

15-440: Distributed Systems

Units: 12

Prerequisites: A grade of “C” or better in 15-213, Introduction to Computer Systems

Course website: <https://web2.qatar.cmu.edu/~mhhammou/15440-f19/index.html>

Course overview

15-440 is an introductory course in distributed systems. The emphasis will be on the techniques for creating functional, usable, and high-performing distributed systems. To make the issues more concrete, the class includes several multi-week projects requiring significant design and implementation.

The goals of this course are twofold: First, students will gain an understanding of the principles and paradigms that underlay distributed systems, such as communication across networks, concurrency, synchronization, consistency and fault-tolerance. Second, students will gain practical experience in designing, implementing, and debugging real distributed systems.

The major themes this course will teach include process distribution, communication, naming, abstraction and modularity, concurrency, scheduling, resource sharing, locking, consistency and replication, failure handling, distributed programming models, distributed file systems, virtualization, and the use of instrumentation, monitoring and debugging tools to solve problems at large-scale. As the creation and management of software systems are fundamental goals of any undergraduate systems course, students will design, implement, and debug large programming projects. Students will learn some of today’s most popular distributed systems, such as Google File System, MapReduce and PowerGraph.

Logistics

Prof. Mohammed Hammoud

mhhammou@qatar.cmu.edu

Room 1006

4454-8506

Office hours: Monday, 10:30 - 11:59 AM

TA. Zeinab Khalifa

zfk@andrew.cmu.edu

Room 1004

4454-8496

Office Hours: Sunday, 2:00 PM – 3:00 PM,
Tuesday, 2:00 PM – 3:00 PM

Class hours

Lectures

Mondays and Wednesdays, from 9:00 AM to 10:20 AM, in Room 3044

Recitations

Thursdays, from 9:00 AM to 9:50 AM, in Room 3044

Course objectives

Distributed systems combine the computational power of multiple computers to solve complex problems. The individual computers in a distributed system are typically spread over wide geographies, and possess heterogeneous architectures and operating systems. Hence, an important challenge in distributed systems is to design system models, algorithms and protocols that allow computers to communicate and coordinate their actions over heterogeneous networked computers so as to solve large-scale problems.

Our aim in this course is to introduce you to the area of distributed systems. You will examine and analyze how a set of networked computers can form a functional, usable and high-performing distributed system.

The course has three major goals:

1. To learn the principles, architectures, algorithms and programming models used in constructing distributed systems.
 2. To examine state-of-the-art distributed systems, such as Google File System.
 3. To design, implement and debug real distributed systems.
-

Learning outcomes

The course encompasses two main learning outcomes:

- Students will identify the core concepts of distributed systems; that is, the way in which several machines can be orchestrated to correctly solve complex problems in an efficient, reliable and scalable manner.
- Students will examine how existing systems have applied the core concepts of distributed systems, and will additionally apply such concepts in developing sample systems.

Understanding the core concepts of Distributed Systems

Students will learn the core concepts that comprise any distributed system. They will recognize the system constraints, trade-offs and appropriate techniques for building distributed systems that best serve the computing needs of different classes of applications. In particular, students will learn the following concepts:

- Access & Location Transparency
- Task Parallelization
- Fault-tolerance

Access & Location Transparency

Exposing the capabilities of machines, yet hiding their details is one of the first steps in designing distributed systems. Such systems penetrate economies and masses which transparently leverage their powers. For instance, in the Internet, which is a successful distributed system, a simple browser interface will allow you to explore information scattered over wide-geographies. In this

course, students will examine how to abstract data and machine locations (which may reside at different physical places) as well as data and machine replications.

Specifically, students will study the following topics:

- **Processes and Communication:** Students will explain and contrast the communication mechanisms between different processes and systems.
- **Naming:** Students will identify why entities and resources in distributed systems should be named, and examine the naming conventions as well as some naming resolution mechanisms.

Task Parallelization

Traditional algorithms that work on a single processor are inefficient - or even fail to work - in a system where multiple machines are working in parallel. In distributed systems, problems/jobs can be solved using parallelization. Generally a job is split into multiple tasks, and all tasks are executed in parallel on different machines. The tasks may access common resources, such as data contained in a shared file. Consequently, two main challenges emerge. First, we ought to ensure that the concurrently running tasks are coordinated and synchronized in a manner that correctly achieves the job's goal. Second, we can potentially replicate and place resources across multiple computers in a way that allows tasks to access them more effectively.

