PBFT: A Byzantine Renaissance

Practical Byzantine Fault Tolerance
(Castro, Liskov 1999-2000)

• First practical protocol for asynchronous BFT replication

• Like Paxos, PBFT is safe all the time, and live during periods of synchrony
The general idea

- One primary, 3f replicas
- Execution proceeds as a sequence of **views**
  - A view is a configuration with a well-defined primary
- Client sends signed commands to primary of current view
- Primary assigns sequence number to client’s command
- Primary is responsible for the command eventually being decided
Certificates

Protocol steps are justified by certificates

- Sets (quorums) of signed messages from distinct replicas proving that a property holds

Certificates are of size at least $2f + 1$

- Any two quorums intersect in at least one correct replica (for safety)
- There is always a quorum of correct replicas (for liveness)
CLIENT issues request

<REQUEST, o, t, c> \sigma_c

Primary

Replica 1

Replica 2

Replica 3
Primary sends $\langle\langle\text{PRE-PREPARE}, v, n, d>_{\sigma_p}, m\rangle$ to all replicas
Replica $k$ sends $\langle PREPARE, v, n, d, k \rangle_{\sigma_k}$ to all replicas

Pre-prepare phase
**Prepare Certificate**

- P-Certificates ensure consistent order of requests within views.

- A replica produces a P-Certificate \((m, v, n)\) iff its log holds:
  - the request \(m\)
  - A PRE-PREPARE for \(m\) in view \(v\) with sequence number \(n\)
  - \(2f\) PREPARE from distinct backups that match the PRE-PREPARE

- A P-Certificate \((m, v, n)\) means that a quorum agrees to assign \(m\) to sequence number \(n\) in view \(v\)
  - **No** two non-faulty replicas with P-Certificate \((m, v, n)\) and P-Certificate \((m', v, n)\)
P-Certificates are not enough

- A P-Certificate proves that a quorum of $2f + 1$ replicas has agreed to assign $m$ to sequence number $n$ in view $v$.
- Yet that assignment could be modified if a view change happens (the primary changes).
  - The new primary may not be convinced to assign $m$ to $n$ in the new view $v'$. 
P-Certificates are not enough

- Yet that assignment could be modified if a **view change** happens (the primary changes)
  - The new primary may not be convinced to assign \( m \) to \( n \) in the new view \( v' \)
  - \( 2f + 1 \) prepares means at least \( f + 1 \) correct replicas received a pre-prepare for \((m,v,n)\)
After collecting a P-Certificate, replica $k$ sends $<\text{COMMIT}, v, n, d, k>_{\sigma_k}$ to all replicas.

<table>
<thead>
<tr>
<th></th>
<th>Pre-prepare phase</th>
<th>Prepare phase</th>
<th>Commit phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replica 1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replica 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replica 3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Commit Certificate**

- C-Certificates ensure consistent order of requests across views
  - **Cannot miss** a P-Certificate during view change

- A replica has a C-Certificate\((m,v,n)\) iff:
  - it had a P-Certificate\((m,v,n)\)
  - its log contains \(2f + 1\) matching COMMIT messages from distinct replicas (including itself)

- A replica executes a request when:
  - it gets a C-Certificate for it
  - it has executed all requests with smaller sequence numbers
After executing a request, replica $k$ replies to the client with $\langle$REPLY, $v$, $t$, $c$, $k$, $r$$\rangle$.
TO ARMS, REPLICA!!

- A disgruntled replica mutinies:
  - Stops accepting messages (except for VIEW-CHANGE and NEW-VIEW messages)
  - Sends $\langle \text{VIEW-CHANGE}, v+1, P \rangle_{\sigma_k}$
  - $P$ contains all P-Certificates known to replica $k$
- A replica joins mutiny after seeing $f + 1$ distinct VIEW-CHANGE messages
- Mutiny succeeds if the new primary collects a new-view certificate $\mathcal{V}$, indicating support from $2f + 1$ distinct replicas (including itself)