Northeastern University

CS 6650 Scalable Distributed Systems

Class Times: Saturday 9:10 PM PT on campus Classroom **w10-101 WeWork**

Instructor: Prasad Saripalli

Email: k.saripalli@northeastern.edu

Description: Covers techniques and concepts associated with constructing software that is distributed, reliable, efficient, and extensible; programming multi-threaded applications, communication among objects on different computers, creating a server accessed by multiple clients, using common object design patterns, locating and tailoring components.

Prerequisites: OS and Networks Please study the Primers made available in Canvas under Basics Module before the first class if you did not formally study OS and Networks previously.

Student Learning Goals: Upon successful completion of the course, students should be able to:

- Devise and argue for the correctness of distributed algorithms that perform computational tasks, using algorithmic building blocks including tokens and logical clocks.
- Construct distributed systems to realize distributed algorithms that perform computational tasks using appropriately chosen models of communication.
- Construct distributed systems with various design patterns (e.g. client-server, peer-to-peer).
- Work within a team to construct the specification for a distributed system and to implement and test one or more parts of the system.

Program Outcomes: This course supports the following MSCSS program outcomes:

- An ability to use current techniques, skills, and tools necessary for computing practice.
- An ability to analyze a problem, to assess an articulate its requirements, to design, to implement, and to evaluate a computer-based system.
- An ability to participate effectively in team projects.
- Recognition of the need for and ability to engage in self-directed learning and continuing professional development.

Student Learning Goals: This course supports the following NEU student learning goals:

- Inquiry and Critical Thinking — Students will acquire skills and familiarity with modes of inquiry and examination from diverse disciplinary perspectives, enabling them to access, interpret, analyze, quantitatively reason, and synthesize information critically.
- Communication/Self-Expression — Students will gain experience with oral, written, symbolic and artistic forms of communication and the ability to communicate with diverse audiences. They will also have the opportunity to increase their understanding of communication through collaboration with others to solve problems or advance knowledge.

Textbook: We will use the following required textbook for this class:

- *Distributed Systems: Concepts and Design. George Coulouris, Jean Dollimore, Tim Kindberg. Addison Wesley. Fifth Edition. http://www.cdk5.net/wp/*
    -

Other textbooks you may find useful as references for this class are included below, although these are not required.

- Java Concurrency in Practice - by Brian Goetz et al. 2019 for Coding projects
- *Distributed Systems: Principles and Paradigms, Andrew Tanenbaum and Maarten van Steen, Prentice Hall.*
- *Operating Systems, Principles and Practice. Thomas Anderson, Michael Dahlin.*

Topics Covered

- Distributed system basics
- Models of inter process communication
- Representation and correctness of distributed programs
- Time on Distributed Systems – physical and logical time
- Consistency
- Distributed mutual exclusion and tokens
- Distributed snapshots and global state
- Fault tolerance
- Consensus

Tentative Schedule - Please note that this schedule may be revised as we progress through the quarter. I will use Canvas to specify the topics for every week, including reading assignments, foils for the week, and projects specifications.

Week	Topic	Exam	Projects
0	***Introduction***		
1	*Introduction, Architecture and Communication Infrastructure Review*		Project #1 Assigned
2	*Processes, Threads, RPC*		
3	*Time & Synchronization*		Project #2 Assigned
4	*Logical Clocks, Group Communication & IP Multicasting*		
5	*Distributed Mutual Exclusion, Network File Systems*	Mid-term Exam	Project #3 Assigned
6	*Distributed File Systems*		
7	*Consensus, Paxos*		Project #4 Assigned
8	*Distributed Parallel Frameworks, MapReduce, Spark, MPI*		
9	*Byzantine Fault Tolerance, P2P Networks*		

	10	*Distributed Memory*	*Shared*		In---class demos of Final Project
Finals	Week			Final Exam	

Online Materials: Lectures, homework assignments, supplemental materials etc., will be shared via Canvas. Unless otherwise indicated, assignments should be submitted through Canvas. Submission links will be provided as they are assigned. **Grading Criteria:**

Projects (4)	30%
Final Project	13 %
Homeworks	25%
Midterm exam	15%
Final exam	15%
Attendance	2%

One project may be submitted up to 48 hours late without penalty. Additional late submissions (beyond one) and submissions more than 48 hours late will not be accepted without mitigating justification discussed with the instructor before the deadline. Overall course grades (decimal) will be calculated from the weighted average of assignment and exam percentage scores using the NEU grading system. The above grading criterion is approximate, and instructor will use a curve to grade for the final grading as needed, adjusting the above % grade credit per how the class performs through the semester. Number of homeworks may vary (+ or – one).
Example grading conversion https://course.ccs.neu.edu/cs5500sp17/grades.html

Teaching Assistants
Your TA are
alladi.p@northeastern.edu Pavan Sai Kumar Alladi
mahluza.m@northeastern.edu Mbade Mbongeni Mahluza
budihalprasad.a@northeastern.edu Anirudha
revadi.p@northeastern.edu Prateek Srigiri Revadi
beura.g@northeastern.edu Gourav Beura
There will be weekly sessions by TAs per assignment or project via Zoom to help answer your questions, one session per TA every week.

Practice your Java and Java Networking skills. We mainly use Java for all the work.
You will need to form teams of 3-4 classmates to work on your Final Projects so reach out and get started. ***Welcome to Distributed Systems!***