DISTRIBUTED SYSTEMS
CSCI 4963/6963

9/1/2015
General Information

- Lectures: TF 12pm – 1:50pm in JEC 3207
- Instructor: Stacy Patterson (me) sep@cs.rpi.edu
- Office Hours: T 2pm – 3pm in Lally 301

- Course enrollment will be done through SIS only.
  - There is a waiting list in SIS.
  - As of today, there are 6 students on the waiting list.
  - Some people will drop in the first week, so seats will open up.

- There will be more theory courses offered in the spring.
General Information (continued)

• There is no textbook for this course.

• Conference and journal papers related to the course content will be posted on the course web site.

• I will not post lecture notes on the web site.
  • If I use slides, I will post them on the web site.

• Grades will be made available through LMS.
Pre-Requisites

• CSCI-2300: Intro to Algorithms
  • Analysis of algorithm correctness and performance
  • Writing correct proofs of algorithm properties

• CSCI-4210: Operating Systems
  • Multi-threaded programming
  • Network communication (socket programming)

• No linear algebra or PDEs in this course.
Course Objectives

• This is a theory course, despite the name.

• The goal is to for you to learn important theory and algorithms for distributed computing systems.
  • Through theory and practice.

• These algorithms are actually used in data centers and cloud computing systems today!
Grading

- Quizzes: 65%
- Programming Projects: 35%
Quizzes

• There will be approximately 8 in-class quizzes.
  • Closed book.
  • About 45 minutes each.

• Quizzes will be done independently (unless otherwise announced).

• Quizzes will be announced in the lecture preceding the lecture in which they will be given.

• Regrade requests must be made within 7 days of return.

• First quiz will be next Tuesday 9/8/2015.
  • You can do this quiz in pairs.
  • Closed book.
Programming Projects

• There will be 2 programming projects.

• Projects will be done in groups of 2.
  • Exceptions to this must be approved by me in advance.

• Projects will give you the chance to implement distributed algorithms in real-world distributed computing systems (hopefully EC2).
  • You can use your language of choice (within reason).
  • More details in a few weeks.
Other Stuff

• If you need special accommodations for this class, please let me know as soon as possible.

• Academic Integrity Policy:
  • For programming assignments, you may discuss the project with other students, but you (your team) must write your own code.
    • No sharing code or reusing code unless approved by me in advance.
  • No collaboration or outside resources are allowed on quizzes unless I announce otherwise.
  • Violation of this policy will result in an academic penalty.
INTRO TO DISTRIBUTED SYSTEMS
What is a distributed system?

“A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages.”

Coulouris et al., Distributed Systems

Significant characteristics

- **Concurrency**: Different operations executed on different computers at the same time
- **No global clock**: Difficult to synchronize (coordinate) actions on different computers
- **Independent failures**: computers can crash, the network may fail or slow down, network partitions may arise.
  - The rest of the system keeps running, may not be aware of failures.
• I want the application to behave like
  • it is running on a single computer with infinite resources that never fails,
  • and I am the only one using that application.
• The application is actually
  • running on thousands(?) of computers,
  • spread across multiple data centers,
  • with thousands(?) of simultaneous users.
What is a distributed system?

“A distributed system is a system in which I can’t do my work because some computer that I’ve never even heard of has failed.”

Leslie Lamport
Topics

- Clocks and the ordering events in distributed systems
- Distributed mutual exclusion
- Distributed logs
- Global snapshots
- Broadcast algorithms
- Leader Election
- Distributed Agreement
Topics

• Transaction Management

• Distributed Commit Protocols

• Replication and the CAP Theorem

• Consistency Models, Cloud databases

• P2P Systems, Distributed Hash Tables, Bit Torrent

• Bit Coin
MODEL OF A DISTRIBUTED SYSTEM
Properties of a Model

• What are the entities that are communicating in the distributed system?
  • An entity is a single process
  • Other options: objects, services, …

• What communication paradigm do they use?
  • Entities communicate by sending messages
  • Other options: shared memory, RPC, publish/subscribe, …

• How are they mapped onto the physical distributed infrastructure?
  • A process runs on a single physical machine
  • Other options: mobile code, mobile agents, …
Two Variants

• **Synchronous System:** Known bounds on times for message transmission, processing, bounds on local clock drifts, etc.
  • Can use timeouts

• **Asynchronous System:** No known bounds on times for message transmission, processing, bounds on local clock drifts, etc.
  • More realistic, practical, but no timeout.