CPSC 426/526

Course Introduction

Ennan Zhai

Computer Science Department
Yale University
Welcome

• My name: Ennan Zhai

• Associate Research Scientist at CS department

• Research interests:
  - Large-scale distributed systems, e.g., cloud systems
  - Security, privacy and applied crypto
  - Programming language and verification
  - See more: http://www.cs.yale.edu/homes/zhai-ennan/
Have you used distributed system?
Have you used distributed system?
Have you used distributed system?
Have you used distributed system?
Have you used distributed system?
Have you used distributed system?
What is a distributed system?

You know you have a **distributed** system when the **crash** of a computer you’ve never heard of stops you from getting any work done.

– Leslie Lamport
What is a distributed system?

- A system of multiple computers (nodes) communicating over a network
What is a distributed system?

- A system of multiple computers (nodes) communicating over a network

Some following questions:
- What is a decentralized system?
- What is a cloud system?
- What is a centralized distributed system?
Why should I care?

• Real life is full of distributed systems
Why should I care?

- Real life is full of distributed systems
Why should I care?

- Real life is full of distributed systems

- Understand what’s underneath distributed systems
  - How do they work? What are Pros. and Cons?
  - Technologies: MapReduce, DHT, etc.
Why should I care?

• Real life is full of distributed systems

• Understand what’s underneath distributed systems
  - How do they work? What are Pros. and Cons?
  - Technologies: MapReduce, DHT, etc.

• Understand the underlying principles
  - How do you make your systems scalable and robust?
  - Smart designs and very difficult problems
Why should I care?

- Use correct way to build your own systems
  - How to build a workable system?
  - Need to scale, be efficient, avoid failures, etc.

- Gain practical experience with technologies
  - The best way to understand it is to build one yourself
  - In our course, you will build some workable systems

- Anticipate what’s possible in the future
What can I learn?

How I feel when my code works.
What can I learn?

• The principle in distributed systems
  - Cover various systems, including reliability and security
  - Go through their design details

• Examine a number of real, past and present systems
  - How they work
  - How and why they succeeded or failed

• Solidify this knowledge by applying it to build workable distributed systems
Lecture Roadmap

• Introduction
• Course Logistics
• Overview
# Course Staff

<table>
<thead>
<tr>
<th>Name</th>
<th>Ennan Zhai</th>
<th>Peizhen Guo</th>
</tr>
</thead>
<tbody>
<tr>
<td>Role</td>
<td>Instructor</td>
<td>TA</td>
</tr>
<tr>
<td>Email</td>
<td><a href="mailto:ennan.zhai@yale.edu">ennan.zhai@yale.edu</a></td>
<td><a href="mailto:peizhen.guo@yale.edu">peizhen.guo@yale.edu</a></td>
</tr>
<tr>
<td>Office hour</td>
<td>Wed. 3-4pm</td>
<td>TBD</td>
</tr>
<tr>
<td>Office</td>
<td>AKW 210</td>
<td>HLH17 Room 228</td>
</tr>
</tbody>
</table>

Please send your email to cs426ta@cs.yale.edu, if you have any questions. Both Peizhen and I will see your email.
Prerequisites

• Good C++ programming skills
  - No need to be a ‘hacker’
  - Need a lot of programming experience
  - DO NOT ASK TA TO DEBUG FOR YOU

• Some familiarity with Linux command line
  - Need to remotely access the Zoo
  - Need to use git repository in the Zoo
Prerequisites

• Basic network knowledge
  - Know how network works, e.g., DNS and routing
  - How nodes communicate with nodes

• Basic system knowledge
  - Responsibilities of software and hardware
  - How does a software system run on your computer
Course Workload

• Midterm (15%) and final (15%)
  - In class and close-book
  - Test your understanding to our lectures

• Four Labs (15% * 4 = 60%)
  - Programming intensive labs
  - Lab4 needs you to define spec and present demo

• Student presentations (10%)
  - Four groups, each for one lecture
  - I will assign papers with different topics to different groups
Midterm and Final

• Midterm (Oct 16) and Final (Nov 29)
  - Final exam will not cover the knowledge before midterm
  - No multiple choices, code writing or complex computation
About Regular Labs

- Three regular labs
  - Build small but workable system, named Peerster
  - Follow my spec posted on our website
  - Bakeoff version and final version
  - No group
  - Test against my reference Peerster

[Image: An iceberg representing output of the program and actual code]
About Regular Labs

• Labs will be built upon previous labs
  - Do not keep deep bugs
  - Make your code modular, flexible and clean

- Lab4: Your own system
- Lab3: File sharing and downloading
- Lab2: NAT and Messaging
- Lab1: Gossip
WHEN CODE IS LITTLE

CODE AFTER SOME TIME
About Lab4

- Define your topic and specification
- Build your lab4 on your peerster
- We will offer some alternative topics to you
- 1-3 students in one group
- Graduate students MUST take research topics
- Present your lab4 in the last two classes
Demo for Lab4
Lab4 may produce papers :)  
- I have advised many Yale students for their research projects  
- We published papers together based on these projects in the top-tier computer conferences, e.g., NSDI  
- If you are interested in publishing papers, work hard on lab4  
- I can offer you concrete topics to work on  
- I can answer any question during your implementation  
- The result will be submitted to top-tier conferences
Policies: Collaboration

- Can we work on labs together? No
- Can I discuss the assignment with others in general terms? Yes
- Can I use code I copied from the web? No
- Can I ask questions about labs? Yes
- I happened to leak my password, and XYZ happened to find it. Will I be penalized for this? Yes
Using Git Repo.

• Be familiar with using git repo
  - Track your version history
  - Submit your labs
  - Recover previous workable versions
Penalty

• For all the four labs
  - The late penalty is 10% per day for the most 3 days
  - No credit for the case more than 3 days late
Expected ‘Payoff’

• You will acquire a set of skills and knowledge that is very high demand right now
  - At Google, Microsoft and at many other places
  - They will be useful in interviews

• Gain interesting insights to computer systems
  - We can publish the paper together
  - Will be useful to your career

• You will have a good basis for other courses
**Downside**

- This is a *bleeding edge and tough* course
  - Each topic needs you to spend a lot of time
  - You have two exams and four coding intensive labs

- Sth. in the course will result in hair loss
  - Debugging distributed code is hard!
  - Deeply understanding papers I signed is hard!
What I do not care

- I do not care whether you attend every class
- I do not care your programming style
- I do not care whether you come to my office
- I can give you high grades as long as you show me you learned a lot and produce great results
What I care

• I care whether you deeply understand what I teach
• I care whether you ask good questions in our class
• I care whether you build a robust system
Lecture Roadmap

• Introduction
• Course Logistics
• Overview
Class Schedule

https://zoo.cs.yale.edu/classes/cs426/2017/sched

• Student presentation days:
  - Sep 25, Oct 4, Oct 23 and Nov 8
  - We need four groups

• Lecture classifications:
  - Lec 2-9: Fundamental stuff in distributed systems
  - Lec 10-14: Security and privacy in distributed systems
  - Lec 15-20: Heading off failures in distributed systems
Next Lecture

• In the lecture Sep 1, I will cover:
  - Basic network knowledge (e.g., socket)
  - The first successful decentralized messaging system
  - UseNet and gossip protocol