EECS 491
Introduction to Distributed Systems

Fall 2019

Harsha V. Madhyastha
Primary Backup Replication

Client → Primary → Backup → Backup → Backup

Primary
Primary Backup Replication

- Promote one of the backups if primary fails
- Replace any failed backup

- When should primary sync with backups?
  - Before making state change externally visible
  - Primary and backups must be externally consistent

- What to sync?
  - Entire state when bootstrapping new backup
  - Thereafter, forward every update
View service

- Maintains current membership of primary-backup service (called view)
  - View number, primary, backup

- When does view service change view?
  - When primary or any backup fails
  - Periodically exchange heartbeat messages to detect failures
Transitioning between views

- Why are backups included in current view?
- Clients query view service only for primary
  - Can only promote a previous backup as new primary

- How does view service know if backup up-to-date?

- Two scenarios for ill-timed primary failure:
  - Primary applies operation but fails before syncing with backup
  - Primary fails before new backup is initialized
Transitioning between views

- View change has three steps:
  - View service announces new view
  - Primary syncs with new backup if there is one
  - Primary acknowledges new view
- View service knows backup is up-to-date once it receives ACK for new view from primary
- Stuck if primary fails in midst of view change
- Liveness detection timeout > State transfer time
Scalability of View service

- Too much load on view service if all clients contacts it before every operation

- Clients can cache view across operations

- When to invalidate cached view?
  - When no/negative response from primary
Construct scenario where primary that client contacts is not the primary, but it thinks it is.
Split Brain

Client

View service

S1

(1, S1, _)
(2, S1, S2)
(3, S2, _)

S2

(2, S1, S2)
(3, S2, _)

(2, S1, S2)

(2, S1, S2)
Avoiding Split Brain

- Primary must forward all operations to backups
  - Goal: Get ACKs from backups that they too recognize primary

- Why can’t backups be mistaken about who is primary?
  - Only a backup can be promoted as primary
View service

- **Valid sequence of views:**
  1. $(1, S1, _) \rightarrow (2, S1, S2) \rightarrow (3, S1, S3) \rightarrow (4, S3, S4) \rightarrow (5, S4, _)$

- **Examples of invalid transitions between views?**
  1. $(1, S1, S2) \rightarrow (2, S3, S4)$
  2. $(1, S1, S2) \rightarrow (2, _, S2)$
  3. $(1, S1, _) \rightarrow (2, S2, S1)$
Announcements

- Project 1 due on Thursday
- Make sure to test on CAEN
  - Ensure changes only to *impl.go files

- How to detect failed RPCs?
  - Only rely on return value
  - Contents of reply argument set only upon success
  - Contents of reply undefined for failed RPCs
Service Development

- Getting coordination right between primary and backups is tricky
  - Easy to mess up

- Must make replication transparent to developer
Transparent Replication

Replicated State Machine

Application

Ordered Updates

Server1

Replicated State Machine

Updates

Server2

Replicated State Machine

Updates

Application

September 17, 2019

EECS 491 – Lecture 5
Transparent Primary Backup

- Application relies on library to keep primary and backups in sync
  - Receive message from client
  - Sync with backups before sending response to client

- Will this solution work?
Example: Bank Server

```c
last_month uint
Deposit(user, amount) {
    balance[user] += amount
    if (curr_month > last_month) {
        add 1% monthly interest
        last_month = curr_month
    }
    return balance[user]
}
```
Example Execution

Primary
- Receive deposit request
- Update balance
- Read current time T1
- Add interest
- Return new balance

Backup
- Receive deposit request
- Update balance
- Read current time T2
- Return new balance

Insufficient to relay inputs from primary to backup
Also need to relay sources of non-determinism

How to capture this?
Virtual Machines

Virtual Machine Monitor

Applications

Process
File system
Virtual memory

Operating System

CPU
Disk
RAM
RSM with Primary Backup

Server1

Application

Operating System

Virtual Machine Monitor

Hardware

Server2

Application

Operating System

Virtual Machine Monitor

Hardware
The Design of a Practical System for Fault-Tolerant Virtual Machines

Daniel J. Scales, Mike Nelson, and Ganesh Venkitachalam
VMware, Inc
{scales,mnelson,ganesh}@vmware.com
VMM-based Primary Backup

- Primary and backup execute on two virtual machines
- Primary logs inputs and outputs
- Backup applies inputs from log
- Primary-backup monitor each other
  - If primary fails, backup takes over