SWE 622 Spring 2017

"Distributed Software Engineering"

Spring 2017: Wednesdays, 4:30-7:10pm, 134 Innovation Hall

Instructor: Prof. Jonathan Bell
Email: bellj@gmu.edu
Twitter: @_jon_bell_
Office: 4422 Engineering Building; (703) 993-6089
Office Hours: Anytime electronically, Weds 3:30-4:30pm, or by appointment

Prerequisites: SWE Foundation courses or equivalent. In particular, students must have strong Java programming skills.

Overview:

This course conveys key concepts for designing and building distributed software systems. The course is geared towards software engineers that work mostly at the application-level, but need to understand the features and limitations of existing middleware for distributed systems. Additionally, the course covers some research topics related to currently open problems. There will be a special emphasis on understanding the tradeoffs between consistency, availability and partition tolerance (the CAP theorem), and how to decide what tradeoffs to make for specific applications. Some specific topics include: definition and scope of distribution, principles of communication and computation, software architecture of distributed systems, middleware systems, service discovery, mobility, security, and fault tolerance.

Homework:

During the course of the semester, students will use Java and a variety of off-the-shelf tools and libraries to construct a distributed filesystem. By the end of the semester, each student will have built a fault-tolerant and performant in-memory distributed filesystem, with the entire filesystem replicated in the cloud – in Dropbox. Students will get hands-on exposure with distributed systems programming, and with specific technologies such as Redis and Zookeeper. This project is broken up into 6 assignments, and students will have 2 weeks to complete each assignment.

Students must work individually on all homework assignments. We encourage you to have high-level discussions with other students in the class about the assignments, however, we require that when you turn in an assignment, it is only your work. That is, copying any part of another student's assignment is strictly prohibited. You are free to reuse small snippets of example code found on the Internet (e.g. via StackOverflow) provided that it is attributed. If you are concerned that by reusing and attributing that copied code it may appear that you didn't complete the assignment yourself, then please raise a discussion with the instructor.

Textbook:

Distributed Systems: Principles and Paradigms (2nd edition), Andrew S. Tanenbaum, Maarten Van Steen, 2007, ISBN 0-13-239227-5

In Class Activities:

Most lectures will feature interactive activities that support the material being presented. You are strongly encouraged to bring your laptop to class so that you can participate. Your 10% participation grade is based on attendance and participation in in-class activities.

Syllabus and Tentative Schedule

Lecture titles will link to slides.

1. 1/25: Class overview; Introduction to Distributed Systems and key challenges in the field
   Homework 1 assigned: Memory-cached Dropbox filesystem (due 2/8)
2. 2/1: Systems fundamentals; synchronization and consistency problems on a single machine
3. 2/8: Abstractions in distributed systems and their limitations
   Homework 2 assigned: Lock server (due 2/22)
4. 2/15: Consistency: Strict
5. 2/22: Consistency: Relaxed
   Homework 3 assigned: Distributed in-memory caching with Redis; lock caching (due 3/22)
6. 3/1: Transactions
7. 3/8: Agreement & Consensus
   Homework 4 assigned: Replicated lock server with Zookeeper (due 4/12)
8. 3/22: Midterm Exam
9. 3/29: Fault Tolerance & Replication
10. 4/5: Case Studies
    Homework 5 assigned: Handling failover with multiple Redis instances (due 4/26)
11. 4/12: Distributed Computation Models
12. 4/19: P2P and Masterless Systems
    Homework 6 assigned: TBA
13. 4/26: Security Challenges in Distributed Systems
14. 5/3: Review
15. 5/10: Final Exam, normal class time and place.

Grading:

HW: 50%
Class participation: 10%
Midterm exam: 20%
Final exam: 20%

Honor Code:

GMU is an Honor Code university; please see the Office for Academic Integrity for a full description of the code and the honor committee process, and the Computer Science Department's Honor Code Policies regarding programming assignments. The principle of academic integrity is taken very seriously and violations are treated gravely. What does academic integrity mean in this course? Essentially this: when you are responsible for a task, you will perform that task. When you rely on someone else's work in an aspect of the performance of that task, you will give full credit in the proper, accepted form. Another aspect of academic integrity is the free play of ideas. Vigorous discussion and debate are encouraged in this course, with the firm expectation that all aspects of the class will be conducted with civility and respect for differing ideas, perspectives, and traditions. When in doubt (of any kind) please ask for guidance and clarification.

Accommodations for Disabilities:

If you have a documented learning disability or other condition that may affect academic performance you should: 1) make sure this documentation is on file with Office for Disability Services (SUB I, Rm. 4205; 993-2474; http://ods.gmu.edu) to determine the accommodations you need; and 2) talk with me to discuss your accommodation needs.

Privacy:

Students must use their MasonLIVE email account to receive important

University information, including messages related to this class. See http://masonlive.gmu.edu for more information.

Acknowledgements:

This course builds upon several excellent distributed systems courses from other universities:

- MIT's 6.824 (developed by Robert Morris and Frans Kaashoek)
- NYU's G22.3033 (developed by Jinyang Li)
- CMU's 15-440 (developed by David Andersen)
- Columbia's 4995 (developed by Roxana Geambasu)