Distributed Systems
CPSC 416
Winter 2016

Course: January 4 - April 8, 2016

Jan 4, 2016 Lecture (first class!)

Course staff

- Ivan Beschastnikh, instructor
- Patrick Colp, TA
- Lynsey Haynes, TA

Logistics

- Waitlist is deep; room at capacity :-(
    - Keep coming, some people will drop, but not everyone will get in.
- To others: consider dropping if you have other courses that look more interesting

Logistics

- Everything on the website, updated continuously:
  Short: http://www.ugrad.cs.ubc.ca/~cs416/
  Long: http://www.cs.ubc.ca/~bestchai/teaching/cs416_2015w2/index.html

- Use Piazza for all course-related communication

Course overview via the website

- Learning goals
- Go programming language (start learning!)
- Schedule (a work in progress)
  - Assignment 1 due Jan 13 (next Wed)
- Assignments (2 individual; 2 in group of 2)
- Group projects (deliverables are write-ups!)
- Exam (just a final)
- Advice for doing well
  - learn Go (a must to pass the course)
  - don’t hack, engineer
  - choose team, wisely
  - invest time into project and write-ups
  - reach out on Pizza/email for help.
- Collaboration guidelines

Distributed system examples

- YouTube
  - Videos are **replicated** (multiple machines host the same video)
  - **Scalable** wrt. client requests for videos (internally **elastic** — can throw more machines at the service to have it scale out further)

Distributed system examples

- DropBox (or google drive)
  - **Replicated** content across personal devices
    - Supports **disconnected operation** (can work while disconnected, and synchronize when re-connected)
    - Maintaining data **consistent** across devices
  - Supports sharing; **access control** policies (security!)

Distributed system examples

- NASDAQ
  - **Transactions** (e.g., ACID semantics from databases). Many DBMS concepts apply to distributed systems!
  - Strong **consistency** and **security** guarantees (otherwise people would not trust it with money)

Some challenges

- Synchronizing multiple machines (protocol complexity)
- Performance (how do you define/measure it?)
- Maintaining consistency: strong models (linearizable) to weak models (eventual) of consistency
- Failures: machine failures (range: failure stop to byzantine); network failures (just a few: disconnections/loss/corruption/delay/partitioning)
- Security (how to prevent malicious control of a single host in a system escalating into control of the entire system?)

For Wednesday

- Install Go on your personal machine
- Work through Tour of Go! and other tutorials.
- **Practice Go!** Start on assignment 1.
- **Bring laptop to class**, if you have one (we will be hacking in Go most of the class)