CPSC 416 Distributed Systems

Winter 2023 Term 1 (September 7, 2023)

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

Welcome!

Welcome to CPSC 416 Winter 2023 Term 1 (September 2023)

Background

Second time I have taught CPSC 416

Why you should take this course:

- Learn core technology
- Challenge yourself
- Worry about *failure*

Why you should not take this course:

- Challenging subject
- Substantial work
- Don’t like *failure*

Course Instructor

Tony Mason, Sessional Lecturer

At UBC since 2017

- Also teaching at Georgia Tech (OMSCS)
- Industry teaching (public/private tech seminars) since 1992
- Research in usable storage (PhD candidate)
  - Where is that file?

Distributed Systems Background:

- Stanford: Cheriton's Distributed Systems Group
- Transarc: AFS, DCE/DFS
- FORE Systems: ATM Networking
- Consulting

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)

Waitlist

Current waitlist has 86 people!

- Come to class
- Work on assignments
- People will drop

Current 80 students:

- Consider workload

Resources

Canvas - https://canvas.ubc.ca/courses/135228

My website - https://fsgeek.ca/teaching/cpsc-416-winter-2023-term-1/

Gradescope - https://www.gradescope.ca/courses/12183

Piazza - https://piazza.com/class/lm3m17tteq1189

Discord - https://discord.gg/t6qDS5KH (Expires 2023/09/13 @ 15:20 PT)

Twitch - https://www.twitch.tv/fsgeek2

Communications

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

- Not on Piazza? Not official.

TA Office Hours: TBA

Instructor:

- Private Meeting: by appointment
  - In person (must book a private space)
  - Online (Discord or Zoom)
- Office Hours (Discord)
  - Monday 14:00-15:00 PT
  - Thursday 16:00-17:00 PT

Course Overview

Learning Goals

Schedule

Exam: Final Only

Advice:

- Make sure you submit **everything.**
- Plan before you code
- Use Piazza and/or Discord
- Pick a good group

Learning Goals

Understand fundamental concepts:

- Persistence
- Consistency
- Two-phase commit (2PC) and three-phase commit (3PC)
- Consensus
- Recovery

Key Skills:

- Design
- Communications
- Meta-analysis (self-reflection)

Workload

Ivan's CPSC 416:

- *The workload for this course is easily double that of any other course I had this term.*
- *Ivan has very high expectations of his students.*
- *I love and hate the fact that this class was a "sink or swim" approach to learning.*

My distributed systems course (GT CS 7210):

- *The most difficult course I've ever taken.*
- *Project 5 [Paxos] was harder than anything I did in a co-op*

Projects

Based on DSLabs: https://github.com/emichael/dslabs

- Some changes to allow using Gradscope
- Your code performance or test passing **is not** part of your grade

Lab 1: Intro

Lab 2: Client-Server

Lab 3: Primary-Backup

Lab 4: Paxos

Lab 5: Sharded Key-Value Store

These go from easy (1/2) to challenging (3) to difficult (4/5).

Distributed System Examples

Blockchains (including Bitcoin and Ethereum)

Hadoop File System (HDFS)

Cyber-physical systems

Folding@home

Kafka

Domain Name System (DNS)

Cloud Services: AWS, Azure, GCP

Distributed Databases

Failure

Distributed Systems focuses on handling certain classes of failures

- Destruction of a single data center (hence this term's theme photo – a warehouse on fire)
- Network disruption
- Failure of a single piece of hardware

Recovery is what makes this hard

- **Backups** are a form of recovery
- How do you know your backups *work* –
  - Wrong time to discover they don't is after a failure.
  - Sorry, your business has failed

Course Structure

Self-Assessments

Design Projects

Capstone Project

Final Examination

Course Structure: Self-Assessment

Before class: read the assigned reading (or watch the assigned video)

Class: be prepared to discuss the reading (or other class activities)

- My goal is to minimize didactic lecture
- This is your best opportunity to ask questions

Post-class: you will be assigned a self-assessment activity

- **Opens** at the end of lecture
- **Closes** at the start of the *next* lecture
- **Grading**: did you submit it. That's it

These activities will account for **10%** of your grade

Course Structure: Design Projects

Three design projects

- DSLabs Project 3: Primary-Backup Replication
- DSLabs Project 4: Paxos
- DSLabs Project 5: Sharded Key-Value Store

Phase 1: Construct your design for the solution to the project

Phase 2: Review **three other** designs, provide feedback to the original author

Phase 3: Pick **one** of the three designs, implement it, submit it

Phase 4: Write an implementation review: “Now how good was that design?”

