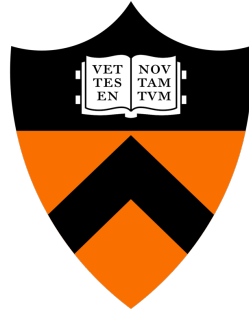


Distributed Systems: Intro and Course Overview



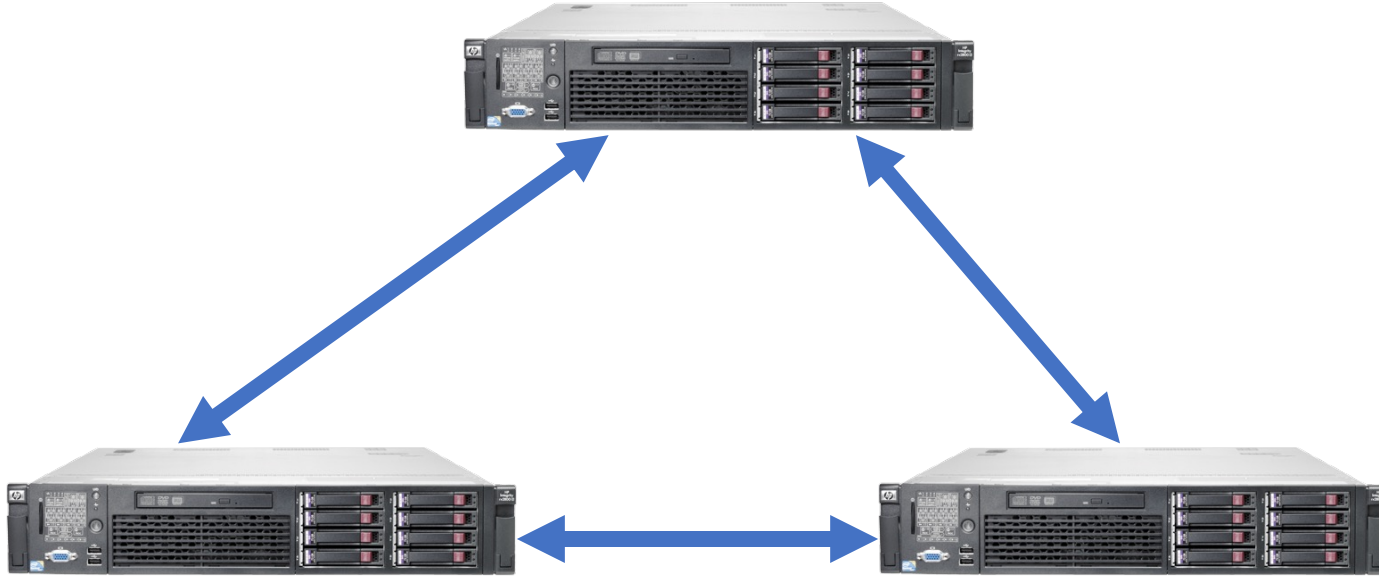
COS 418/518: Distributed Systems

Lecture 1

Fall 2026

Jialin Ding, Mike Freedman

Distributed Systems, What?



- 1) Multiple computers
- 2) Connected by a network
- 3) Doing something together

Distributed Systems, Why?

- Or, why not 1 computer?
 - Failure
 - Limited computation/storage/...
 - Physical location

Distributed Systems, Where?

