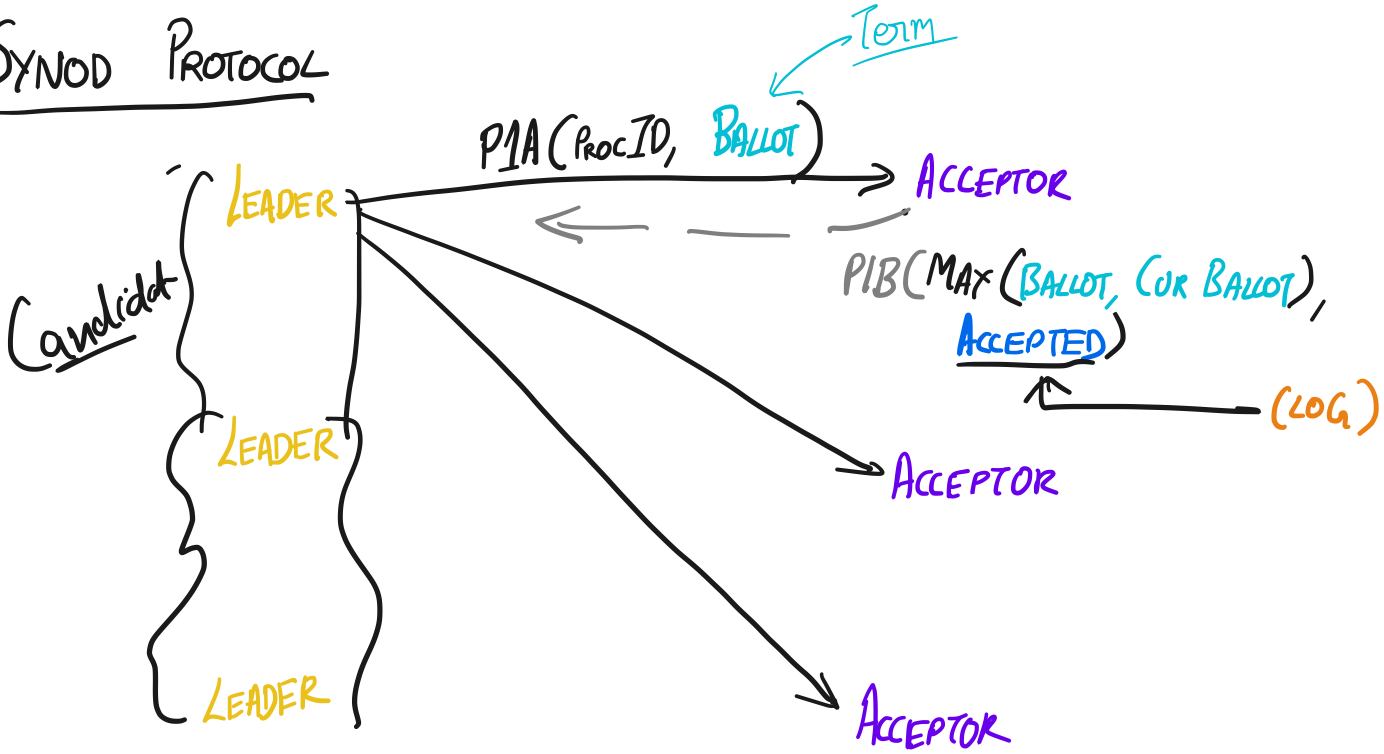
Replica Leader Acceptor

Replica Leader Acceptor

$f = 1$

[Need
$\geq 2f+1$ acceptors
$\geq f+1$ replicas, leaders
]

Synod Protocol



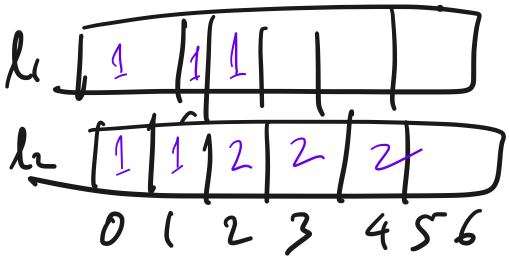
$\langle P1A, P1B \rangle \equiv$ Leader Election/View Change

DIFFERENCES

- No log completeness check.

- IMPLICATION: MUST ENSURE CHOSEN LEADER HAS COMMITTED ENTRIES.

