Computer Science 425
Distributed Systems

CS 425 / CSE 424 / ECE 428

Fall 2012

Indranil Gupta (Indy)

August 27-December 11, 2012

Lecture 1-29

Website: http://courses.engr.illinois.edu/cs425/fa2012/

Our First Aim in this Course was… (first lecture)….

(First lecture slide)

To Define the Term Distributed System

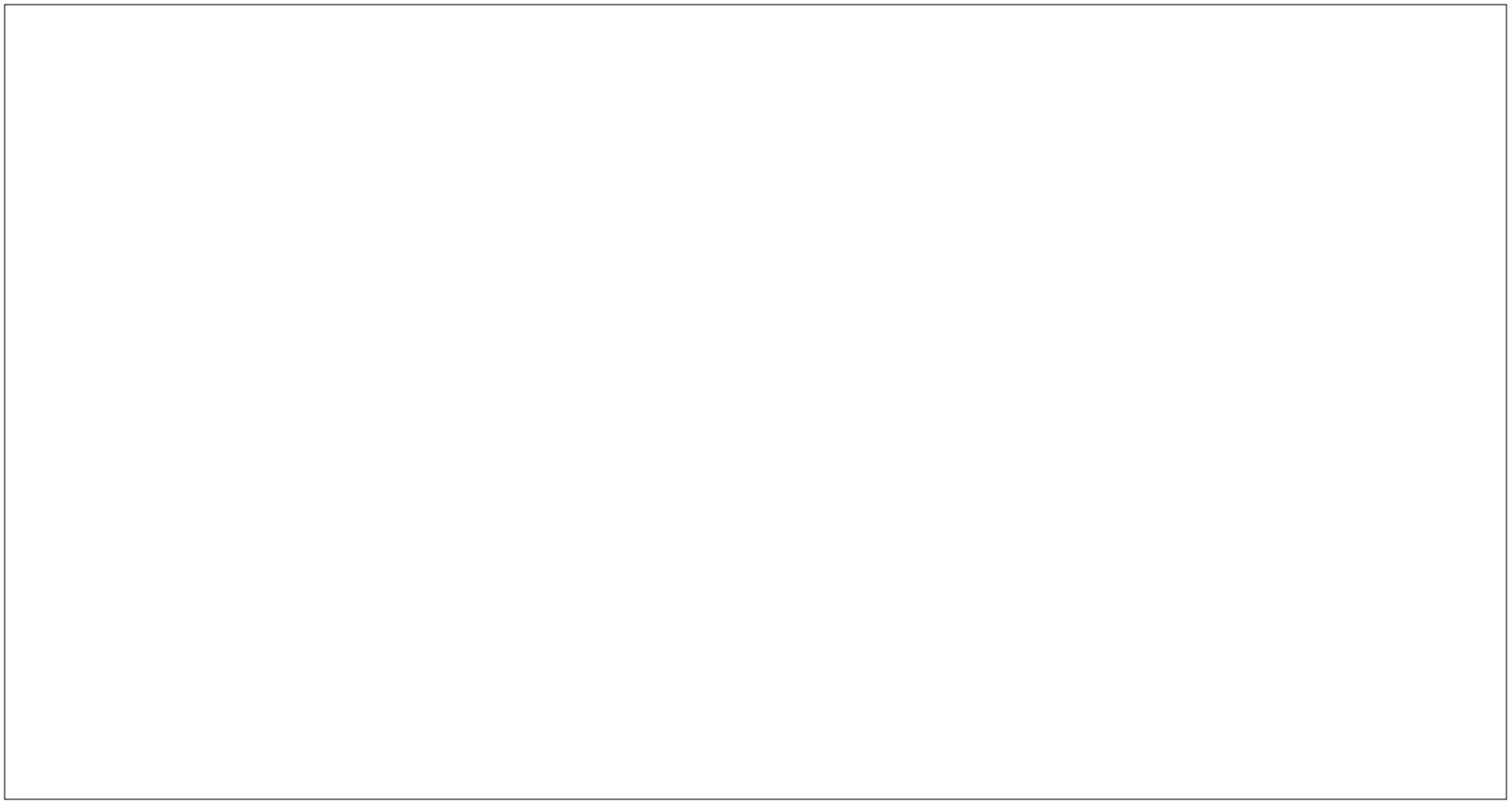
Can you name some examples of Distributed Systems?