Phase 5: Review **three other** implementation reviews.

Design projects are **45%** of your total grade.

Course Structure: Capstone Project

Groups allowed **up to five students**

Weekly Reports: document your progress
Project Design: Due November 23, 2023
Project Code: Due December 5, 2023
Project Report: Due December 7, 2023
Project Presentation: Due December 7, 2023

Capstone Project is **25%** of your total grade.

Course Structure: Final Exam

Similar to last term:

- 10 True/False questions
- 30 Multiple-choice questions
- 10 Multi-option match questions (“A & B are true” , “A & C are true”)

Questions attempt to probe your understanding of the material.

Final exam is **20%** of your total grade.

Rationale

My goal is to de-emphasize the *grade* you get in this course:

- Primary goal is *learning*; grading gets in the way (gamification)
- Remove the possibility of failure
  - Self Assessment + Design Projects = 55%
  - Credit is based **upon submission**.

The primary benefit to you for taking this course is

- Learning why *planning* and *design* are highly effective tools
- Understanding how to provide constructive feedback
- Hands-on experience with distributed systems challenges (hot topic)

Failure

Distributed Systems must handle *failure*

Failure occurs *all the time*.

Exhaustive failure testing *is not possible*

Each lecture I will share some example of a failure

- Real systems
- Real failures
- Real explanations

Failure: World of Warcraft

November 28, 2022: Blizzard Entertainment Released “World of Warcraft: Dragonflight”

- Simultaneous release all over the world at 15:00 PT

Everything melted down: https://us.forums.blizzard.com/en/wow/t/an-engineering-update-on-the-dragonflight-launch/1437657

Distinct changes interacted negatively:

- Timed Event Trigger (3 pm PT release)
- Encrypted Data

Failure (2)

We now know that the lag and instability we saw last week was caused by the way these two systems interacted. The result was: they forced the simulation server (that moves your characters around the world and performs their spells and abilities) to recalculate which records should be hidden more than one hundred times a second, per simulation. As a great deal of CPU power was spent doing these calculations, the simulations became bogged down, and requests from other services to those simulation servers backed up. Players see this as lag and error messages like “World Server Down”.

Failure (3)

As we discovered, records encrypted until a timed event unlocked them exposed a small logic error in the code: a misplaced line of code signaled to the server that it needed to recalculate which records to hide, even though nothing had changed.

Failure (4)

Mitigation makes it worse!

Boats have been a problem in the past, so we turn on portals while we continue investigating. Our NFS is clearly overloaded. There's a large network queue on the service responsible for coordinating the simulation servers, making it think simulations aren't starting, so it launches more and starts to overwhelm our hardware. Soon we discover that adding the portals has made the overload worse, because players can click the portals as many times as they want, so we turn the portals off.

Failure (5)

Recovery after failure can cause more problems

Pushing a fix to code used across so many services isn't like flipping a switch, and new binaries must be pushed out and turned on. We must slowly move players from the old simulations to new ones for the correction to be picked up. In fact, at one point we try to move players too quickly and cause another part of the service to suffer. Some of the affected binaries cannot be corrected without a service restart, which we delay until the fewest players are online to not disrupt players who were in the game. By Wednesday, the fix was completely out and service stability dramatically improved.

MURPHY WAS AN OPTIMIST

Discussion

Too much background… Let's talk.

Petrov, Chapters 1 & 2

Questions?


THE UNIVERSITY OF BRITISH COLUMBIA
THE UNIVERSITY OF BRITISH