# Computer Science 425 Distributed Systems

Lecture 28

“Wrap-Up”

Klara Nahrstedt

## *Administrative*

- MP3 posted
  - Deadline December 7 (Monday) – pre-competition in cs216 SC
    - » Top five groups will be selected for final demonstration on Tuesday, December 8
  - Demonstration Signup Sheet for Monday demonstrations – sign up online on the website

## Administrative

- MP3 - Tuesday, December 8 Final Competition
  - Start at 2pm in 216 SC
  - Competition goes between 2-5pm
  - Groups will be informed on Monday, December 7 by 6pm about the results (email and website)
  - Each group starts with power point presentation and follows up with the demonstration of the features
  - Competition Criteria:
    - » presentation clarity,
    - » demonstration of features,
    - » explanation of features during demonstration,
    - » response to questions,
    - » stability of demonstration
  - Final results from the competition will be announced at 5pm in 216 SC – so please come all !!!

## Administrative

- MP3 - Documentation due by December 9
  - Email URL link to Ying (TA) with all your project descriptions (see next slide for project template description link)
  - Readme file must include:
    - » Boot-straping routine – how one install your system – developers manuscript
    - » How one use your system – usage prescription for users
    - » Known bugs, what are the issues with your system/application
  - Tar or zip your source code and upload it to agora wiki
    - » URL Information will be provided on the web/in class/on newsgroup
  - Fill out project template as specified
    - » Template Information will be provided on the web/in class/on newsgroup

## *Administrative*

- MP3 instructions for Project Documentation Preservation
  - Here's the template page for cs425 students to copy and fill out.

https://agora.cs.illinois.edu/display/mlc/cs425-TemplateProject

- Website only cs425 students and instructors can access to post the template page and also upload attachments

https://agora.cs.illinois.edu/display/mlc/cs425-fa09-projects

## *Plan for Today*

- Brief review what this course was all about
- Information about the Final Exam
  - Room, time, grading, etc.
- Short examples of final exam problems
- ICES – course evaluation

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

To Define the Term Distributed System

## *Can you name some examples of Distributed Systems?*

- Client-server (e.g., NFS)
- The Internet
- The Web
- An ad-hoc network
- A sensor network
- DNS
- Napster, Gnutella, Fast-track (peer to peer overlays)
- Any other?

## FOLDOC definition (Free On-line Dictionary Of Computing)

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

- 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 communicating through an unreliable communication medium.*

- Our interest in distributed systems involves
  - design and implementation, maintenance, study, algorithmics
- Entity=a process on a device (PC, PDA)
- Communication Medium=Wired or wireless network
- *[Is this definition still ok, or would you want to change it?]*
- *What Evidence/Examples have we seen?*

## *Problems we have Seen and Solved Since Then*

- **Time and Synchronization**
- **Global States and Snapshots**
- **Multicast Communications**
- **Mutual Exclusion**
- **Leader Election**
- **Impossibility of Consensus**
- **Failure Detectors**
- **Peer to peer systems – Gnutella and Chord**
- **Networking – Internet Routing protocols (RIP, OSPF)**

## Problems we have Seen and Solved in this Class

- Time and Synchronization
- Global States and Snapshots
- Multicast Communications
- Mutual Exclusion
- Leader Election
- Impossibility of Consensus
- Failure Detectors
- Peer to peer systems – Gnutella and Chord
- Networking – Internet Routing
  - RIP, OSPF

Basic Theoretical Concepts

Bridge to Systems

What Lies Beneath

# Problems we have Seen and Solved in this Class

- RPCs & Distributed Objects ← Basic Building Blocks
- Transactions
- Concurrency Control
- Distributed Transactions
- Replication Control

} Distributed Services (e.g., databases)

- Distributed File Systems
- Ubiquitous / Mobile Systems

} Distributed Apps that people use directly

- Security ← Necessary properties
- The Grid ← Computational Facility that scientists use directly

## *Problems we have Seen and Solved in this Class*

*(and relation to other courses)*

Core Material of this course

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

Related to CS 525

- Failure Detectors
- Peer to peer systems – Gnutella and Chord

Related to CS 438

- Networking – Internet

# *Problems we have Seen and Solved in this Class*

***(and relation to other courses)***

Core Material of this course

- **RPCs & Distributed Objects**

Related to CS 411/CS 511

- **Transactions**
- **Concurrency Control**
- **Distributed Transactions**
- **Replication Control**

Related to CS 423/CS 523

- **Distributed File Systems**

Related to CS 439, 598rk, 598rc

- **Ubiquitous /Mobile Systems**

Related to CS 461/463/

- **Security**

Related to CS 525

- **The Grid**

# A range of Challenges for Designers of Distributed Systems

-
- Heterogeneity
- Openness
- Security
- Scalability
- Failures
- Concurrency
- Transparency
-
-

What are these?

