L19: Replicated state machines

Sam Madden
6.033 Spring 2014
Last Time: Two-Phase Commit

(Not to be confused with two phase locking)

Allows transactions to be run across multiple distinct nodes (each with their own data)

Key idea:
- **prepare** all nodes to commit
- atomically **commit** them via a single log record
Two Phase Commit

**Coordinator**

- Commit point
- If all yes, log commit
- Log done

**Worker**

- prepare
  - yes
  - Log prepare
  - commit
  - ack
  - Log commit

- After prepare, worker cannot decide xaction outcome on its own.
  - Even if it crashes, it must recover into prepared state!
Single-copy consistency

• Operations appear to execute as if there is only a single copy of the data
Replicated state machine

• A general approach to building consistent replicas of a server:
  – Start with same initial state
  – Provide each replica with the same input and in same order
  – Ensure all inputs are deterministic
RSM is hard in practice [comp.risks]

- Leap day 2012, Microsoft's Azure
  - Configuration error crashed all servers
- April 2011, Amazon's EC2
  - Replication storm exhausted all storage
- September 2010, Facebook clients
  - Clients overwhelmed database after misconfig.
- September 2009, Google's gmail
  - Request routers overloaded

*Complete outages*
RSM with view server

• Primary must wait for backup to accept each request
• Non-backup must reject forwarded requests
• Non-primary must reject direct client requests
• Primary in view $i$ must have been primary or backup in view $i-1$
• New backups copy state from primary
RSM w/ View Server Example

View Server

1: R1, R2

Replica 1

View: 1: R1, R2

Replica 2

View: 1: R1, R2
Client

View: 1: R1, R2

View Server

1: R1, R2

Replica 1

View: 1: R1, R2

Replica 2

View: 1: R1, R2
Client

View: 1: R1, R2

View Server

1: R1, R2
2: R2, --

Send request to Replica 1

Replica 1

View: 1: R1, R2

Error!

Replica 2

View: 2: R2, --

Replica 2 refuses to process request
Client

View:
2: R2, --

Get view

View Server

1: R1, R2
2: R2, --

Replica 1
View:
2: R2, --

Replica 2
View:
2: R2, --
Client

View: 2: R2, --

1: R1, R2
2: R2, --

View Server

Replica 1
View: 2: R2, --

Replica 2
View: 2: R2, --

Send request to primary

Send request to backup
Summary

- Single-copy consistency
- Replicated state machine
- Consensus with view server
- Wednesday: distributed consensus