# CPSC 416 Distributed Systems

Winter 2023 Term 1 (September 26, 2023)

**Tony Mason (fsgeek@cs.ubc.ca), Lecturer**

# Logistics

# Teaching Assistants

Andy Hsu (andy.hsu@alumni.ubc.ca)

Hamidreza Ramezanikebrya (hamid@ece.ubc.ca)

Jonas Tai (jonastai@student.ubc.ca)

Cathy Yang (kaiqiany@student.ubc.ca)

# Office Hours

Remember: Use Piazza for **all** official course-related communications

- Not on Piazza? Not official.
- Canvas "comments/messages" **are not monitored**

Office Hours:

| Who | When | Where |
|---|---|---|
| Tony | Monday 14:00-15:00<br>Wednesday 16:00-17:00 | Discord |
| Andy | Thursday 19:00-20:30 | Discord |
| Hamid | Friday 16:30-18:00 | Kaiser 4075 |
| Jonas | Thursday 11:00-12:30 | X150, Table 1&2 |
| Cathy | Friday 09:00-10:30<br>(Starting Sep. 22) | X237 |

# Self-Assessment

**This week**

- Post-lecture self-assessment activity – Due Tuesday (Sep 26 @ 17:00)
  - Structure change: fewer reflection questions
  - Structure change: more content questions (T/F, MC)

Note:

- You are strongly encouraged to collaborate with others on this
- You should use tools at your disposal to answer these questions
- As previously noted, you get full credit if you submit. **Do not forget to submit it.**

# Today’s Failure

# Design Project 1

# Learning Goals

Change focus from *coding* to *designing*

Learn how to design

- Learn by “doing”
- Most people will struggle with this and turn out a poor-quality design

Learn what works and what doesn’t work

- Peer feedback – expect it to be a struggle for the first time
  - Likely see some poor designs
  - Likely won’t know what a *good* design needs to be
- Implementation – expect to be unhappy with the design
  - **This is the real purpose – reflect on *what would have made it better.***
- Implementation Report: how to make the design *better*.

# DSLabs Challenges

The DSLabs does **not** provide you with a model of what it expects.

**It uses tests to define the expected behaviour of the system.**

- Trying to infer the problem from DSLabs is challenging
- Goal: write the design document that should have been included with DSLabs!

Components:

- Clients
- Identical database server(s)
- View Server

# Client

A **client** is some independent execution state that:

- Can query the view server for the current view
- Can send a request to the service (primary and/or backup)
- **Will not proceed to a new request until the current request is answered** (this is the general behaviour model.)
- May re-send the request as many times as necessary (“At least once”)

# Server

In this system a server is:

- An identical implementation of the same logic
- May have some or all of the contents of the “actual” database
- Understands the *reconciliation* protocol with other servers
- Can take on the role of *primary* or *backup*

Open question: can the backup answer any queries from clients?

Initial state:

- Server is nominated as primary by View Server. **Database is empty**

# Server

Responsibilities:

- Respond to client requests (get/put/delete/append)
- Respond to heartbeat messages (“pings”) from other components
- Synchronize changes between primary/backup server

Failures:

- Server crashes and recovers
- Network partitions

# View Server

A *view server* in this design lab has three distinct requirements:

- It must respond from a request for the current (active) view
- On first initialization it must choose a server to be the *primary*
- It may **propose** a change to the view
  - To add a backup
  - To switch the backup to primary

Failures it must consider:

- Primary or backup failure (partition, crash, etc.)
- Network partition (it cannot talk to the servers)

# Key Points

Define your guarantees:

- Consistency: Replicated database **will not roll backwards** (client perspective)
- Replicated database **may be temporarily unavailable**
  - Question: how “temporarily?” 1 second, 1 minute, 1 decade?
  - Question: can you improve its availability without compromising consistency?

Work through failure scenarios

- Partition scenarios

Define *recovery strategies*

# Learning Goals

# Learning Goals (Petrov Chapter 8, Part 2)

Goals for this conversation is understanding:

- FLP Impossibility
- Failure Models
- Recovery (“Handling failures”)

Stretch: *heartbeats* is the first part of Chapter 9, useful for Design Project 1.

# Questions?

THE UNIVERSITY OF BRITISH COLUMBIA