## What we Have Learned Since Then

- Heterogeneity – protocols should run in spite of heterogeneous nodes, e.g., supernodes in Kazaa
- Openness – each service/protocol can build on other services/protocols, e.g., layered or stacked architecture
- Security – it should be possible to add-in security to any existing protocol/system, e.g., encryption and signatures
- Scalability – peer to peer systems need to scale in overhead as the size of the system increases
- Failures – any protocol should be tolerant to the failure of nodes and of communication
- Concurrency – the more the better the performance, but is a tradeoff with consistency requirements
- Transparency – distributed file systems, transactions, virtual synchrony, sequential consistency, etc. provide an abstraction of one property while allowing sufficient flexibility at run-time

## *Final Exam Information*

- **Date: Wednesday, December 16, 2009**
- **Time: 7-10pm**
- **Location:**
  - **1105 SC : Students with last name starting with letter: A-L**
  - **1214 SC: Students with last name starting with letter: M-Z**
- **Extended Office Hours:**
  - **December 15, 9-10am, 3104 SC**
  - **December 10,  Reading Day – no office hours**
- **Score: 30% of your grade**
- **Logistics:**
  - **Allowed to bring a *cheat sheet* to the exam (A4 size, two sides)**

## *Final Exam Material*

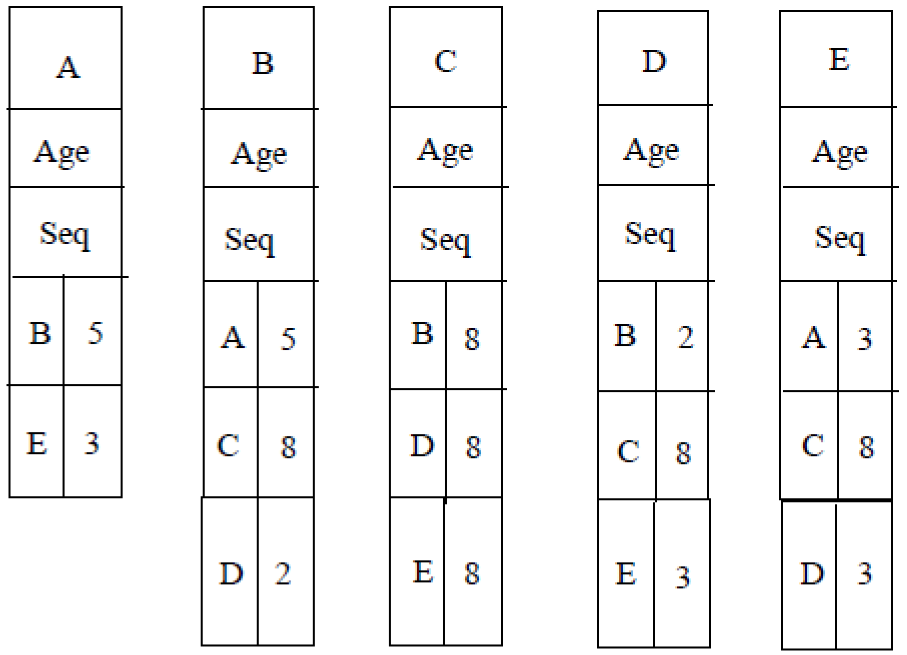
- All Material of the class
  - » 1/3 problems from material before Midterm (Basic Theoretical Concepts )
    - Snapshots and Time synchronization (Lamport clock); Reliable and ordered multicast; Group Communication ; Mutual Exclusion; Leader Election; Consensus; Failure Detection
  - » 2/3 problems from material after Midterm
    - P2P, Networking/Routing, RPC, Distributed Services, Self-stabilization, Transactions and Concurrency Control, Distributed Transactions, Replication Control, DFS, Security, Mobile and Sensor Systems, and Grid

# Final Exam Problem Types: Multiple Choice

- When a browser process is supplied with a website's URL, this eventually results in the invocation of several services (in order to locate and contact the web-service's server). Which of the following best describes these services
  1. A routing protocol only
  2. File System only
  3. RPC only
  4. All of the above
- In AFS, if the client and the server are different hosts, which of the following client-side file operations lead(s) to an RPC that crosses the network?
  1. Open()
  2. Read()
  3. Write()
  4. All of the above

## Calculation and Algorithmic Problems

- Consider a subnet that consists of routers A,B,C,D,E (symmetrical cost where a lower cost is more desirable)



| A | |
|---|---|
| Age | |
| Seq | |
| B | 5 |
| E | 3 |

| B | |
|---|---|
| Age | |
| Seq | |
| A | 5 |
| C | 8 |
| D | 2 |

| C | |
|---|---|
| Age | |
| Seq | |
| B | 8 |
| D | 8 |
| E | 8 |

| D | |
|---|---|
| Age | |
| Seq | |
| B | 2 |
| C | 8 |
| E | 3 |

| E | |
|---|---|
| Age | |
| Seq | |
| A | 3 |
| C | 8 |
| D | 3 |

  - Suppose the subnet uses link state routing. If router A receives the above link state advertisements (LSA) packets from all other routers, derive the network topology.

## *Discussion and Trade-Off Problems*

- **Two distributed applications A1 and A2 each run on multiple clients at the same time, but they have very different requirements. A1 is massively parallel and cares more about performance than about consistent sharing of data among multiple clients. A2 cares more about data consistency than about performance. If all client machines use NFS to access and update files, discuss and enumerate tradeoffs which (i) server caching, and (ii) client caching might be more appropriate for A1 and A2.**

## Design Problems

Assume a web-service (client/server) where clients and web server communicate over Internet using HTTP/TCP/IP. Assume that the initial web-server design has a high failure rate. Design a robust web-service so that its failure rate becomes very low (close to 99.999%). Discuss your design decisions.

## Comparison Problems

- Provide three different authentication approaches between Alice and Bob and compare them with respect to their network performance (message overhead).

## Course Evaluation

- Consider ICES online!!! (First Time)  for all students (CS and ECE)
- Online ICES are available to students from 11/27/08 until 12/10/08.
- Fill out the ICES form and provide constructive feedback
- Be part of the redesigning the cs425 material

***Thank You***