CPS 512: Distributed Systems
Spring 2022

Class Meetings

TTh 3:30 – 4:45 in Old Chemistry 116

Instructor

Jeff Chase
Office hours: WF 2:00-3:00 PM, or at other times as available.
By Zoom until further notice. My office is D306 LSRC.

Teaching Assistant

Shihan Lin
Office hours: TBD

CPS 512 is an advanced course in distributed and networked systems. We focus on core concepts in distributed systems, using geo-distributed mega-services on cloud infrastructure as a motivation and driving example. Modern cloud applications are layered above common service platforms that handle the hard problems: elastic infrastructure. tracking groups of participating servers (views), distributing state and functions across a group, coordinating control and ownership of data, managing consensus, and recovering from server and network failures. The course focuses on the design of these service platforms and their abstractions.

This offering has a modest programming component and a substantial independent project, making it suitable for a broader audience than previous semesters. We use a subset of the Java-based DSLabs. Since CPS 512 is a "quals course" there are also two exams.

Topics. The course divides loosely into three parts. This semester we start with Internet naming (DNS) to illustrate multi-domain services and the concepts of principals, identity, authority, trust, governance, and secure communication. Part two moves into challenges of stateful services and elastic scaling, illustrated with elements of the Google cloud network stack: Kubernetes service abstractions, request routing, load balancing, and coordination services. We then dive deeper into foundational distributed systems topics that underlie these systems: distributed transactions, geo-replication, logical time and causality, eventual consistency with vector clocks, views and leader election, and consensus. As time allows we explore how these concepts apply in blockchain and ''Web 3'' platforms and services. A detailed plan of topics is available on the course web [PDF]. Here is a rough overview:

- **Internet as a distributed system.** DNS/SEC, basic crypto, TLS, CDNs and Web services.
- **Services and RPC**. Reliable RPC; NFS; caching, handling failures.
- **Foundations of mega-services.** Sharding, load spreading, state storage; elastic infrastructure.
- **The datacenter stack**. Linearizability, RPC revisited.
- **Storage abstractions**. Consistency; atomicity; transactions.
- **Causality and logical time.** Weak orderings and their application.
- **Consensus.** Coordination services, CAP, and CALM.
- **Distributed transactions**. Transactional concurrency control and deadlock.
- **Managing trust**. Byzantine failures; blockchains; governance.

Preparation. You should be comfortable with the fundamentals of software systems and networks. You will be programming in Java. You should have some experience in managing concurrency. CPS 310 is sufficient preparation.

Readings. There is no required textbook. Readings for this course consist of tutorials, surveys, and research papers written by researchers in networked systems. The readings are available through the course website. This course does not emphasize reading of research papers: although there are research papers to read, and you are expected to wrestle with a few of them, the purpose of the readings is to support the concepts presented in lectures. The intent is to "distill" core concepts out of the readings, and build a solid grounding in those concepts without getting lost in the details. Exams cover only the material discussed in class and represented on the lecture slides.

Base workload. There are two assigned programming assignments ("labs"), one midterm exam, and a final exam. The final project involves a short written pre-proposal and proposal, a two-minute "lightning" summary presentation, and a final report with a 5-minute presentation. The labs and project may be done individually or in groups of 2-3. In addition, according to need and whim, there may be short quizzes or exercises out of class. Quizzes are averaged into your exam score (time-weighted), and exercises are ungraded but may be testable on the exams.

Calendar. Here are the dates for Spring 2022:

- Jan 26 (W) Lab #1: Reliable RPC
- Feb 2 (W) Project pre-proposal
- Feb 9 (W) Lab #2: Primary/backup replication
- Feb 16 (W) Project proposal
- Feb 24 (Th) Midterm exam on material through Feb 16
- March 3 (Th) Lightning project round
- March 4-14 Spring break
- April 12 (T) Last class: project presentations
- April 25 (M) Project report
- Apr 29 (F) Final exam (7-10 PM)

Late work. Late work receives a penalty of up to 5% per day depending on circumstances. It is much better to do the work and hand it in late than to receive a zero on an assignment.

Assistance. We will provide online assistance through Ed (*edstem.org*): see the course web. Please post your questions there. The instructor and TA hold regular office hours and are available at other times by arrangement. If you are having trouble or just want to talk, please visit!

Attendance and participation. Attendance is not recorded. However, it is expected that you will attend and participate actively. You should prepare questions or opinions about the reading, and I may ask you to speak in class.

Grading. The semester grade is determined from your exam/quiz grades (40%), labs (20%), and project (40%). I may make adjustments of up to half a letter grade for participation/engagement and other factors. Additional information about grading policies, project, and exams is available on the course web.

Policy on collaboration for CPS 512. The Duke Community Standard applies in all aspects of this course: we value your honor and your honesty. Collaboration on lab work and project work is encouraged. Help each other. However, any work you turn in must be your own, and you may be called upon to explain (alone) your choices and approaches in more detail. You may incorporate public software into your work to a reasonable extent, but not so much as to undermine the educational purpose and spirit of the work. You must acknowledge any sources of your ideas and software when they are not your own, and ***you must disclose in advance, without any specific request, any sources you used.*** Do not use code from a student who took the course in a previous semester or who completed related labs in another course or at another school. It is a violation to provide your lab code to anyone else, e.g., by posting on a public or shared repo. All students should understand that we have software that flags copied code with a high degree of certainty and precision. (The tools do not differentiate the makers from the takers.) No assistance or access to external material of any kind is acceptable during exams. If you have any questions about this policy, please ask.