Three Phase Commit

10/24/2016
Requirements

• Agreement:
  – No two processes decide on different values

• Validity:
  – If any process starts with “abort”, then “abort” is the only possible decision value.
  – If all processes start with “commit” and there are no failures, then “commit” is the only possible decision value.

• Termination:
  – If at any point, all failures are recovered, and there are no failures for sufficiently long, then all processes will eventually decide.
Phase 1: Voting

1. Coordinator sends “vote request” message to all participants.
Phase 1: Voting

1. Coordinator sends “vote request” message to all participants.
2. Participant sends vote, “commit” or “abort” to coordinator.
   If vote is “abort”, participant decides abort.
   Otherwise its state is uncertain.
Phase 2: Pre-Commit

1. If all participants voted "commit", coordinator sends "pre-commit" message to all participants.

   Otherwise, coordinator sends "abort" message.
Phase 2: Pre-Commit

2. If participant receives “abort” from coordinator, it decides abort.
   If participant receives “pre-commit”, it sends “ack” to coordinator.
   Its state is **pre-committed**. It has not yet decided.
Phase 3: Commit

1. Once coordinator has received “ack” from all participants, it sends “commit” to all of them.
2. When participant receives “commit” message, it decides “commit”.

Diagram:
- Coordinator (C)
- Participants (P1, P2, P3, P4)
- Arrows showing message flow: C → P1, C → P2, C → P3, C → P4
Correctness of Three-Phase Commit

• If no failures, algorithm satisfies agreement, validity, and termination properties.

• How are failures handled?
If coordinator fails before a participant receives a vote request messages, then the participant decides “abort”
Failures in Voting Phase

If participant fails before sending vote, coordinator decides “abort” and sends decision to participants.
If coordinator fails before sending all pre-commit messages.... some participants may be in uncertain phase
Need to execute **election protocol**.
Failure in Pre-Commit Phase

• If participant fails before sending “ack”, the coordinator knows all participants have voted “commit”.
• Sends commit to all participants.
Phase 3: Commit

- If coordinator fails before sending all commit messages....
  - Some participants are in pre-commit state:
    - Participant in pre-comment doesn’t know whether other processes are in pre-commit, commit, or uncertain states.
  - Need to execute election protocol.
Election Protocol

• Participants elect a new coordinator.

• New coordinator collects state information from all participants, implements a termination rule to make decision, and sends decision to all participants.

• Possible participant states:
  – Aborted
  – Uncertain
  – Pre-committed
  – Committed
## Possible Participant States

<table>
<thead>
<tr>
<th></th>
<th>Aborted</th>
<th>Uncertain</th>
<th>Pre-commit</th>
<th>Committed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aborted</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>Uncertain</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Pre-commit</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Committed</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>
Termination Rule

• If some process is aborted, send “abort” decision to all participants.

• If some process is committed, send “commit” decision to all participants.

• If all active processes are uncertain, send “abort” decision to all participants.

• If some process is pre-committed and none are committed,
  – send “pre-commit” to all participants,
  – wait for “acks”,
  – then send “commit” to all participants.
Failures in Termination Protocol

• Coordinator only collects state from active participants.
  – So protocol is non-blocking

• If coordinator fails, new coordinator is elected and termination protocol begins again.
Message Complexity