

# CS 262

# Introduction to Distributed Computing

Jim Waldo  
Ian Rose

# Administrivia

- Staff
  - Jim Waldo
    - MD 339, MW, 2-3:45 or by appointment
    - jim.waldo@sun.com or waldo@eecs.harvard.edu
  - Ian Rose
- Classes
  - M,W 4-5:30
  - MD G135
  - In case of bad weather, check class web site
    - <http://eecs.harvard.edu/cs262>

# Distributed Computing

- A Distributed System is characterized by
  - Multiple machines
  - Connected by a network
  - Cooperating on some task
- ...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

# Not Networking

- Networking is worried about
  - Sending a message from here to there
  - Not what you do with the message
- Distributed Computing
  - Assumes that there is some way to send the message
  - Worries about the properties of those messages
  - How you build a system using those messages

# Some Problems

- Sharing
  - Data
  - Processing
  - Consistency
- Finding
  - Where are things
  - Once found, how used?
- Programming models

# Failure

- Non-distributed system
  - Total failure
  - When failure happens, you know
  - Failure strategy: restart
- Distributed system
  - Partial failure
  - Unknown failure
  - Failure strategy.....

# Electronic Funds Transfer

- Four stages
  - Payment is sent
  - Payment is received
  - Payment credited
  - Receipt returned
- Failure points
  - You send, payment not received
  - You send, payment received, payment not credited
  - You send, payment received, payment credited, return receipt fails
- If a failure occurs, how can you tell what it was?
  - It matters...

# System Design

- Sometimes, it is how you design the interfaces
- Obvious funds transfer interface
  - ```
newBalance = transfer(double amount,  
                      account source,  
                      account dest)  
throws InsufficientFundsException;
```
- Better interface
  - ```
newBalance = transfer(double currentBalance,  
                      double amount,  
                      account source,  
                      account dest)  
throws InsufficientFundsException;
```

# It's Even Worse

- Problem: Design an algorithm that
  - Deals with two processors, a and b
  - Connected by an asynchronous network
  - Communicating by passing messages
  - Some messages may be lost
  - Guarantees a and b agree on some value

**It Can't Be Done**

# The Eight Fallacies

- The network is reliable
- Latency is zero
- Bandwidth is infinite
- The network is secure
- Topology doesn't change
- There is only one administrator
- Transport cost is zero
- The network is homogeneous
  - Peter Deutsch and James Gosling

**Why Bother?**

**It's Interesting**

**It's Good for You**

# Better Designs

- Interface design
  - One of the reasons we will use Java
- Easy to change implementations
- Isolation is easily enforced
- Protocol design

**Scale**

**Reliability**

**You Have To**

**You Will**

# Where the work is

- Internally scaled systems
  - Amazon
  - Google
  - EBay
  - Facebook
- Externally scaled systems
  - Grids
  - Web

# Your Choice

- Hope they go away
  - They won't
- Figure them out yourself
  - This hurts
- Learn where the problems are (and how to cope)
  - This class

# Class Organization

- Theory AND Practice
  - Mondays for theory
  - Wednesdays for practice
- Short papers
- Programming problems
- Exams (2)
- Final Project

# Grades

- Exams : 30%
- Final Project : 30%
- Short Papers : 15%
- Programming Assignments : 20%
- Class Participation : 5%