THE UNIVERSITY OF CHICAGO

MPCS: 52040
Distributed Systems

Kyle Chard

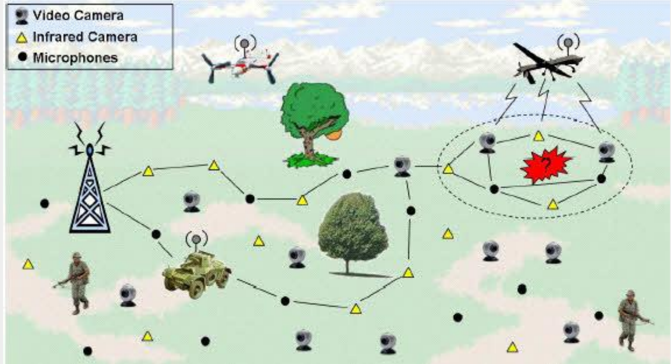
chard@uchicago.edu

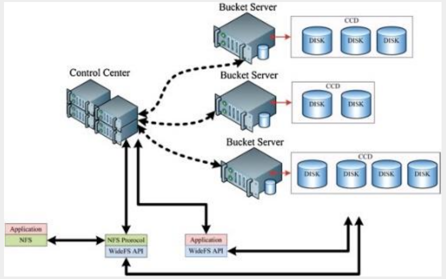
Welcome – Distributed Systems

*A distributed system is a **network** that consists of **autonomous computers** that are connected using a **distribution middleware**. They help in sharing different resources and capabilities to provide users with a single and integrated coherent system*

NETFLIX

BitTorrent™





Courtesy of Dante Aubert

Introduction

Making all research data accessible, discoverable, and usable

Globus Labs is a research group led by Prof. Ian Foster and Dr. Kyle Chard that spans the Computation Institute, Department of Computer Science, and Math and Computer Science Division at the University of Chicago and Argonne National Laboratory. Our modest goal is to realize a world in which **all research data are reliably, rapidly, and securely accessible, discoverable, and usable**. To this end, we work on a broad range of research problems in data-intensive

computing and re

Institutes of Healt

fields as diverse as

research data mar

FEATURED PRO

We are developing

extract scientific f

publications

10 Years of Connecting the Research Universe

BREAKTHROUGHS USING GLOBUS ›

Research data management simplified.

funcX

Parsl

DLHub

WHOLE TALE

What is a distributed system?

Tanenbaum:

- … a collection of autonomous computing elements that appears to its users as a single coherent system

Colouris:

- … is one in which components located at networked computers communicate and coordinate their actions only by passing messages

Schroeder:

- … is several computers doing something together. Thus, a distributed system has three primary characteristics: multiple computers, interconnections, and shared state.

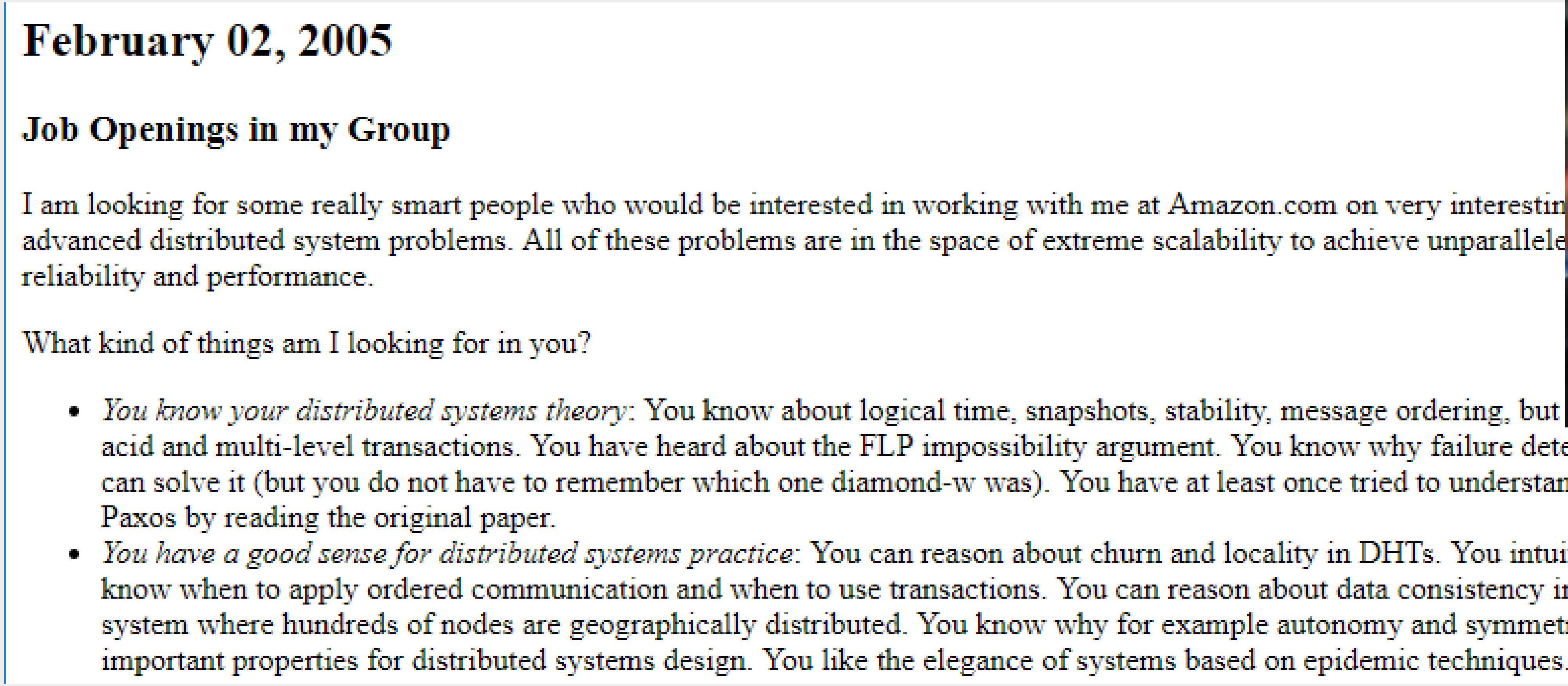
Wikipedia:

- … is a system whose components are located on different networked computers, which communicate and coordinate their actions by passing messages to one another

Why are distributed systems important?

- Performance/scalability
  - End of Moore’s law
- Efficiency
  - Increasing specialization (hardware heterogeneity)
  - Workloads are often sporadic
- Availability/fault tolerance
  - Avoid single point of failure
  - Nodes/software fail (often) and take time to recover
  - Power outage, natural disasters, etc. can affect entire data centers
- Durability
  - Storage decays over time

Increasingly important in industry



February 02, 2005

Job Openings in my Group

I am looking for some really smart people who would be interested in working with me at Amazon.com on very interestin advanced distributed system problems. All of these problems are in the space of extreme scalability to achieve unparallele reliability and performance.

What kind of things am I looking for in you?

- *You know your distributed systems theory*: You know about logical time, snapshots, stability, message ordering, but acid and multi-level transactions. You have heard about the FLP impossibility argument. You know why failure detectors can solve it (but you do not have to remember which one diamond-w was). You have at least once tried to understand Paxos by reading the original paper.
- *You have a good sense for distributed systems practice*: You can reason about churn and locality in DHTs. You intuitively know when to apply ordered communication and when to use transactions. You can reason about data consistency in a system where hundreds of nodes are geographically distributed. You know why for example autonomy and symmetry are important properties for distributed systems design. You like the elegance of systems based on epidemic techniques.

Werner Vogels

Amazon CTO

Increasingly important in industry

- *“Even though I was interviewing nearly a year after your class, I talked about what I learned from your class in every interview. Client-server architecture, sql vs no-sql design decisions and horizontal scaling were the most common topics, and I think interviews that focused on more advanced topics (vector clocks + NTP, Raft, virtualization/Docker, lambda architecture, etc.) were what landed me a promotion.”*

Logistics

TAs

Tyler Skluzacek

Yadu Babuji

Alok Kamatar

Wenyi Wang

Course logistics

- Lectures:
  - Friday 10:30-1:30

- TAs
  - Tyler (zoom): Wednesday 2-3
  - Yadu: Tuesday 3-4 (JCL 205)
  - Wenyi: Monday 2-3 (JCL 205)
  - Alok: Thursday 3-4 (JCL 205)

- Me:
  - Kyle Chard
  - chard@uchicago.edu
  - Office: JCL 303

  - Office hours: Thursday 1-2

Weekly Schedule

- Learning resources:
  - In-person lecture
  - More detailed pre-recorded videos
  - Will follow the book closely (where possible)
- Discussion session and exercises during class
- Office hours (Q&A)
- TA sessions
  - Starting with 4 sessions per week (see previous slide or Canvas) we can scale for homework assignments/project as needed

Course logistics continued

- Slack for interactive discussions
- Canvas for course info
- Homework posted after class and due the following week
- GitHub classroom for all programming homework, classroom exercises, and project submission
- Other issues, please email me (chard@uchicago.edu)