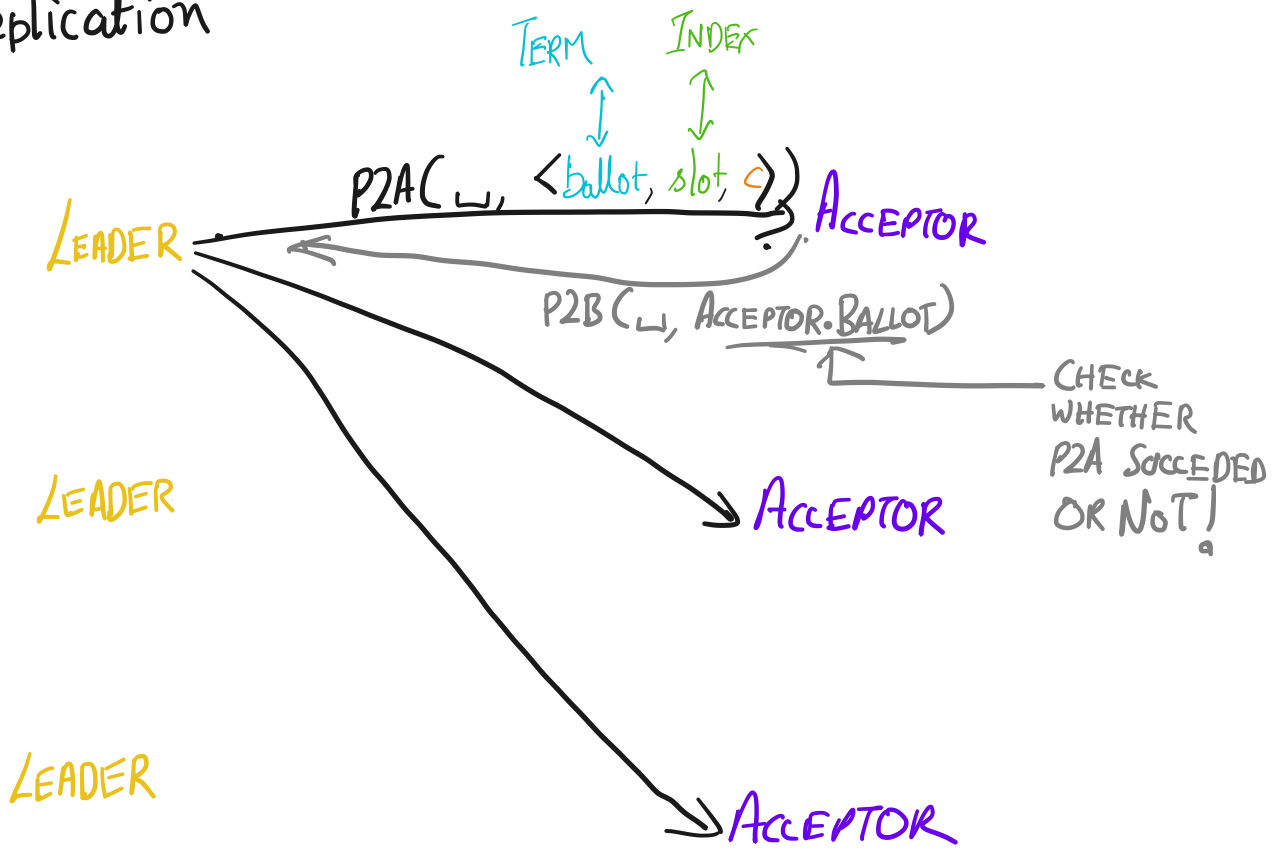
HOW?