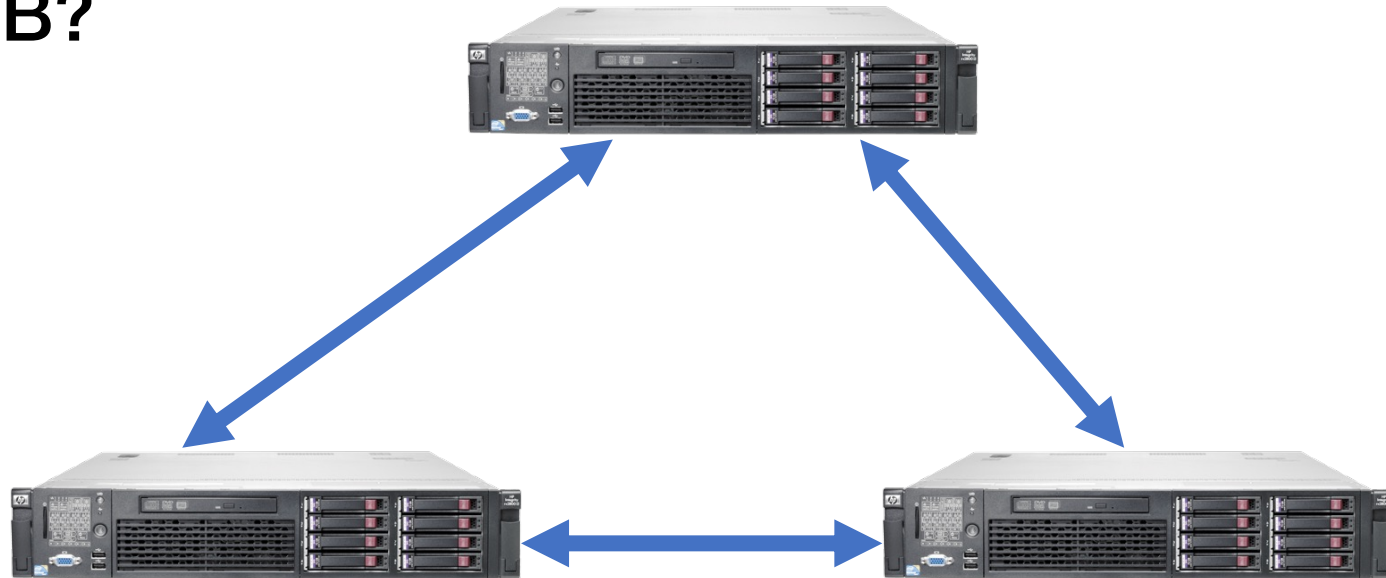
- Web Search (e.g., Google, Bing)
- Shopping (e.g., Amazon, Walmart)
- File Sync (e.g., Dropbox, iCloud)
- Social Networks (e.g., Facebook, Twitter)
- Music (e.g., Spotify, Apple Music)
- Ride Sharing (e.g., Uber, Lyft)
- Video (e.g., Youtube, Netflix)
- Online gaming (e.g., Minecraft, Roblox)
- AI Models (OpenAI, Anthropic)

Distributed Systems Goal

- Service with higher-level abstractions/interface
 - e.g., file system, database, key-value store, programming model, ...
- Hide complexity
 - Scalable (scale-out)
 - Reliable (fault-tolerant)
 - Well-defined semantics (consistent)
- Do “heavy lifting” so app developer doesn’t need to

Fundamental Challenges in Distributed Systems

- How can you tell if a message reached another machine?
- How can you ensure a message doesn't reach another machine twice?
- How can you tell an event happened first at machine A or machine B?



Challenges in Scaling Distributed Systems

- Scale computation across many machines
 - How do you split the work?
 - Example: MapReduce
- Scale storage across many machines
 - How can you figure out where a piece of data is stored?
 - Examples: sharding, consistent hashing (Chord)

Challenges in Fault-Tolerant Distributed Systems

- Retry on another machine
 - How do you take a snapshot of a distributed system?
 - Example: Chandy-Lamport snapshots
- Maintain replicas on multiple machines
 - How can you ensure that all machines agree on the same state of the world?
 - Example: Paxos, RAFT

Challenges in Consistent Distributed Systems

- If machine A writes some data, when will machine B be able to read it?
 - Examples: eventual consistency, linearizability
- What if you want to write multiple pieces of data at the same time?
 - Examples: transactions, atomic commit, serializability

Additional Topics

- **Blockchain as a distributed system**
- **Distributed systems for AI**

Learning Objectives

- Reasoning about concurrency
 - Reasoning about failure
 - Reasoning about performance
-
- Building systems that correctly handle concurrency and failure
 - Knowing specific system designs and design components

Course Logistics

- Lectures and exams are for both COS 418 and 518
- For COS 418 only: Friday precepts and programming assignments
- For COS 518 only: Tuesday precepts
- Refer to course website and Ed (both are linked from Canvas)

