CPSC 416 Distributed Systems

Winter 2023 Term 1 (September 14, 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 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 (Starting Sep. 22)	X237

Self-Assessment

This week

- Post-lecture review assignment (Useful?) – Due before *next* lecture
- Distributed Systems Design Recipe assignment – Due Friday (Sep 15 @ 23:59)

Note:

- You are strongly encouraged to collaborate with others on this
- You should use tools at your disposal to answer these questions
  - **Including your favourite large language model** (e.g., ChatGPT, your older sibling, or your pet.)
  - Share your prompts on Piazza (Discord #chatgpt channel)
- As previously noted, you get full credit if you submit. **Do not forget to submit it.**

Design Project 1

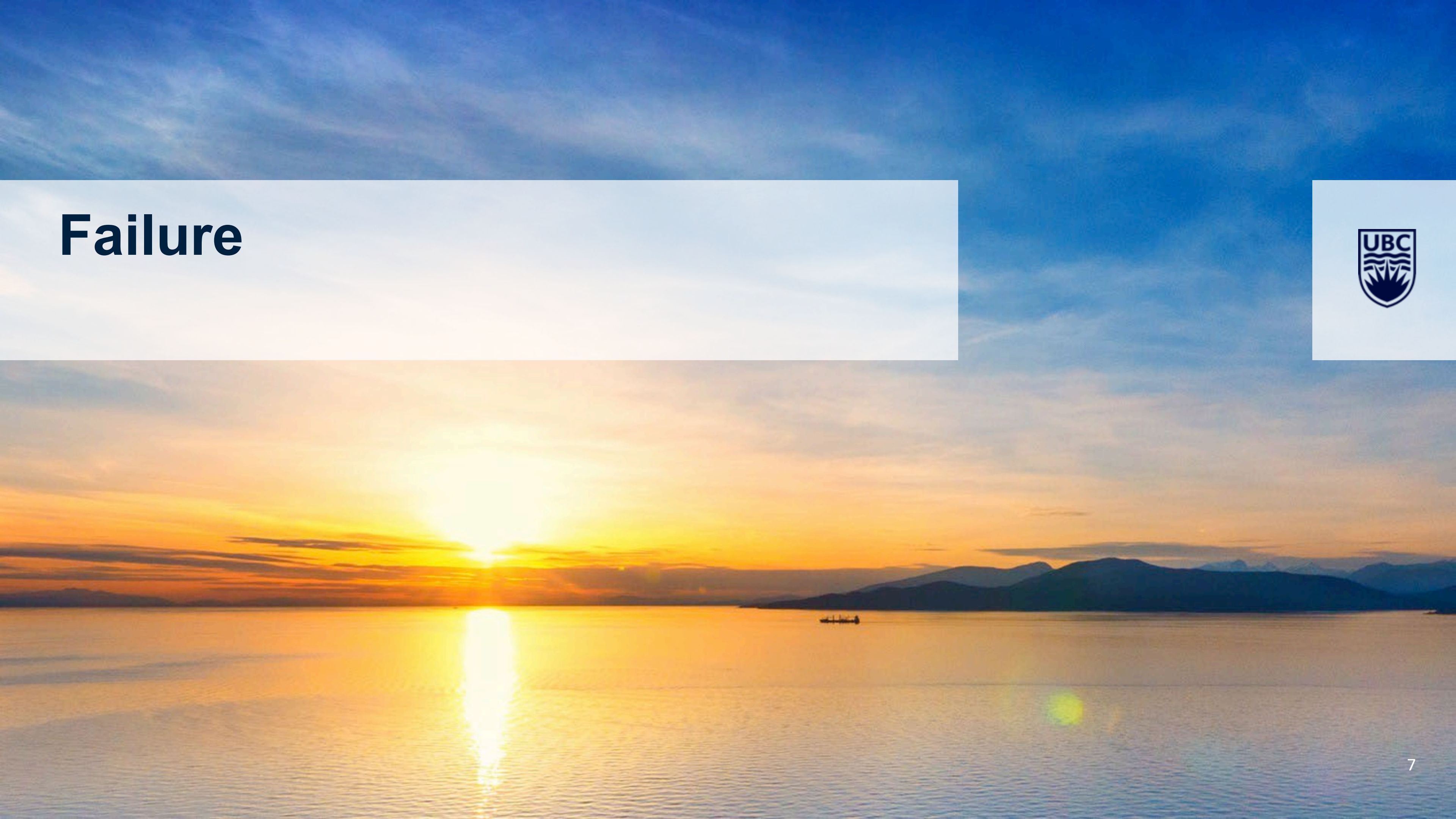
Distributed Systems Design Recipe: Due September 15, 2023 @ 11:59 PM

Design Project 1: Primary/Backup Replication

- See DSLabs Project 3: https://github.students.cs.ubc.ca/CPSC416-2023W-T1/project3
- Deliverable 1: Project Design (Due September 26 @ 17:00 PT on Canvas)
- Deliverable 2: Design Review (Due October 3 @ 17:00 PT on Canvas)
- Deliverable 3: Code (Due October 10, 2023 @ 17:00 PT on Gradescope)
- Deliverable 4: Implementation Report (Due October 10 @ 23:59 on Canvas)
- Deliverable 5: Report Review (Due October 17 @ 17:00 on Canvas)

Note: Submission = 100% credit. Deliverables 1, 2, 4, and 5 will be shared with class.

