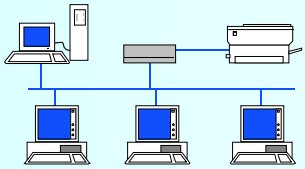


# Distributed Systems
# 600.437
## Introduction

Department of Computer Science
The Johns Hopkins University



# Course Information

- Lecture : Wednesday 3pm - 5pm, Shaffer 303.
- Tutorial: Monday 3pm - 4pm, Shaffer 303.
- E-mail: yairamir@cs.jhu.edu, jonathan@cs.jhu.edu
- Phone : 410-516-4803 410-516-5562
- Office hours: NEB-316/313 Monday 10:30 - 11:30
- TA : Yu Qiu yuqiu@cs.jhu.edu
- Office hours: NEB-332 Tuesday 10:30 - 11:30

E-mail is best. Come to office if necessary. Call only if really really needed.

## Grading Policy

- No exam.
- A few written assignments.
- Three programming assignments.
- One final project.

20-25%+ 45-50% + 35-30% = 100%

The difference between 337 and 437 :

Depth of the exercises and the project.

There is no difference in grading.



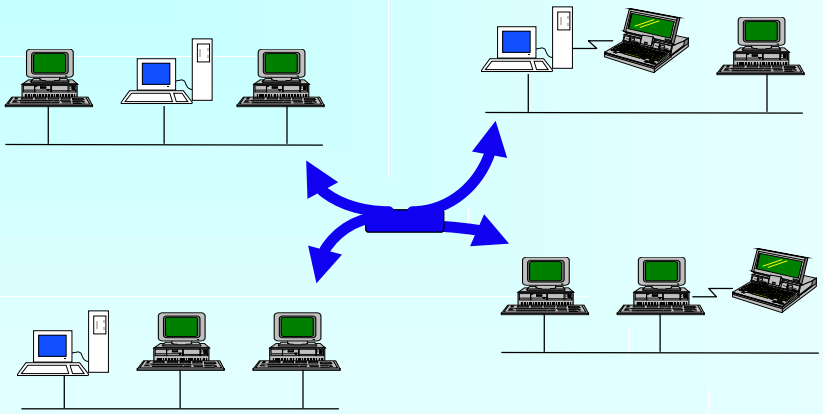
# Course Overview

## Lecture 1

Recommended reading: Building Secure and Reliable Network Applications
Kenneth P. Birman
Manning Publications

http://www.cnds.jhu.edu/courses/cs437/

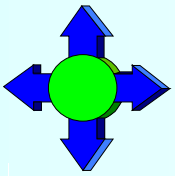
# A Distributed System





# Why Distribute?

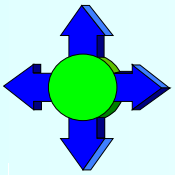
- Bridge geographic distances.
- Improve performance.
- Improve availability.
- Maintain autonomy.
- Lower the cost.
- Allow for interaction.



## Why Distribute?

?

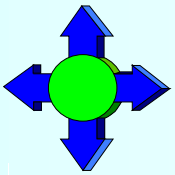
- Bridge geographic distances.
- Improve performance.
- Improve availability.
- Maintain autonomy.
- Lower the cost.
- Allow for interaction.





## Why Distribute?

- Bridge geographic distances.
- Improve performance.
- Improve availability.
- Maintain autonomy.
- Lower the cost.
- Allow for interaction.



Software solutions are needed!

Shaffer 303

## Course Content

- Basic communication protocols. Sep 20
- Synchronous models in distributed environments. Sep 27
- Multicast protocols. Oct 4
- Group communication. Oct 11
- Asynchronous models in distributed environments. Oct 18
- Distributed Operating Systems Oct 25
- Distributed transactions. Nov 1



Shaffer 100

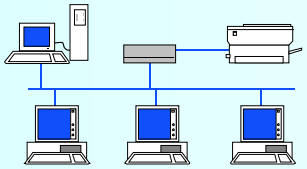
## Course Content

- Replication. Nov 8
- Metacomputing. Nov 15
- Advanced stuff Nov 22 (thanksgiving week)
- Distributed Objects Nov 29
- Knowledge in Distributed Systems Dec 6.

More or less...