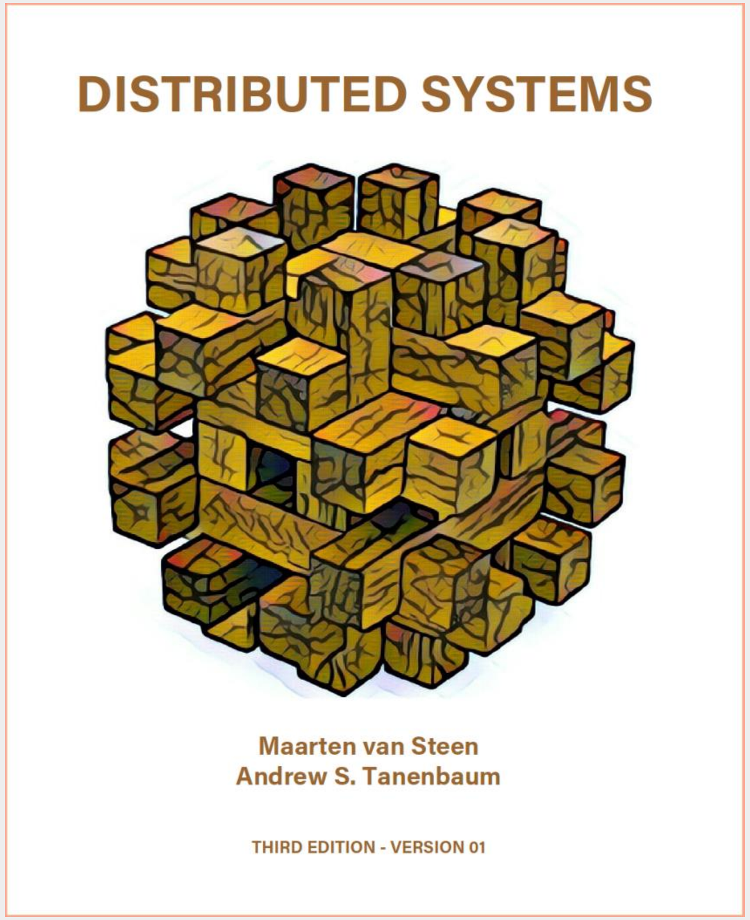
Assessment

- Class exercises: 5% (P/F)
- Homework assignments: 20%
- Project: 30%
- Midterm Exam: 10%
- Final Exam: 35%

Course schedule (* things might change a little)

Date	Lecture discussion topic	Class	Assessment
January 10	1. Introduction to Distributed Systems	Intro/Logistics	
January 17	2 Distributed architectures 3 Processes and virtualization	1 Docker	Homework 1 due
January 24	4 Networks and Communication	2 RPC/ZMQ/MPI	Homework 2 due
January 31	5 Naming	3 DNS/LDAP	Homework 3 due
February 7	6 Coordination and Synchronization	Project description 4 REST	**Mid term exam**
February 14	7 Fault tolerance and consensus	Raft	Homework 4 due (Project released)
February 21	8 Consistency and replication	5 FaaS	
February 28	9 Distributed data	6 Distributed data	
March 7	10 Data-intensive computing		**Project due (March 9)** **Final exam (March 14)**

Textbook



Van Steen and Tanenbaum, Distributed Systems

Free online (or hard copy available on Amazon)

Course Goals

At the end of the course you will:

- Know what distributed systems are
- Understand where distributed systems are used
- Understand the challenges of distributed systems
- Understand solutions for common distributed systems problems
- Gain practical knowledge of systems, tools, libraries, algorithms that can help you build real systems

Expectations

- Weekly content
  - Lecture
  - Class exercises
  - Slides posted online

- Homework extensions (not for project/exams):
  - 2 late chips (total) for the course

Expected workload

Approx. 5-10 hours per week:

- Homework ~5-10 hours
- Project ~40 hours
- Other ~5 hours

Academic integrity

- You are expected to follow the student guide on academic integrity (https://studentmanual.uchicago.edu/)
- Cite everything you use
- Do not copy directly from external sources
- Indicate if you have worked with others and ensure that you submit your own version of code, document, etc.
- Do not use ChatGPT

Prerequisites

- Programming
  - Intermediate knowledge
- Python
  - Basic knowledge
- Knowledge of Linux and shell scripting
- (helpful) Knowledge of operating systems and networks

Asking Questions

- You are going to encounter problems during the course and its important to ask questions that will help clarify your understanding, ambiguity in homework, etc.

- Useful resources:
  - Borja’s guide: https://uchicago-cs.github.io/student-resource-guide/getting-help/questions.html
  - Stack overflow’s guide: https://stackoverflow.com/help/how-to-ask

Key takeaways

- Read the homework carefully (and often)
- The homework won’t include step by step instructions, we expect that you will read documentation, lecture notes, etc.
- Before asking a question
  - Search Slack for similar questions
  - If the question is about how to use a technology, syntax, etc. it is fine to Google
  - Try breaking the problem down, start from a known “good” state
- Asking a question
  - Clearly state what problem you are facing, what error you see, what you are doing to see that error (if its complicated, add the code to GitHub and point us at the problem)
  - State what you have tried to resolve the problem
  - Add logging to understand the flow of the program

Important Links

- **Zoom**: https://uchicago.zoom.us/j/92185332591?pwd=WnNLUHROcWJ0K0ViYzdSd3RxVnFadz09
- **Slack**: https://join.slack.com/t/mpcs52040-winter2025/shared_invite/zt-2xg1ruc05-IhRrV~o~7FRB6utGcYX5IA
- **Canvas**: https://canvas.uchicago.edu/courses/61955

Questions?