What is a distributed system?

- A distributed system is one in which hardware and software components located at networked computers communicate and coordinate their actions only by passing messages.
  - Coulouris, Dollimore, Kindberg and Blair

- A distributed system is one in which the failure of a computer you didn’t even know existed can render your own computer unusable.
  - Leslie Lamport

Distributed systems builds on a diverse set of areas

- Networking
- Concurrency
- Algorithms and Graph Theory
- Cryptography
- Failure recovery and consistency models
- Probability theory
- Machine learning
- Information Retrieval
- Transactional Systems
Distributed Systems: CHALLENGES (1/2)
- Scale with increases in data and users
- Responsiveness
  - Regardless of data size, responses must be prompt
- Intelligent
  - Correlate all sorts of information

Distributed Systems: CHALLENGES (2/2)
- Dealing with system conditions
  - Murphy’s Law
  - Malicious Users
  - Byzantine failures
- Security
  - Detection
  - Privacy and Accountability
  - Authorizations

About me
- I do research in this area
- Areas that I have worked or actively work in include:
  - Cloud computing and analytics
  - Internet-of-Things (IoT)
  - Content dissemination and streaming systems
  - Grid computing
  - P2P systems
  - Object Request Brokers

My research has been deployed in
- Commercial internet conferencing systems
- Defense applications
- Earthquake sciences
- Epidemic modeling
- Environmental monitoring
- Healthcare informatics & Brain Computer Interfaces
- High energy physics
- Visualizations

Course Logistics, Expectations, and such
Course webpage

- All course materials will be on the course webpage [http://www.cs.colostate.edu/~cs455](http://www.cs.colostate.edu/~cs455)
- Schedule
- Lectures
- Assignments
- Announcements
- Grades will be posted on Canvas
- There is also a link to the CS455 Pizza forum on the course webpage
- FAQs and discussions for assignments

Office Hours

- Instructor: Shrideep Pallickara
  - Computer Science (CSB 364)
  - Office Hours: 4:00 – 5:00 PM Tuesday 9:00 – 10:00 AM Friday
- GTA: Office hours will be in CSB 120
  - Naman Shah (Office Hours TBA)
- Lab Sessions:
  - CSB 130 from 4:00 – 5:00 pm on Friday
- Please send all e-mails to: cs455@cs.colostate.edu

Course textbook

- This class has two optional textbooks

When I make slides …

- I usually refer to several texts
- And technical papers and articles (with URLs)
- I always list my references at the end of every slide set

Textbooks that I will refer to during the course include …

On the Course schedule page
https://www.cs.colostate.edu/~cs455/schedule.html

- You will see the topics that will be covered and the order in which I will cover them.
- The readings section will list the books (and the chapters therein) that form the basis for the materials.
- You will also see the complete schedule for when the assignments are posted and when they are due.

Grading breakdown
- Assignments: 55%
  - HW1: 15%; HW2: 20%; HW3: 20%
- Term project and paper: 10%
- Term project presentation: 5%
- Quizzes (10 best): 10%
- Mid Term: 10%
- Final exam: 10%

Grading Policy
- Letter grades will be based on the following standard breakpoints:
  - >= 90 is an A, >= 88 is an A-
  - >= 86 is a B+, >=80 is a B, >=78 is a B-
  - >= 76 is a C+, >=70 is a C
  - >= 60 is a D, and <60 is an F.
- I will not cut higher than this, but I may cut lower.

Grading Policy
- There is no extra credit
  - Any credit you earn, you must do so on a level-playing field with your peers.
- There will be no make-up exams

Grading Policy
- Every assignment will be posted at least 4 weeks before the due date.
- Every assignment will include information about:
  - How much it will count towards the course grade
  - How it will be graded
- Late submission penalty: 7.5% per-day for the first 2 days
- Submissions after the late submission period will have an automatic ZERO
- If you submit the wrong files: 30% deduction
- Detailed submission instructions posted on course website.
- Assignments will be graded within 2 weeks of submission.
Grading Policy

- If you have problems with the grading
  - Talk to the GTA first
- The GTA will strive to ensure that the grading is consistent across the board

Quizzes, mid term, and final

- Each account for 10% of the course grade
- Final is comprehensive
- There will be 12 quizzes
  - 2 quizzes where you had your lowest scores will be dropped
  - We will take your 10 highest scores

If you are interested in taking this course with the honors option

- Honors courses are expected to be tougher courses
- You will be given 1 extra assignment
  - Reliable, ordered communications using UDP
    - Demonstrate this with HW1-PC where TCP is replaced with your implementation
    - The best you can do on this assignment is get a 0
- You might have gotten an A in the regular course
- But deductions in the extra assignment may result in you getting a lower grade

Assignments: What to expect

- There will be no busy work
  - No GUI
- Complexity will not be through obfuscation
- You will be able to look back and feel good about them
  - Delayed gratification

CS455 is a capstone course
So there are writing components

ASSIGNMENTS
Assignments have been designed so that they incrementally add ...