(First lecture slide)

- **Client-Server (NFS)**
- **The Web**
- **The Internet**
- **A wireless network**
- **DNS**
- **Gnutella or BitTorrent (peer to peer overlays)**
- **A "cloud", e.g., Amazon EC2/S3, Microsoft Azure**
- **A datacenter, e.g., NCSA, a Google datacenter, The Planet**
- **(Society?)**

What is a Distributed System?

(First lecture slide)



FOLDOC definition

(First lecture slide)

A collection of (probably heterogeneous) automata whose distribution is transparent to the user so that the system appears as one local machine. This is in contrast to a network, where the user is aware that there are several machines, and their location, storage replication, load balancing and functionality is not transparent. Distributed systems usually use some kind of client-server organization.

Textbook definitions

(First lecture slide)

- **A distributed system is a collection of independent computers that appear to the users of the system as a single computer**

  [Andrew Tanenbaum]

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

  [Michael Schroeder]

A working definition for us

(First lecture slide)

A distributed system is a collection of entities, each of which is autonomous, programmable, asynchronous and failure-prone, and which communicate through an unreliable communication medium.

- **Entity=a process on a device (PC, PDA)**
- **Communication Medium=Wired or wireless network**
- **Our interest in distributed systems involves**
  - **design and implementation, maintenance, algorithmics**
- ***What Evidence/Examples have we seen?***

Problems we have Seen and Solved Since Then

- **Failure Detectors**
- **Time and Synchronization**
- **Global States and Snapshots**
- **Multicast Communications**
- **Mutual Exclusion**
- **Leader Election**
- **Impossibility of Consensus**
- **Peer to peer systems – Napster, Gnutella Chord**
- **Cloud Computing**
- **Networking and Routing**
- **Sensor Networks**
- **Measurements from real systems**
- **Datacenter Disaster Case Studies**

Problems we have Seen and Solved in this Class

- **Failure Detectors**
- **Time and Synchronization**
- **Global States and Snapshots**
- **Multicast Communications**
- **Mutual Exclusion**
- **Leader Election**
- **Impossibility of Consensus**

Basic Theoretical Concepts

- **Peer to peer systems – Napster, Gnutella Chord**
- **Cloud Computing**

Bridge to Systems

- **Networking and Routing**
- **Sensor Networks**
- **Measurements from real systems**
- **Datacenter Disaster Case Studies**

What Lies Beneath

Problems we have Seen and Solved in this Class (2)

- **RPCs & Distributed Objects** ← Basic Building Blocks
- **Concurrency Control**
- **2PC and Paxos**
- **Replication Control**
- **Gossiping**

Distributed Services (e.g., storage)

- **Key-value and NoSQL stores**
- **Distributed Graph Processing**

Cloud Computing

- **Self-stabilization**
- **Distributed File Systems**
- **Distributed Shared Memory**

Miscellaneous

- **Security and Byzantine Fault-tolerance** ← Important

Problems we have Seen and Solved in this Class (3)

- **Midterm**
- **HW's and MP's**

How to get good grades (and regrades, and jobs in some cases)

  - **You've built a new cloud computing system from scratch!**

Something to boast about to your friends (and in interviews!)

