Last week: RSMs. A framework for building fault tolerant distributed applications.

Deterministic State Machine → Replicated For Fault Tolerance

Raft is one such protocol.

Assumptions:
- Partial Synchrony

Safety Guarantees:
- Agreement
- Validity

Useful to Understand Requirements
1 alice invokes command \textit{c}

2 System responds to alice

\textit{c}'s effects are guaranteed to persist despite permissible failures

\textit{Commit Point in Raft}

- After command \textit{c} is committed at slot \textit{i}, slot \textit{i} always contains \textit{c}

- All slots 0..\textit{i}-1 contain committed commands

\textit{Log Completeness}
All of the portions of the protocol are designed to maintain these properties:
- Replication (AE, AER)
- Leader Election (Rv0, Rv0,Rv0)
- Reconfiguration

**Core Approach**: Quorum Intersection

Any two changes to Raft state (term, log, configuration) must have been signed off by a common node.

Common building block for all consensus protocols

Parameter: \( f \): # of nodes that can fail.

Min quorum size?

\[ n - f \quad n - f + t \quad \text{for } t = 3, 20 \]
Invariants:
- Term: Z : Ø or 1 leader. (Leader election ensures this)
- (term, index) ←→ command on 1
- In steady state, leader's log is authoritative.

Replication/Log Synchronization

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>Term</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

0 1 2 3 4 5 6 7 8

Append Entry Contents:
- Term: 3
- Command(0) = C
- Last log index = Why?!
- Last log term = 3
- CI

Processing at B
Processing responses at A (👑)

- When is a command committed?
  
  \[ n = 2f + 1? \]

- When can A execute the command?

- When can D execute the command?
Addendum: When is a command committed?
Conflict

- Stable leadership is necessary for progress. Why?

Avoid leader election when unnecessary

- Want to detect that a leader has failed soon. Why?

Solution

- Heartbeats: no-op
  append entry requests

- Handling at Leader

- Handling at follower.

Election Timeout

- Relation to Heartbeats?
Leader Election

Requirement 1: New leader's log should contain all committed log entries.

Slight refinement of:

After command \( c \) is committed at slot \( i \), slot \( i \) always contains \( c \).

Requirement 2: At most one leader at a time.

Protocol assumes there is no preference for what node becomes leader.
What does a node (B) do on receiving RV?

- **Agree to Leader** at most
- **Update term** (if received term is higher)
- **Consequence**
  - Check if B has already voted in term
  - Check who has more up-to-date log
When a candidate → leader?

When does leader in term(t-1) → FOLLOWER (in term t)?

Re-configuration
- Goal: Change the set of nodes participating

- Why hand? Need to pick a slightly complex setting

<table>
<thead>
<tr>
<th>Old Term=2</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</td>
<td>D</td>
<td>B</td>
<td>C</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>t2</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td>t2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Must ensure that D or C do not get elected. Why?

But A cannot continue as leader either. Why?

Things we need to fix:

1) Synchronize logs so that a quorum of nodes in the new config contain committed entries.
Observation: AE/AER synchronizes logs for a quorum of nodes so they match the leaders log.

Idea: 1. Run leader election to find a node that

Joint \( \{ \)

\( a \) Can commit to nodes in the new configuration

\( b \) Has all committed entries

\( \} \)

\( \text{Old Term=2} \)

\( \text{A} \) \( \text{B} \) \( \text{C} \) \( \text{M} \)

\( \text{New} \)

\( \text{D} \) \( \text{B} \) \( \text{C} \)

2. Have the new leader commit an entry
Notes on practice

- Leader election

- Reconfiguration