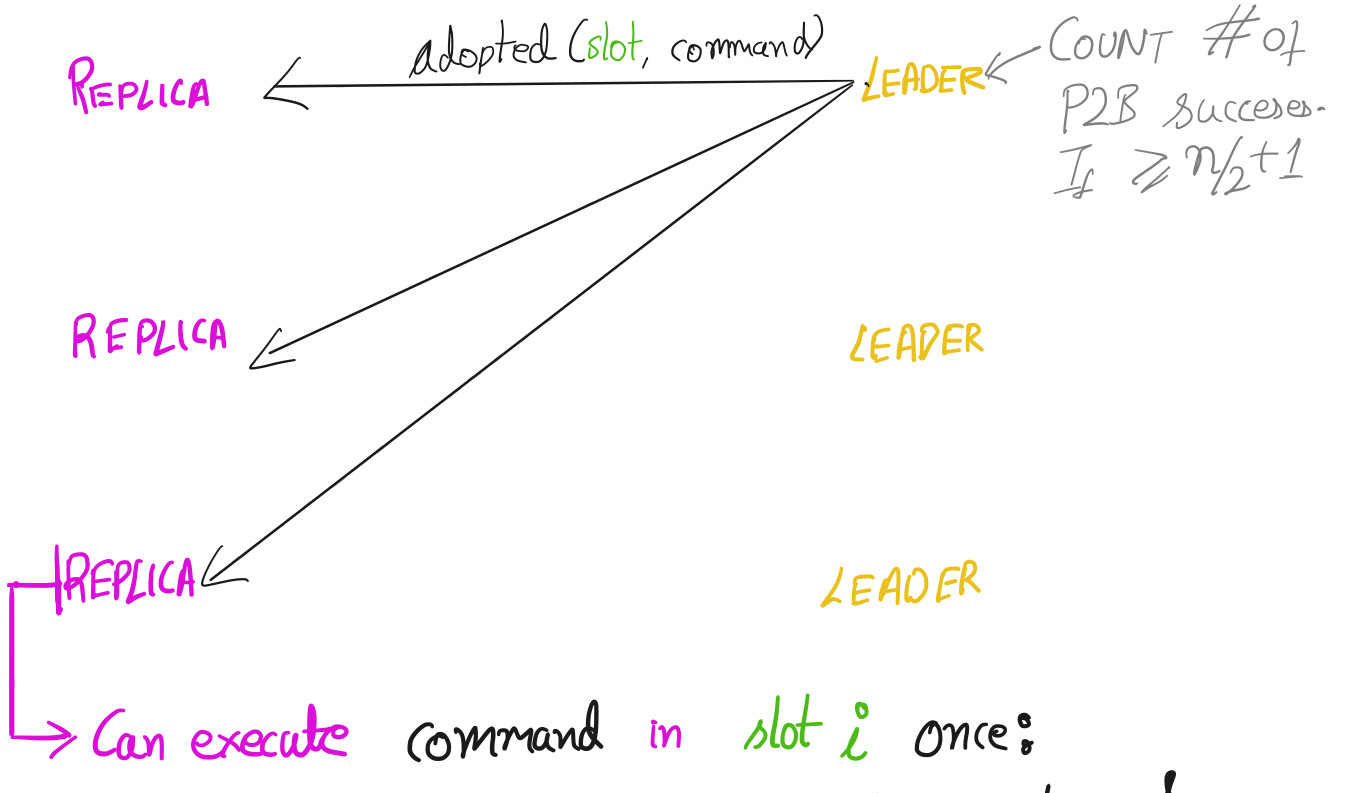


- WHEN IS P1A TRIGGERED & LEADER STABILITY?

SCOUTS

Phase 2: Replication





Can execute command in slot i once:

* Command in $i-1$ adopted & executed &
* Command in slot i ADOPTED

ALTERNATE VIEW

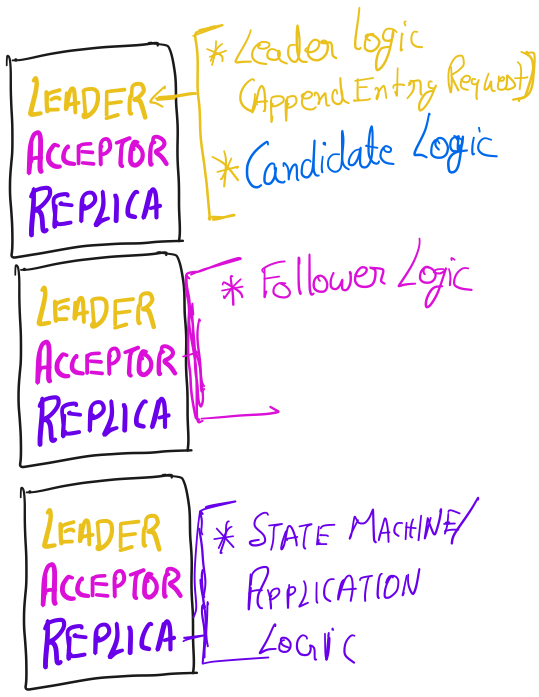
MPAXOS

REPLICA LEADER ACCEPTOR

REPLICA LEADER ACCEPTOR

REPLICA LEADER ACCEPTOR

RAFT



SOME THINGS LEFT UNDERSPECIFIED

* FAILURE DETECTION LOGIC

MY CONTENTION: RAFT IS DESIGNED FOR A PARTICULAR USE

MULTI PAXOS PROVIDES A DESIGN PATTERN THAT CAN BE APPLIED TO DIFFERENT DEPLOYMENTS

EXAMPLES (MORE IN THE PAPER)

DISK PAXOS: ACCEPTORS ARE DISKS

* WHY?

* What changes?

Acceptor Interface: Read Block, Write Block

Not Active

Mencius/Epaxos: Consensus In The Wide Area

Consensus In Practice: Lock Services.