Specifically, students will study the following topics:

- **Concurrency and Synchronization:** Students will identify issues on how to coordinate and synchronize multiple tasks in a distributed system.
- **Caching, Replication and Consistency:** Students will understand how replication and caching of resources can optimize performance and scalability, as well as examine various models that allow maintaining consistency of replicated and cached data.

Fault-tolerance

In distributed systems, a failure of a single or a part of a computer (or what is known as partial failure) is very likely. If such a failure is not tolerated, the whole system might come to a grinding halt or result in a random and unpredictable behavior. Students will learn how to avoid and recover from partial failures, a concept referred to as fault-tolerance.

Practical Application of the State-of-the-Art Distributed Systems:

Students will also learn how to apply principles of distributed systems in a real-world setting. In particular, they will learn the following topics:

- **Distributed Frameworks:** Students will learn some of the distributed frameworks such as MapReduce, GraphLab and Pregel. These distributed frameworks allow developers to easily program distributed problems/algorithms, while ensuring correctness, fault-tolerance and efficiency.
- **Distributed File Systems:** Students will learn how a file can be striped and placed anywhere in a distributed system (or what is referred to as distributed file system), yet be accessed transparently - as if it is a local file. They will examine how to apply distributed system principles to ensure transparency, consistency and fault-tolerance in distributed file systems.
- **Virtualization:** Students will learn the concept of system virtualization, where a state of a computer is abstracted from the underlying hardware. This allows masking the heterogeneity of the machines that comprise a distributed system, besides increasing overall system utilization and security.

Textbooks

The primary textbooks for this course are:

- Andrew S. Tannenbaum and Maarten Van Steen, *"Distributed Systems: Principles and Paradigms"*, Second Edition, Pearson, 2007.
- George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair *"Distributed Systems: Concepts and Design"*, Fifth Edition, Addison Wesley, 2011.

In addition, we recommend the following text books:

- Randal E. Bryant and David R. O'Hallaron, *"Computer Systems: A Programmer's Perspective"*, Prentice Hall, 2003.
 - Tom White, *"Hadoop: The Definitive Guide"*, Second Edition, O'Reilly Media, 2010.
-

Organization

The participation of students in the course will involve five forms of activities:

- Attending lectures and recitations
 - Solving assignments (including writing and reading assignments)
 - Solving large programming projects
 - Taking exams and quizzes
 - Participating in class discussions
-

Assessment

Final Grade Assignment and Assessment methods

Each student will receive a numeric score with a corresponding letter grade, based on a weighted average of the following:

Projects

The projects will count for a total of 40% of your final score. There will be 4 projects throughout the course. All projects are individual projects (i.e., no team can work on the same project). The first project is worth 15% and the last three projects are worth 7.5%, 10% and 7.5% respectively. *It's important to note that Project 2 heavily relies on completing Project 1, and therefore, it's strongly advised to finish P1 early to avoid complications with P2 (If you have not completed P1, you will not be given any solution files to work on P2!).*

You are encouraged to submit the projects on time. For all projects except the final one, the following rules apply. If you submit one day late, we will deduct 25% of the project score as a penalty. If you submit two days late, 50% will be deducted. The project will not be graded (and you will receive a zero score on it) if you are more than two days late. However, there is a grace-days quota for projects. In particular, you will be given 3 grace days for all projects, except for the final one. You can use the grace days as needed. For instance, you can submit your first project three days late and still receive no penalty. In this case, you

will be penalized starting from the 4th day after the deadline. Furthermore, when you consume all your grace days, you will be left with no grace days for the rest of the projects.

Note that the final project is unique in two aspects. First, you cannot use grace days for it. As such, if you are left with some grace days before the final project, you will lose them all. Hence, plan how to utilize your grace-days quota judiciously. Second, there will not be any penalty system for this project either. More precisely, if you are one day late in submitting the project, it will not be graded and you will receive a zero score on it.

Exams

There will be two in-class exams – midterm and final – which together will count for 30% of your final score. The midterm is worth 10% and the final is worth 20%. *Both exams are open-book. That is, you are allowed to bring your textbooks, slides, and other supporting documents of your own. Any electronic equipment is not allowed, however.*

Problem Sets

There will be 5 assignments that will test you on problem analysis and solving skills. These assignments will altogether carry 15% of your final score. *At the end of the semester, the Problem Set with the lowest score will be dropped.*

Quizzes

There will be 2 quizzes, which together will count for 10% of your final score. These quizzes are meant to test your understanding and preparation for the concepts covered throughout the course.

Class-Recitation Participation and Attendance

Your attendance of both, classes and recitations, as well as your participation in discussions during presentations will count for 5% of your final score.

