Distributed Systems 1

CUCS Course 4113
https://columbia.github.io/ds1-class/

Instructor: Roxana Geambasu

Lecture 0

Course Introduction

Interested in...

1. scalable web services?
2. big data processing and management?
3. and the large-scale infrastructure systems making these possible?

If so, you're in the right room/zoom.

Distributed Systems Classes at CUCS

1. **Fundamentals of Distributed Systems (DS1, Roxana, Fall).**
   - Teaches basic concepts, principles of large-scale DS design.
   - Discusses application of those concepts/principles in real-world systems (e.g., Google/Facebook/Amazon).
   - Homework series teaches how to build basic DS.

2. **Advanced Distributed Systems (DS2, Roxana, Spring).**
   - Research seminar that discusses topics in depth by reading the most influential papers that originated them.
   - We include both classical papers on well-understood topics and new, state-of-the-art papers.

Related CU Classes

- Multiple cloud computing/web programming/big data processing classes are offered @CU.
  - Those classes teach you how to **use** various popular DSes.
  - This class teaches you the how those and other systems are **built**, so you can build and use them better in the future.

- Similar to the **OS class**, but for the distributed setting. And in the "cloud" era, everything is distributed!
  - If you want to do "big data," you need DS.
  - If you want to do mobile apps, you need DS.

This Class

1. Foundational concepts of large-scale DSes: challenges, algorithms, techniques, abstractions.

2. The inner-workings of several DSes serving as infrastructure for large companies.
   - E.g.: Google's protobuf/Spanner/MapReduce, Yahoo's Hadoop, Amazon's Dynamo, etc.

3. How to build a DS yourself: through a series of coding assignments, you will build your own DS.

Topics

1. **Distributed systems primer**: challenges and goals of distributed systems.
2. **Communication models**: remote procedure calls (RPC), RPC libraries, failure models, semantics.
3. **Time and coordination**: challenges, physical and logical clocks, distributed mutual exclusion.
4. **Sharding, replication, and the agreement problem**: commitment and consensus, use cases for each.
5. **Local transactions (background)**: ACID semantics, concurrency control, recovery mechanisms.
6. **Commitment protocols:** 2-phase commit, 2-phase commit, safety/liveness tradeoffs with the two.
7. **Consensus protocols**: Paxos overview, key ideas, basic algorithm, examples, liveness failure mode.
8. **Replication architectures:** fault-tolerant architectures (primary/secondary, master/slaves); the design of Google's Chubby lock service; the 2PC+Paxos approach to both scalability and fault tolerance.
9. **Case study: Google's Spanner:** TrueTime, Spanner and its fault-tolerant, linearizable, distributed transactions.
10. **Other consistency and isolation semantics**: sequential, causal, and eventual consistency; a few isolation semantics; mechanisms to achieve each; tradeoffs between them.
11. **Distributed computation:** MapReduce design, TensorFlow design, approximate computing engines.
12. **Distributed systems security primer**: authentication protocols, Needham-Schroeder, Kerberos, byzantine fault tolerance, maybe blockchain.

Important Addresses

- Website: https://columbia.github.io/ds1-class/.
- Discussions: Piazza.
  - CAs will be active on Piazza, most homework questions should go there.
  - For questions on lecture material, please come to class or my OH and pose them there!

Teaching Staff

- **Instructor:** Roxana Geambasu, Associate Professor of CS.
- **Advising CAs:**
  - Mingen Pan, MS in CS
  - Julius Song, MS in CS
  - Jinhai Su, MS in CS
  - Harrison Wang, ugrad in CS
  - Alex Dziena, MS in CS (CVN)

Grading CAs:
- Jinhai Su, MS in CS
- Ke Li, MS in CS
- Possibly one more TBD

- All CAs took DS-1 last year and did excellently. (Introduce CAs on zoom.)

Prerequisites

- You must have **solid programming experience** (C, C++, Java), preferably system-level programming experience.
- Columbia courses (or equivalents):
  - COMS W3137 Data Structures and Algorithms;
  - COMS W3157 Advanced Programming;
  - COMS W3827 Fundamentals of Computer Systems;
  - Optional, but very useful: COMS 4118 Operating Systems.
- If you lack these prerequisites, **do not take** the class, because heavy coding accounts for vast portion of grade.
  - Use HW1 to determine if you have sufficient experience.

Grading

- Grading formula: **100% for five graded homeworks.**
  - 10%: extra credit for extraordinary contributions on Piazza and/or in class (think: someone who has significantly improved the class).

- Grading policies:
  - We offer a 72-hour grace period over the entire semester. Once you consume those, submitting your homework one second late will result in a zero for that homework.
  - Can discuss, preferably openly on Piazza, but *NOT* look at others' code or answers.

Homework 0

- Not graded but mandatory and important:
  - Reviews academic integrity and collaboration policies. Please be sure to read them carefully before you sign!
  - Sets up github, which you'll use for all homeworks.

- Preliminary deadline: Wednesday, September 16, 2020.
  - Depends on github classroom approval.

In Class

- 2020 version is **online only**:
  - Participation is not required and attendance is not taken.
  - Classes are recorded and posted to Courseworks Video Library. (Not sure, but recordings may be kept only for 30 days.)
  - If you miss a class, please view recording **before you join next class**.
- Structure:
  - I will cover the week's topic lecture-style. Detailed notes on class website.
  - Want to encourage interaction but cannot talk over each other. Please **post questions in the chat** or **raise your hand**. I will invite you to speak.
  - There will be polls and breakouts. Please be active on these!
  - Bio break at 2:20pm-2:30pm.

An Ask

- This is the first time I'm teaching a large online-only class.
- I will try hard to make the classes interactive and educative.
- But I will need your help: any **feedback** you can give me on what works and what doesn't will be highly appreciated!
- In the second part of the semester I want to invite students who have industry experience with large, interesting DSes to talk about their experience. Please email me if you have such experience and interest in participating.