CS 455: INTRODUCTION TO DISTRIBUTED SYSTEMS

[INTRODUCTION]

The Road Ahead
- Winds from ports to packets
- And thence onto sockets
- Over which all data must traverse
- Intricacies of threads unraveled
- To prod CPUs along roads less traveled

SHRIDEEP PALICKARA
Computer Science
Colorado State University

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

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

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

DISTRIBUTED SYSTEMS
QUICK OVERVIEW

Professor: SHRIDEEP PALICKARA

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)

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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

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)

Topics covered in this lecture
- Introduction
- Course overview and expectations
- Communications

Why Distributed Systems?
- Your hard-drive’s primacy has been eroding
- Data and programs are delivered over the network
  - No single hard drive can hold all the data you need
- Services themselves are distributed
  - Google search is backed by a massive distributed cloud

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)
  - Big Data
  - Content dissemination and streaming systems
  - Grid computing
  - P2P systems
  - Object Request Brokers

My research has been deployed in

- Urban sustainability
- 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:
  - Schedule
  - Lectures
  - Assignments
  - Infospaces, Syllabus, and Announcements
- Grades will be posted on Canvas
- There is also a link to the CS455 Pizza forum on the course webpage
- FAQ's and discussions for assignments
- The course website, Piazza, Checkin, and Canvas are all live now

Office Hours
- Professor: Shrideep Pallickara
  - Computer Science (CS B 364)
  - Office Hours: 9:00 – 10:00 AM Friday
- GTA: Office hours will be in CS B 120
  - Brandon Gildemaster: 2:00-4:00 pm Tuesday and Thursday
  - Jason Stock: 2:00-4:00 pm Monday and Wednesday
- Lab Sessions: CS B 130 from 4:00 – 5:00 pm on Friday
- Please send all e-mails to: compsci_cs455@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 ...

Textbooks that I will refer to during the course include ...
On the Course schedule page
http://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.

Infospaces (http://infospaces.cs.colostate.edu)

- New knowledge repository that we are building to enhance learning.
- All videos are designed to be less than 2 minutes.
- Improving Infospaces:
  - Let us know what you would like to see.
  - If you’d like to contribute to this repository let us know!

Grading breakdown

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

Grading Policy

[1/4]

- 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

[2/4]

- 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 [3/4]
- 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 and notice after 2 days? 40% deduction
- Detailed submission instructions posted on the course website.
- Assignments will be graded within 2 weeks of submission

Grading Policy [4/4]
- If you have problems with the grading
  - Talk to the GTAs first
- The GTAs strive to ensure that the grading is consistent across the board

Quizzes, midterm, and final
- Each component accounts for 10% of the course grade
- Final is comprehensive
- There will be 13 in-class quizzes
  - 3 quizzes where you had your lowest scores will be dropped
  - We will take your 10 highest scores
  - If you have an interview or need to miss class for some reason, you do not need to let me know and there will be NO MAKEUP for quizzes (please don’t ask to do this)
  - Exceptions with documentation: University sanctioned absences, medical emergencies, personal loss

Quizzes
- Are held at the end of class,
- Quizzes will have checkboxes that require to indicate if you were
  - On-time or less than 1.5 minutes late: No deductions
  - 1.5+ minutes late (20% deduction), 30+ minutes late (40% deduction), 45+ minutes late (60% deduction), 60+ minutes late (80% deduction)
  - This is self-reported. Failure to report, doubles the deduction

Quizzes, midterm, and final
- I will only ask questions about what I teach
- If the concepts were covered in my slides
  - You should be be able to answer the questions

ASSIGNMENTS

CS455 is a capstone course
So there are writing components
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

Assignments have been designed so that they incrementally add ...
- Networking
- Threading
- Processing

There will be 4 assignments
- Routing Packets Within a Structured Peer-to-Peer (P2P) Network Overlay
- Threading Assignment
- Analytics using MapReduce (Hadoop)
- RamCoin: Cryptocurrency and Blockchain

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, extensions, optimizations, choice of data structures, etc.
- Will be posted after the programming portion is due
- Please submit the written component EVEN IF you have not completed the programming assignment
Term project [1/2]
- This will be based on Apache SPARK
- Term project deliverables
  - Source codes [7 points]
  - Term Project Report [3 points]
  - Term Project Presentation [5 points]

Term project [1/2]
- 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 project) 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
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 designed 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
Recorded lectures are ONLY for distance students

- The on-campus course is the on-campus course
- Students in the on-campus section will not be given the recording links
- Exceptions: University sanctioned absences, medical emergencies, personal loss
- Distance students should not post (or share) the EchoCenter links for the recorded lectures on the Piazza forum or any other forum.

Help me help you

- We will have surveys at the end of every class
- You will provide a list of
  1. 3 concepts you followed clearly
  2. 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 GTAs, 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
  1. But you should get over this yourself
  2. It will help you with the next problem you encounter

Topics covered in CS455

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

- Distributed mutual exclusion
- Election algorithms
- File systems and network storage
- Distributed server topologies
- Distributed storage systems
- Blockchains
**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?
    - Talks 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 getInputSteam() 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

```java
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
  ```java
  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 travelling 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)

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

NRZ

Manchester Encoding
Some more about Manchester encoding

- Doubles the rate at which signal transitions are made on the link
- Receiver has $\frac{1}{2}$ the time to detect each pulse
- Rate of signal changes: baud rate
- Bit rate is $\frac{1}{2}$ the baud rate
- Encoding is considered 50% efficient

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

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
  - 4 bits of actual data encoded in a 5-bit code
  - Breakup long sequences of 1s or 0s
  - 5-bit codes are carefully selected
  - No more than 1 leading 0 & no more than 2 trailing 0s

4B/5B encoding

<table>
<thead>
<tr>
<th>4B</th>
<th>5B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000</td>
<td>11110</td>
</tr>
<tr>
<td>0001</td>
<td>01101</td>
</tr>
<tr>
<td>0010</td>
<td>10100</td>
</tr>
<tr>
<td>0011</td>
<td>10101</td>
</tr>
<tr>
<td>0100</td>
<td>01110</td>
</tr>
<tr>
<td>0101</td>
<td>10110</td>
</tr>
<tr>
<td>0110</td>
<td>10111</td>
</tr>
<tr>
<td>0111</td>
<td>01101</td>
</tr>
<tr>
<td>1000</td>
<td>10010</td>
</tr>
<tr>
<td>1001</td>
<td>10011</td>
</tr>
<tr>
<td>1010</td>
<td>11010</td>
</tr>
<tr>
<td>1011</td>
<td>11011</td>
</tr>
<tr>
<td>1100</td>
<td>11001</td>
</tr>
<tr>
<td>1101</td>
<td>11011</td>
</tr>
<tr>
<td>1110</td>
<td>11100</td>
</tr>
<tr>
<td>1111</td>
<td>11101</td>
</tr>
</tbody>
</table>

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

- Objective is to ensure that in each translation there 1s:
  - 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