Feb 13
Distributed Commit Protocols

Vijay Chidambaram

The problem

- We have multiple sites
- Data is divided among these sites (not replicated)
- We want to run a distributed transaction over this distributed data with ACID guarantees:
  - Atomic: all sites are updated with tx results, or none are
  - Consistent: updates are applied in a consistent fashion at all sites
  - Isolation: no tx sees partial results of other concurrent tx
  - Durability: after the tx commits, even if all sites lose power, the data is still available after reboot

Distributed Txs

- Each site has a Transaction Manager (TM)
- Txs are submitted to the TM at their local site
- Read(x) or write(x) forwarded to TM where X lives
- A Tx is Committed or Aborted — decision needs to be taken by all sites
- Incorrect if one site decides to Commit, and another site decides to Abort
- What to do if a tx site fails?

Failures in a distributed system

- A site could fail, or a link connecting sites could fail
- We assume there is a path from every site to every other site
- Failures are fail-stop
- Partial failure: some nodes are up, others are down
  - Partial failures are tricky because sites are unsure of the status of other sites (no common knowledge)
- Total failure: all nodes are down

Network Partition

- Dividing the network into two disconnected graphs, with no messages flowing from one side of the graph to the other
- Can happen due to router/link failures
- Usually temporary
- Once connection is restored, nodes can then talk to each other across the partition

Detecting Failures

- Done via time-out T
- If a node hasn't responded in time T, it is assumed to be dead
- Setting the value of T is tricky
  - If T is too high, we detect failures very late
  - If T is too low, we have false positives where we detect failures spuriously

Conditions to commit

- A Tx T can commit at Site S if:
  - T has read only committed values
  - All the values written by T are durably stored
- Each site contains a distributed coordination log where information is recorded about txs

Atomic Commitment Protocol

- ACI: All processes that reach a decision reach the same one.
- AC2: A process cannot reverse its decision after it has reached one.
- AC3: The Commit decision can only be reached if all processes voted Yes.
- AC4: If there are no failures and all processes voted Yes, then the decision will be to Commit.
- AC.5: Consider any execution containing only failures that the algorithm is designed to tolerate. At any point in this execution, if all existing failures are repaired and no new failures occur for sufficiently long, then all processes will eventually reach a decision.

Implications of the conditions

- A process can unilaterally abort the tx by voting No
- Once a process can voted Yes, it cannot later unilaterally decide No
- Uncertainty period: where a process has voted but is uncertain about the outcome (happens only when voting Yes)
- Proposition 7.1: If communication failures or total failures are possible, then every ACP may cause processes to become blocked.
- Proposition 7.2: No ACP can guarantee independent recovery of failed processes.

Two Phase Commit Protocol

- The coordinator sends a VOTE-REQ (i.e., vote request) message to all participants.
- When a participant receives a VOTE-REQ, it responds by sending to the coordinator a message containing that participant's vote: YES or NO. If the participant votes No, it decides Abort and stops.
- The coordinator collects the vote messages from all participants, If all of them were YES and the coordinator's vote is also Yes, then the coordinator decides Commit and sends COMMIT messages to all participants.
- Otherwise the coordinator decides Abort and sends ABORT messages to all participants.
- Each participant that voted Yes waits for a COMMIT or ABORT message from the coordinator. When it receives the message, it decides accordingly and stops.

Augmenting 2PC

- 2PC as outlined before satisfies AC 1-4
- Does not satisfy AC5: ".. if all existing failures are repaired and no new failures occur for sufficiently long, then all processes will eventually reach a decision."
- To handle failures, each site should durably record its vote and decision in the distributed coordination log
- Termination protocol: uncertain processes contact coordinator to learn of decision
- Cooperative Termination Protocol: uncertain processes contact each other instead of the coordinator