- Networking
- Threading
- Processing

There will be 3 assignments

- Using Dijkstra’s Shortest Paths to Route Packets in a Network Overlay
- Scalable Server Design: Using Thread Pools to Manage Active Network Connections
- Analytics using MapReduce (Hadoop)

Each assignment will have TWO components

- Programming element
- Written element

Assignments: The programming component

- This will account for 80% of the grade for the assignment
- You will have about 4 weeks to complete each assignment
- The assignments will include milestones that should be achieved for each week

Assignments: The written component

- This will account for 20% of the grade for the assignment
- You will have 48 hours to complete this
- The questions will be reflective
  - Design decisions, possible extensions, optimizations, choice of data structures, etc.
- Will be posted after the programming portion is due

Term project (1/2)

- This will be based on Apache Spark
  - Team Project
- Term project deliverables
  - Source codes [7 points]
  - Term Project Report [3 points]
  - Term Project Presentation [5 points]
Term paper (2/2)
- The term project is a group effort
- Team size = 2-3 and you can choose your teammate
- Please respond to your teammate’s e-mails on time!
  - Make sure he/she has the e-mail that you check regularly
- If you have problems finding a teammate, please let us know

Assignments: Logistics
- Assignments will be due at 5:00 pm
  - Programming assignments are due on Wednesdays
  - Written assignments are due on Fridays
- You are allowed to submit up to 2 days late
  - There is a 7.5% deduction for each day that you are late
- All assignments (except the term paper) are individual assignments

Expectations
- You will attend all classes
- You will focus on the discussions, and not on …
  - Other assignments
  - Social networking updates
- Assignments have to be done individually
- You will be challenged in this course
  - Assignments are geared toward real systems

What it takes to succeed (1/3)
- You are required to work at least 12 hours per week outside of class
  - Coding and reviewing material from class
- If you miss a lecture
  - Add about 3 hours per missed lecture

What it takes to succeed (2/3)
- Work on the assignments every day
  - There is no such thing as waiting for inspiration to strike!
- Reflect about how you could have done things differently for better performance
  - Even after you have submitted an assignment
  - It will improve the choices you make in the next assignment
What it takes to succeed (3/3)

- Work in bigger-sized chunks
  - Too many short bursts = Too many context switches
  - You will be busy doing nothing
- Document your code

How to fail this course?

- Believing that you can learn via osmosis
- **Missing lectures**
  - If you don’t have the discipline to show up, you will most likely not have the discipline to catch up
- **Procrastinating**
  - The assignments cannot be done in a week
  - Organize your schedule so that you can succeed

Other pitfalls

- Poor management of **course loads**
  - Plan the number and type of courses you take
  - Don’t spread yourself so thin that you do not give yourself the opportunity to succeed
- **Not attacking the problem and working on the fringes**
  - Spend your time wisely on critical paths

Use of laptops, cell phones, tablets, and other electronic devices

- If you must use a laptop or tablet you will have to
  - Turn off wireless
  - And use it only for taking notes
- **Authorized laptop/tablet users**
  - Pledge forms on table
  - Will sit in the back row starting at the corners
- **When the class is in session, put away your cell-phones!**

Why attend lectures if all the slides are posted?

- Slides are only part of the story
  - They anchor the discussion
- **Any field has a language** associated with it
- People who have worked in an area for a long time speak the language
  - Sitting in classes helps you learn how to frame questions and responses
- Often there are surprising questions
  - Some of these may be asked by interviewers

Help me help you

- We will have **surveys** at the end of every class
- You will provide a list of
  - 3 concepts you followed clearly
  - 3 concepts you had problems keeping up with
- Problem areas for the majority of the class will be addressed in the next class.
Interactions

- You can have discussions with me, the GTA, and your peers.
- There are two constraints to these discussions:
  1. No code can be exchanged under any circumstances.
  2. No one takes over someone else's keyboard.
- Bumps are to be expected along the way:
  - But you should get over this yourself.
  - It will help you with the next problem you encounter.

Topics covered in CS455

- Communications
- Threads: Safety and Concurrency
- Building scalable servers
- MapReduce
- Spark
- RPCs and Distributed Objects

Example:
Setting up connections to a server

- Programs open a socket to a server that's listening for connections.
- To create a Socket, you need to know the Internet host you want to connect to.
- Servers don't know who will contact them.
  - If it did, difficult to synchronize when this would happen.
An analogy

- Server is like a person sitting by the phone
  - Doesn’t know who will call and when
  - When the phone rings?
    - Talk to whoever is on the other line

Java provides a ServerSocket to enable writing servers

- ServerSocket runs on the server
  - Listens for incoming network connections on a particular port on the host that it runs on
- When a client socket on a remote host attempts to connect to that server port
  1. Server wakes up
  2. Negotiates a connection between the client and server
  3. Opens a regular Socket between the two hosts