To this end, the below table shows the breakdown of the five forms of activities that the course involves, alongside the quantity and the overall weight of each activity. Take into account that small differences in scores can make the difference between two letter grades. Letter grades will be determined by absolute standards. The total score will be plotted as a histogram. Cutoff points are determined by examining the quality of students' work on the borderlines. Individual cases, especially those near the cutoff points may be adjusted upward or downward based on factors such as attendance, class participation, improvement observed throughout the course, exam performance, and special circumstances.

Type	#	Weight
Projects	4	40%
Exams	2	30%
Problem Sets	5	15%
Quizzes	2	10%
Class Participation and Attendance	42	5%

Because of the importance of understanding both the theoretical and hands-on elements of the class, students must pass both components of the course (projects as one component, and exams, quizzes and problem sets as the second) in order to receive a passing grade for the course. This does not affect the actual letter grade assignment unless one of the components is not completed to a passing standard.

Getting Help

For urgent communication with the instructor and the teaching assistant, it is best to send an email (preferred) or give a phone call. If you want to talk to any of them in person, remember that their posted office hours are merely nominal times when they guarantee that they will be in their offices. You are always welcome to visit them outside of their office hours if you need help or want to talk about the course.

We ask that you follow a few simple guidelines. The instructor normally works with his office door being open. Whenever the office door is open, he welcomes visits from students. However, if his office door is closed, this means that he is busy with meetings or phone calls, thus prefers not to be disturbed.

We will use the course webpage as the central repository for all information about the class. Through the webpage, you can:

1. Obtain copies of any handouts or assignments. This is especially useful if you miss a class or lose a document.
2. View announcements that relate to the course.
3. Find links to any electronic data you need for your assignments.
4. Read clarifications and changes made to any assignments, schedules, or policies.

Lastly, you can use Piazza for asking questions and receiving answers without all the emails! Posting your questions on Piazza will help the whole class benefit and will certainly avoid redundancy. Find our class Piazza page at: <https://piazza.com/qatar.cmu/fall2019/15440>

Policies

Working Alone on Assignments/Projects

Assignments/projects that are assigned to students should be performed individually. This course does not include any team projects or assignments.

Handing in Assignments/Projects

All assignments/projects are due at 11:59PM (one minute before midnight) on the specified due date. All hand-ins are electronic and should be submitted using Gradescope
<https://www.gradescope.com/courses/61075>

Making up Exams, Assignments and Projects

Missed exams, assignments and projects can be made up on a case by case basis, but only if you make prior arrangements with the instructor. However, you should have a good reason for doing so. You need a written consent from the instructor for making up exams, assignments or projects. It is your responsibility to get your projects and assignments done on time. Be sure to work far enough in advance to avoid unexpected problems, such as illness, unreliable or overloaded computer systems, etc.

Appealing Grades

After each exam, assignment, and/or project is graded, you have 7 calendar days to appeal your grade. All your appeals should be provided in writing. If after appealing you are still not satisfied, please visit the instructor. If you have questions about an exam, an assignment or a project grade, please visit the instructor directly.

Cheating

Each project or assignment must be the sole work of the student turning it in. Projects and assignments will be closely monitored, and students may be asked to explain suspicious similarities with any write-up or piece of code available. The following are guidelines on what cheating is and is not:

What is cheating?

1. Sharing code or other electronic files: either by copying, retyping, looking at, or supplying a copy of a file.
2. Sharing written assignments: either by re-writing, looking at, or supplying a copy of an assignment.

What is NOT cheating?

1. Clarifying ambiguities or vague points in class handouts.
2. Helping others use the computer systems, networks, compilers, debuggers, profilers, or other system facilities.
3. Helping others with high-level design issues.
4. Helping others debug their codes.

Consequently, be aware of what constitutes cheating (and what does not) when interacting with your colleague students. Same rules of cheating as above apply when collaborating with other students. In short, you cannot share written assignments, code, and/or other electronic files with other students. If you are unsure, ask the teaching staff.

Finally, be sure to store your work in protected directories. The penalty for cheating is severe, and might jeopardize your whole career as a student - cheating is not worth the trouble. By cheating in the course, you are cheating yourself; the worst outcome of cheating is missing an opportunity to learn. Besides, you will be removed from the course and assigned a failing grade. We also place a record of the incident in your permanent university profile.

Health & Wellness

Learning Disabilities

Carnegie Mellon University is committed to providing reasonable accommodations for all persons with disabilities. To access accommodation services you are expected to initiate the request and submit a Voluntary Disclosure of Disability Form to the office of Health & Wellness or CaPS-Q. In order to receive services/accommodations, verification of a disability is required as recommended in writing by a doctor, licensed psychologist or psycho-educational specialist. The office of Health & Wellness, CaPS-Q and Office of Disability Resources in Pittsburgh will review the information you provide. All information will be considered confidential and only released to appropriate persons on a need to know basis.