Lectures

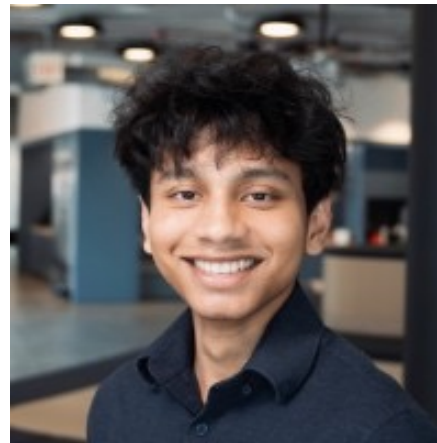
- **Tuesdays/Thursdays 10:40am-11:30am**
 - Slides posted on course website just before class
- **Schedule posted on course website**

Precepts

- **For 418: three precepts on Fridays**
 - Slides will be posted after all precepts
 - Helps with assignments and/or reinforces lecture material
 - First precepts this Friday
- **For 518: one precept on Tuesdays 1:20-2:10pm**
 - Attendance is mandatory
 - Do assigned reading before the precept
 - First precept is next Tuesday

Course Staff

- Instructors: Jialin Ding, Mike Freedman
- Grad TAs:
 - Raye Kimmerer (P1)
 - Sachin Konan (P2)
 - Anupta Argo (P3)
 - Zhuofu Chen
 - Karan Tandon
- Lab TA: TBD
- Office hours posted on Ed



Grading (for COS 418)

- **Assignments (50%)**
 - Five assignments 10% each
- **Two exams (50%)**
 - Midterm 25%
 - Final 25%

Grade Cutoffs

- We do not anticipate a curve up
- We will not curve down
- A [93, 100]
- A- [90, 93)
- B+ [87, 90)
- B [83, 87)
- ...

Programming Assignments

- Reinforce / demonstrate all learning objectives!
- 1: “MapReduce” in Go
- 2: Distributed Snapshots
- 3: Raft Leader Election
- 4: Raft Log Consensus
 - Assignment #4 builds on your Assignment #3 solution, i.e., you can't do #4 until your own #3 is working!
- 5: Key-Value Storage Service

Programming Assignments

- All are individual assignments
 - Write your own code. Do not copy another person's program.
- Due dates posted on course website schedule
- Assignment submission is through Github. Look out for instructions on Ed about how to get started.
- We will give you all the tests
 - Non-determinism for later tests though
 - Each failed test run => lose 50% of points for a test
- Grading script emails your grade
 - We use exactly this grade

Programming Assignments- Late Policy

- Late policy
 - G = Grade you would have earned if turned in on time
 - 1 day late $\Rightarrow 90\% * G$
 - ...
 - > 5 days late $\Rightarrow 50\% * G$
- 3 “free” late days
 - We assign them at the end of the semester to maximize your score
 - Used in 1 day granularity
 - Cannot use for last assignment (Deans date)
- For assignment grading issues, reach out to Jialin/Anupta

AI policy for programming assignments

- You can use any AI tool you want
- However, there will be assignment-specific questions on the exam (for 418 students only)

Programming Assignment Statistics

- Self Reported Hours Spent on Assignments: Median [Min-Max]
- 1-1: 2 [1-6]
- 1-2: 3 [1-8]
- 1-3: 4 [1-10]
- 2: 6 [2-15]
- 3: 12 [5-29]
- 4: 30 [10-100+]
- 5: 14 [3-25]

Exams

- In person exams
 - 2.5-3 hours
 - Closed book
- Midterm
 - Registrar will schedule during midterm week
 - Likely Thursday, October 15, 7:30pm-10pm
- Final
 - Saturday, December 19, 8:30am–11:30am

Exam Content

- Will primarily cover material from lectures
 - Friday precepts will reinforce the material, but no question will come exclusively from precepts
- For 418 only: additional questions about programming assignments
 - Will not ask you to write code, may ask you to interpret/debug code
- For 518 only: additional questions about assigned papers, precept discussions
- Final exam only covers material from second half of course

Exam Preparation

- We will release practice midterm and final questions around a week before each exam
- Lecture before midterm (likely Thursday, Oct 10) will be a review session

General help

- Office hours
 - 2 hours of OH each weekday
 - Schedule posted on Ed
- EdStem
 - Course staff will aim to respond within one business day

Course Overview Conclusion

- Attend lecture, attend precept
- Start programming assignments early
- Distributed systems stuff starts Tuesday

