CSE 4/586 Distributed Systems

Murat Demirbas

August 31, 2020

1 Course description

> Distributed systems need radically different software than centralized systems do. —A. Tannenbaum

This course covers the fundamental concepts for the design and analysis of distributed systems & algorithms including: reasoning about distributed programs, handling the lack of global state and global time, achieving distributed consensus in the presence of faults, and designing fault-tolerance for distributed systems. The course also reviews state-of-the-art cloud computing systems with a focus on distributed databases and big data processing systems.

This course will teach the students to identify fundamental distributed systems problems in abstract form, to develop and formally specify distributed algorithms, to reason about correctness and reliability of distributed algorithms, and to understand the trade-offs involved in real-world distributed systems.

There is no required textbook for the course, however ample reading material will be assigned. If you like to use a reference book, you can consider:

- Maarten van Steen Andrew S. Tanenbaum. Distributed Systems. Freely available from `https://www.distributed-systems.net/index.php/books/distributed-systems-3rd-edition-2017/`
- Paolo Sivilotti, Introduction to Distributed Systems, 2005. Available as a pdf download at `http://www.cse.buffalo.edu/~demirbas/CSE586/book.pdf`

2 Topics

1. Introduction; Distributed systems definition and concepts (nodes, network, protocols, faults, concurrency, asynchrony, causality diagrams)
2. Two-phase commit as working example; TLA+ introduction; Coordinating Attack and FLP impossibility results
3. Reasoning about safety properties (next, stable, invariant); Reasoning about liveness properties (transient, ensures, leads-to, variant functions)
4. Consensus problem; Replicated State Machines (RSM); Paxos; Paxos variants; Chain replication
5. Failure detectors; Faults and fault-tolerance
6. Time: logical clocks, vector clocks, hybrid logical clocks; State: distributed snapshots using logical clocks, marker algorithm
7. Datacenter computing; Cloud computing; Cloud services
8. NoSQL databases; CAP theorem; Georeplicated databases; Cassandra; MongoDB; Spanner; Cosmos DB
9. Big data; Big data analytics: MapReduce, Spark, Lambda vs. Kappa architectures
10. Decentralized ledgers and blockchains

The reason we go about learning distributed systems in this way is explained here.

`http://muratbuffalo.blogspot.com/2020/06/learning-about-distributed-systems.html`

3 Logistics

Lectures
Mon/Wed 18:30-19:20 EST
via Zoom (check your email)

Instructor Murat Demirbas, Professor
Office hours: Mon/Wed 20:00-21:00 EST
demirbas@buffalo.edu

4 Grading

- %20 Assignments (4-5 will be assigned)
- %20 TLA+ project
- %10 Quizzes (via Blackboard; one per week, best 10 quizzes will be used)
- %25 Midterm1 (open book)
- %25 Midterm2 (open book, inclusive of Midterm1 content)

486 students are graded on a separate curve than 586 students. Two different curves will be calculated, one for 486 students, one for 586 students.

5 Policies

Late policy: All assignments are due on the day and time posted.

- Homework is due at the beginning of class. They may be submitted up to 1 day late, and late submissions will be scored out of 70 points instead of 100. No homework will be accepted more than 1 day after the assigned due date. Late day starts at the beginning of class when the homework is due and ends 24 hours later.
- Project: No late submission is accepted! Start early, and aim to submit one week before the deadline.

Exams: If you miss an exam because of sickness or similar reasons, visit a physician and obtain a note detailing the period during which you were medically incapable of taking the exam. Notify the instructor immediately via e-mail or telephone (voice mail) if you are going to miss an exam, before the exam takes place unless medically impossible. See the instructor as soon as you return to class. If you miss an exam without a valid excuse, you will receive a zero grade for that exam. No make-up exam will be available without a valid excuse.

Extra work will **NOT** be given to improve your grade.

Academic Integrity: Zero tolerance on cheating!

- 0 in the particular assignment/exam for first attempt.

- Fail the course on the second attempt.
- Team members are equally responsible.
- Consult the University Statements on Academic Integrity: `https://engineering.buffalo.edu/computer-science-engineering/information-for-students/policies/academic-integrity.html`
- Students who do share their work with others are as responsible for academic dishonesty as the student receiving the material. Students are not to show work to other students, in class or outside the class. Students are responsible for the security of their work and should ensure that printed copies are not left in accessible places, and that file/directory permissions are set to be unreadable to others.
- Excuses such as "I was not sure" or "I did not know" will not be accepted. Make sure you go through these questions, and check your answers to these by reading the links on academic integrity.