Typical Distributed Systems Design Goals

(First lecture slide)

- **Common Goals:**
  - **Heterogeneity**
  - **Robustness**
  - **Availability**
  - **Transparency**
  - **Concurrency**
  - **Efficiency**
  - **Scalability**
  - **Security**
  - **Openness**

Typical Distributed Systems Design Goals

- **Common Goals:**
  - **Heterogeneity:** different types of servers, of networks, of applications, of services, of consistency guarantees
  - **Robustness:** fault-tolerance to a variety of failures
  - **Availability:** of data, of operations, in spite of failures and network partitions
  - **Transparency:** provide an abstraction of one property while allowing sufficient flexibility at run-time, e.g., clouds, transactions, virtual synchrony, sequential consistency, etc
  - **Concurrency:** support many clients (millions)
  - **Efficiency:** fast operations, e.g., reads and writes in NoSQL
  - **Scalability:** many operations per second in spite of thousands of servers, millions of clients
  - **Security:** system should be protected from attackers and bugs, e.g., encryption and signatures
  - **Openness:** each service/protocol can build on other services/protocols, e.g., layered or stacked architecture

Problems we have Seen and Solved in this Class

(and relation to other courses)

- **Failure Detectors**
- **Time and Synchronization**
- **Global States and Snapshots**
- **Multicast Communications**
- **Mutual Exclusion**
- **Leader Election**
- **Impossibility of Consensus**
- **Peer to peer systems – Napster, Gnutella Chord**
- **Cloud Computing**
- **Sensor Networks**
- **Measurements from real systems**
- **Datacenter Disaster Case Studies**
- **Networking and Routing**

Core Material of this course

Related to **CS 525 (Advanced Distributed Systems Offered Spring 2013)**

Related to CS 438/538

Problems we have Seen and Solved in this Class

(and relation to other courses)

- **RPCs & Distributed Objects**
- **Concurrency Control**
- **2PC and Paxos**

Core Material of this course

- **Replication Control**

Related to CS 411/CS 511

- **Gossiping**
- **Key-value and NoSQL stores**
- **Distributed Graph Processing**

Related to **CS 525**

- **Self-stabilization**
- **Distributed File Systems**

Related to CS 421/CS 433

- **Distributed Shared Memory**

Related to CS 423/523

- **Security and Byzantine Fault-tolerance**

CS525: Advanced Distributed Systems (taught by Indy)

CS 525, Spring 2013

- Looks at hot topics of research in distributed systems: clouds, p2p, sensor networks, distributed algorithms and and other distributed systems
- We read many papers and webpages for cutting-edge systems (research and production)
- If you liked CS425's material, it's likely you'll enjoy CS525
- Project: Both Research and Entrepreneurial projects
  - » Your project will build a cutting edge research distributed system, and write and publish a paper on it
  - » Your project will build a distributed system for a new startup company idea (your own!) and perform associated research with it
- Both graduates and undergraduates welcome! (let me know if you need my consent).
- Class size is around 45

Questions?

A working definition for us

A distributed system is a collection of entities, each of which is autonomous, programmable, asynchronous and failure-prone, and which communicate through an unreliable communication medium.

- **Entity=a process on a device (PC, PDA)**
- **Communication Medium=Wired or wireless network**
- **Our interest in distributed systems involves**
  - **design and implementation, maintenance, algorithmics**

[Is this definition still ok, or would you want to change it?]

Final Exam

- **Regular Indy + TA (Imranul and Abhishek) office hours the rest of this week (as per usual schedule)**
- **Final Exam**
  - **Friday Dec 14th 7:00-10:00pm 1RAL 116 (Roger Adams Laboratory).**
    - » **Goodwin/Oregon intersection**
  - **Syllabus: Includes all material since the start of the course. More emphasis on material since midterm**
  - **Cheat sheet: Allowed to bring a *cheat sheet* to the exam (A4 size, two sides only, at least 1 pt font)**
  - **Structure: Final will be similar in structure to Midterm, only longer**
  - **Preparing: Revising homework problems, and midterm problems, and textbook problems**

Course Evaluation

- **Main purpose: to give us feedback on how useful this course was to you (and to improve future versions of the course)**
- **I won't see these evaluations until after you see your grades**
- **Use pencil only**
- **Answer questions 1 and 2 (you can skip #5)**
- **Please write your detailed feedback on the back – this is valuable for future versions of the course!**

- **Need a volunteer:**
  1. **Please collect all reviews, and drop envelope in *campus mail box***
  2. **Return the box of pencils to me (3112 SC)**

Thank You