Administrivia

- If you need access to lecture videos, please email cs244b-staff
  - Subject: downloadable lecture videos
  - I need the ability to download lecture videos and I promise to delete all downloaded videos at the end of the quarter.
- Please re-do poll from last class here (or class poll link on class web page)
  - Contrary to zoom documentation I was unable to get results after last lecture
- Jim office hours announcement

Plan for next three lectures

- Today: PBFT – classic BFT replication algorithm
  - First practical algorithm, still quite relevant (e.g., hyperledger)
- Wednesday: Randomized BFT algorithms
  - Very different BFT techniques with different tools, trade-offs
- Monday 5/4: Other topics in BFT, HotStuff
  - Advances since 1999 (when PBFT published)
  - Partial synchrony
- Then we switch gears and talk about higher-level systems

Voting safety in fail-stop model

- Suppose you have N nodes with fail-stop behavior
- Pick a quorum size T > N/2
- If T nodes (a quorum) all vote for a value, output that value
  - E.g., Quorum A unanimously votes for 9, okay to output 9
  - Nodes cannot change their vote
  - Any two quorums intersect ⇒ agreement
- Problem: stuck states
  - Failure could mean not everyone learns of unanimous quorum
  - Split vote could make unanimous quorum impossible

What voting gives us

- You might get system-wide agreement or you might get stuck
  - Can’t vote directly on consensus question (what RSM op to apply)
- How do you know you agreed?
  - If more than f = N − T nodes fail, will always get stuck
  - If f + 1 nodes see T votes, even if f fail one can spread word
Byzantine agreement

Quorum A

\[
\begin{array}{cccc}
V_0 & \cdots & V_{N-T} & \cdots & V_{T-1} & \cdots & V_{N-1}
\end{array}
\]

Quorum B

- What if nodes may experience Byzantine failure?
  - Byzantine nodes can illegally change their votes
    - In fail-stop case, safety required any two quorums to share a node
    - Now, any two quorums to share a non-faulty node
  - Safety requires: # failures \( \leq f_5 = 2T - N - 1 \)
  - Liveness requires: # failures \( \leq f_1 = N - T \)
    - At least one entirely non-faulty quorum exists
  - Typically set \( N = 3f + 1 \) and \( T = 2f + 1 \) so \( f_5 = f_1 = f \)

When has a vote succeeded?

- If \( f_5 + 1 = 2T - N \) nodes malicious, system loses safety
- Suppose \( f_5 + 1 \) nodes all claim to have seen \( T \) votes for \( a \)
  - Can assume system is \( a \)-valent with no loss of safety
  - In fact, \( f_5 + 1 \) signed msgs = proof of system state (or unsafety)
- Now say \( f_1 + f_5 + 1 = T \) nodes all make same assertion
  - If \( f_1 \) fail, system loses liveness (0 correct nodes in whole system)
  - If \( f_1 \) fail, \( \geq f_5 + 1 \) remaining nodes can notify rest
  - So either catastrophe or all non-faulty nodes will eventually hear it

\[ a \text{-valent} \quad a \text{ agreed} \]
When has a vote succeeded?

Quorum A

\[ V_0 \cdots V_{N-T} \cdots V_{T-1} \cdots V_{N-1} \]

Quorum B

\[ V_0 \cdots V_{N-T} \cdots V_{T-1} \cdots V_{N-1} \]

- If \( f_S + 1 = 2T - N \) nodes malicious, system loses safety
- Suppose \( f_S + 1 \) nodes all claim to have seen \( T \) votes for \( a \)
  - Can assume system is \( a \)-valent with no loss of safety
  - In fact, \( f_S + 1 \) signed msgs = proof of system state (or unsafety)
- Now say \( f_L + f_S + 1 = T \) nodes all make same assertion
  - If \( > f_L \) fail, system loses liveness (0 correct nodes in whole system)
  - If \( \leq f_L \) fail, \( \geq f_S + 1 \) remaining nodes can notify rest
  - So either catastrophe or all non-faulty nodes will eventually hear it