Some more about the two types of sockets

- ServerSockets wait for connections
- Client Sockets initiate connections
- Once the ServerSocket has set up the connection?
  - Data always travels over the regular Socket

Using the ServerSocket

- Created on a particular port using the ServerSocket(port) constructor
- Listen for communications on that port using accept()
  - Blocks until a client attempts to make connection
  - Returns a Socket object that connects the client to the server
- Use the Socket’s getInputStream() and getOutputStream() to communicate

Creating the ServerSocket

- ServerSocket serverSocket = new ServerSocket(5000);
- Tries to create a server socket on port 5000
- ServerSocket serverSocket = new ServerSocket(5000, 100);
- Can hold up to 100 incoming connections
- ServerSocket serverSocket = new ServerSocket(5000, 100, InetAddress.getHostByName("address2.cs.colostate.edu"));
- On a multi-homed host, specify the network-address over which connections should be accepted

Accepting network connections

ServerSocket serverSocket = new ServerSocket(portNum);
while(true) {
  Socket socket = serverSocket.accept();
  ...
}
Closing the client and server sockets

- Closing a ServerSocket frees a port on the host that it runs on.
- Closing a Socket breaks the connection between the local and remote hosts.

We exchange byte streams over the socket

- The java.io package contains the DataInputStream and DataOutputStream that lets you do this elegantly.
- DataInputStream `din = new DataInputStream(socket.getInputStream());`
- DataOutputStream `dout = new DataOutputStream(socket.getOutputStream());`

Elements that play a role in communications

- Transmission media
  - Wire, cable, fiber, and wireless channels
- Hardware devices
  - Routers, switches, bridges, hubs, repeaters, and network interfaces
- Software components
  - Protocol stacks, communication handlers, and drivers

Communications & Networking: Topics that we will cover

- Data transmission
- Switched Networks
- Bandwidth and Latency
- Multiplexing
- Internet Architecture
- IP routing
- The TCP and UDP protocols

How is the data sent?

- Are we sending 1’s and 0’s?
- Whatever the physical medium, we use signals
  - Electromagnetic waves traveling at the speed of light
  - Speed of light is different in different mediums
Components of encoding binary data in a signal

- Modulation
- Duplexity

Encoding binary data: Modulation

- Objective is to send a pair of distinguishable signals
- Vary frequency, amplitude, or phase of the signal to transmit information
  - E.g. vary the power (amplitude) of signal
  - \( x(t) = A \sin(2\pi ft + \theta) \)

Encoding binary data: Duplexity

- How many bit streams can be encoded on a link at a time?
  - If it is one, nodes must share access to link
- Can data flow in both directions at the same time?
  - Yes \(\Rightarrow\) full-duplex
  - No \(\Rightarrow\) half-duplex

For our purposes, let’s ignore details of modulation

- Assume we are working with two signals
  - High and low
- In practice:
  - Different voltages on a copper-based link
  - Different power-levels on an optical link

Let’s do the obvious thing

- Map 1 to a high signal
- Map 0 to a low signal

Non-return to zero (NRZ)

0 0 1 0 1 1 1 1 0 1 0 0 0 0 1 0
Problems with NRZ because of consecutive 1’s and 0’s: **BASELINE WANDER**
- Receiver keeps average of the signal seen so far
- Average is used to distinguish between low and high
- Lots of consecutive 1/0’s will make it difficult to detect a significant change

Problems with NRZ because of consecutive 1’s and 0’s: **CLOCK RECOVERY**
- Every clock cycle, sender transmits and the receiver receives
- Sender and receiver’s clocks must be perfectly synchronized
  - Otherwise, it is not possible to decode the signal

---

**Manchester encoding**
- 0 is a low-to-high transition
- 1 is a high-to-low transition

**Manchester encoding and NRZ**

NRZI (Non return to zero inverted)
- Make a transition from current signal to encode a 1
  - Stay at current signal to encode a 0
- Solves the problem of consecutive 1’s
  - But does nothing for consecutive 0’s

Some more about Manchester encoding
- Doubles the rate at which signal transitions are made on the link
  - Receiver has ½ the time to detect each pulse
- Rate of signal changes: baud rate
  - Bit rate is ½ the baud rate
  - Encoding is considered 50% efficient
4B/5B encoding

- Attempts to address inefficiencies in Manchester encoding
- Without suffering from problems due to extended high/low signals
- The crux here is to insert extra bits into bitstream
- Breakup long sequences of 1s or 0s
- 4 bits of actual data encoded in a 5-bit code
- 5-bit codes are carefully selected
  - No more than 1 leading 0 & no more than 2 trailing 0s

4B/5B: Rules for the conversion of 4-bit codes to 5-bit codes

- Objective is to ensure that in each translation there is:
  - No more than one leading 0
  - No more than two trailing 0s
  - When sent back-to-back
    - No pair of 5-bit codes results in more than 3 consecutive 0's being transmitted
- 5-bit codes are transmitted using NRZI
  - This is why they are so concerned with consecutive 0's

The contents of this slide-set are based on the following references