Do not put personally identifiable information in your deliverables if you do not want them shared.

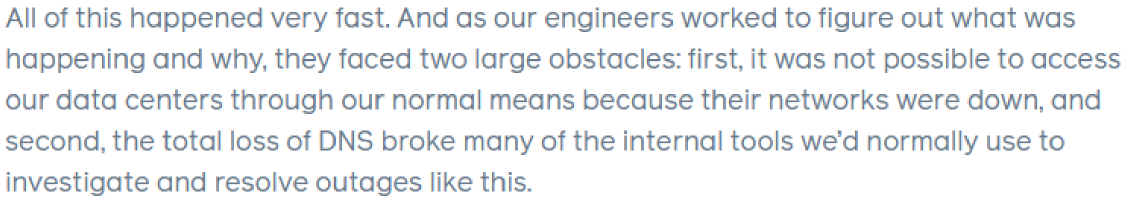


Failure

Today's Failure

October 4, 2021

Facebook Outage

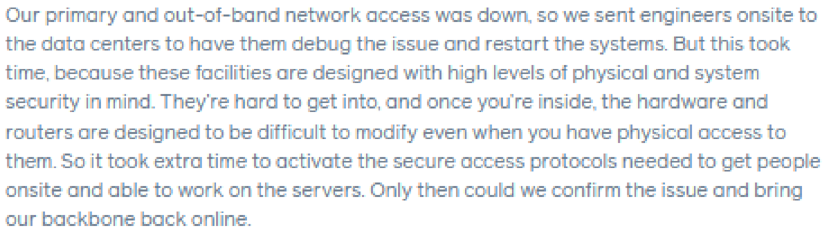


All of this happened very fast. And as our engineers worked to figure out what was happening and why, they faced two large obstacles: first, it was not possible to access our data centers through our normal means because their networks were down, and second, the total loss of DNS broke many of the internal tools we'd normally use to investigate and resolve outages like this.

Root cause: Incorrect command was issued

Secondary cause: Audit tool that should have blocked the incorrect command was flawed.

Result: Facebook disconnected from the Internet



Our primary and out-of-band network access was down, so we sent engineers onsite to the data centers to have them debug the issue and restart the systems. But this took time, because these facilities are designed with high levels of physical and system security in mind. They're hard to get into, and once you're inside, the hardware and routers are designed to be difficult to modify even when you have physical access to them. So it took extra time to activate the secure access protocols needed to get people onsite and able to work on the servers. Only then could we confirm the issue and bring our backbone back online.

We prepare for the worst and we prepare for the best and sometimes we get the unknown.

Reading Discussion

Learning Goals (Chapter 5)

Chapter 5: Transaction Processing and Recovery

- Understand data organization to enable *fixing things when something breaks*?
- Explore how concurrent access can occur in the presence of partial/complex changes
- What is a log (or journal) and how does it work?
- How does serialization work?

Learning Goals (Chapter 7)

Chapter 7: Log-Structured Storage

- Understand an extreme model of logging: *everything is in the log*.
- How does tiered storage work?
  - DRAM
  - NVMe/SSD
  - Hard Drive
  - Archival (Tape, Optical)
- What is write amplification and why is it an issue?
- What are cool ways of optimizing storage systems?

Big Question

What does this have to do with **distributed systems**

Answer: **everything**

- **Storage is what we distribute**
- **Service levels differ: database versus file system**

Most common model for data replication: **replicated log**

- Hence, you now know what a log is
- You should already know about finite state machine (FSM)

Simplest model of replication: Replicate a log and feed it to duplicate copies of the FSM

Discussion

So you hate talking about databases… Let’s talk about concurrency, logs, and transactions.

Petrov, Chapters 5 & 7

Note: I skip Petrov Chapter 6 because it is more about B-Trees; interesting but not necessary.

Chapter 3: High Level

Cell Layout and Data Types

Slotted Pages

Variable Size Data

B-Tree Layout and Pages

Versioning and File formats

Checksumming and Data Integrity

Chapter 5: High Level

Delta Nodes and Update Operations

Structural Modification Operations (SMOs)

Latch-Free Operations

Concurrency and Latching

Reader-Writer Locks (**my favourite** locks)

Transaction Isolation (more on this later, too)

Buffer management (correctness and optimization)

Chapter 7: Log-Structured Storage

Aside: check out **An Implementation of a Log-Structured File System for UNIX**

Write-Ahead Logging (WAL)

Checkpoints

Operations versus Data Logging

Shadow Paging

Why Database Internal **matter**

Bw Trees

Example: Apache Kafka

Widely used distributed system platform

Robust: replication, fault tolerance, handles node failures

Scalable: handles millions of events per second, scale out by adding nodes

Real-time: low-latency, real-time data streaming

Data pipelines: producer-consumer model with robust **storage**

Streaming Applications support

General purpose model: add sources and syncs including other storage systems.

I’ll discuss this more later

Questions?

THE UNIVERSITY OF BRITISH COLUMBIA