Once the accommodations have been approved, you will be issued a Summary of Accommodations Memorandum documenting the disability and describing the accommodation. You are responsible for providing the Memorandum to your professors at the beginning of each semester.

For more information on policies and procedures, please visit [this document](#).

Taking Care of Yourself

Do your best to maintain a healthy lifestyle this semester by eating well, exercising, getting enough sleep, and taking some time to relax. This will help you achieve your goals and cope with stress.

All of us benefit from support during times of struggle. You are not alone. There are many helpful resources available on campus and an important part of the college experience is learning how to ask for help. Asking for support sooner rather than later is often helpful.

If you or anyone you know experiences any academic stress, difficult life events, or feelings like anxiety or depression, we strongly encourage you to seek support. Counseling and Psychological Services (CaPS-Q) is here to help: call 4454 8525 or make an appointment to see the counselor by emailing student-counselling@qatar.cmu.edu. Consider reaching out to a friend, faculty, or family member you trust for help.

If you or someone you know is feeling suicidal or in danger of self-harm, call someone immediately, day or night, at 5554-7913.

If the situation is life threatening, call 999.

Class Schedule

Please refer to below schedule for the tentative schedule for the class. The schedule indicates the project and the assignment activities as well. Any changes will be always announced and reflected on this webpage.

Week	Session	Date	Topic	Class Type	Projects	Assignments
1	1	26-Aug	Introduction	Lecture		
	2	28-Aug	Networking - Part I	Lecture		
	3	29-Aug	Intro to Java	Recitation		Start PS1
2	4	2-Sep	Networking - Part II	Lecture		
	5	4-Sep	Remote Procedure Calls - Part I	Lecture		
	6	5-Sep	Socket Programming in Java	Recitation	P1 Out	
3	7	9-Sep	Remote Procedure Calls - Part II	Lecture		PS1 Due
	8	11-Sep	Architectures - Part I	Lecture		
	9	12-Sep	Project 1 - Part I	Recitation		PS2 Out
4	10	16-Sep	Architectures - Part II	Lecture		
	11	18-Sep	Naming - Part I	Lecture		
	12	19-Sep	Project 1 - Part II	Recitation		
5	13	23-Sep	Naming - Part II	Lecture		PS2 Due
	14	25-Sep	Naming - Part II Cont'd & Quiz Review	Lecture		
	15	26-Sep	Quiz (1)	Quiz		
6	16	30-Sep	Synchronization - Part I	Lecture		PS3 Out
	18	2-Oct	Synchronization - Part II	Lecture	P1 Due	
	19	3-Oct	Synchronization - Part II (Cont'd)	Lecture	P2 Out	
7	20	7-Oct	Synchronization (Cont'd) & Project 2	Lecture/Recitation		
	21	9-Oct	MIDTERM	Exam		

	22	10-Oct	Synchronization - Part II (Cont'd)	Lecture		PS3 Due
		14-17Oct	Fall Break			
8	23	21-Oct	Synchronization - Part II (Cont'd)	Lecture		
	24	23-Oct	Synchronization - Part III	Lecture		
	25	24-Oct	Synchronization - Part III (Cont'd)	Lecture		
9	26	28-Oct	MPI	Lecture	P2 Due	
	27	30-Oct	Distributed Frameworks	Lecture	P3 Out	
	28	31-Oct	MPI, MPI (CR)	Recitation		
10	29	4-Nov	Distributed Frameworks - Part II	Lecture		PS4 Out
	30	6-Nov	Distributed Frameworks - Part III, Caching - Part I	Lecture		
	31	7-Nov	Distributed Frameworks - Part IV	Lecture		
11	32	11-Nov	Distributed Frameworks (Concluded)	Lecture		
	33	13-Nov	QUIZ II & Caching (Intro)	Lecture		PS4 Due
	34	14-Nov	MapReduce - Part I	Recitation		
12	35	18-Nov	Caching - Part I	Lecture		
	36	20-Nov	Caching - Part II	Lecture	P3 Due	PS5 Out
	37	21-Nov	Caching - Part III	Lecture	P4 Out	
13	38	25-Nov	Replication - Part I	Lecture		
	39	27-Nov	Replication - Part II	Lecture		
	40	28-Nov	Fault Tolerance	Lecture		PS5 Due
14	41	2-Dec	Final Review	Recitation	P4 Due	
	42	TBA	FINAL EXAM	Exam		