Shaffer 303

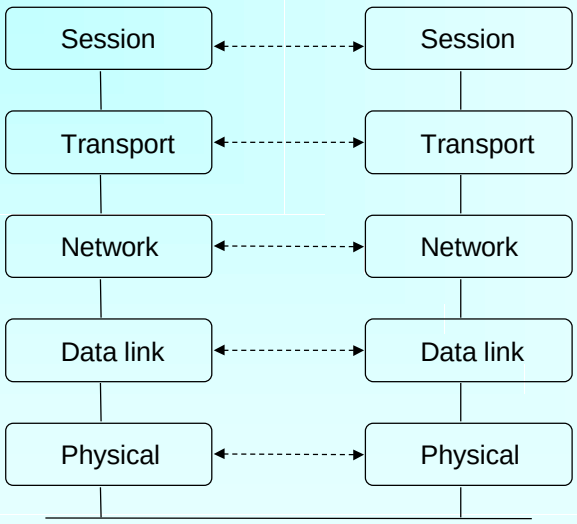
# Tutorial



- Presenting and discussing exercises.
- Communication using Unix sockets:
  - Reliable point-to-point communication (TCP/IP).
  - Non-reliable p-to-p communication (UDP/IP).
  - Non-reliable Broadcast, Multicast (UDP/IP).
- The Mosix system.
- The Spread system.
- Related research at the Center for Networking and Distributed Systems.

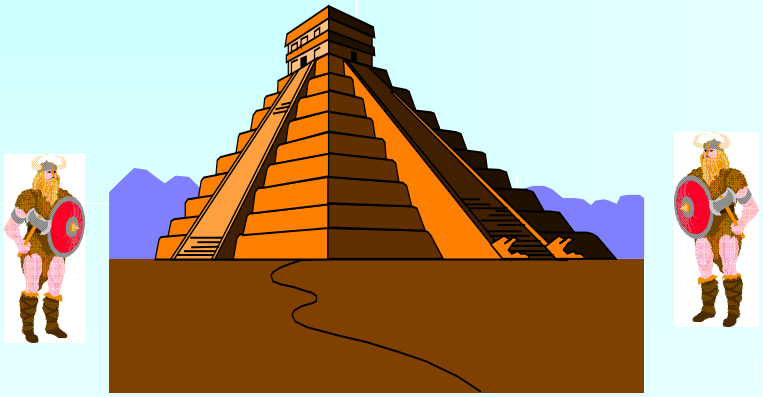


# Basic Communication Protocols



# Synchronous and asynchronous models.

Some Theory :)

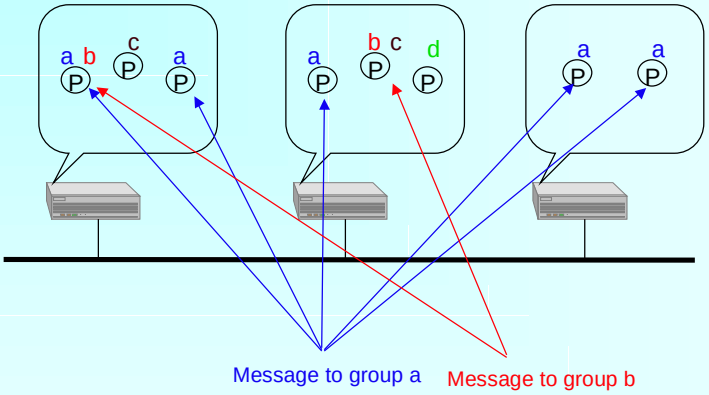




# Multicast Technology

- Reliability
- Ordering
- Membership
- Network span
- Routing
- Quality of Service

# Group Communication



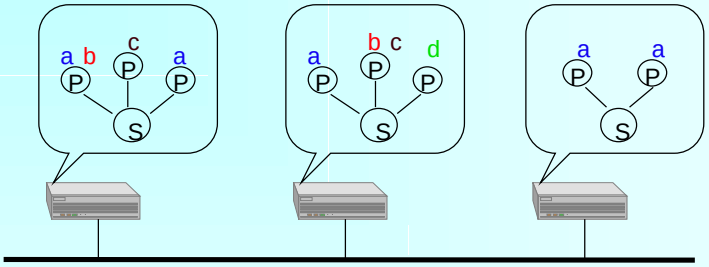



# Spread: A Group Communication Toolkit

Spread provides:

- Process groups across local and wide area networks (Internet).
- Consistent service semantics:
  - Various levels of reliability
  - Various levels of message ordering
  - Membership services.
- High performance.

# Process groups in Spread

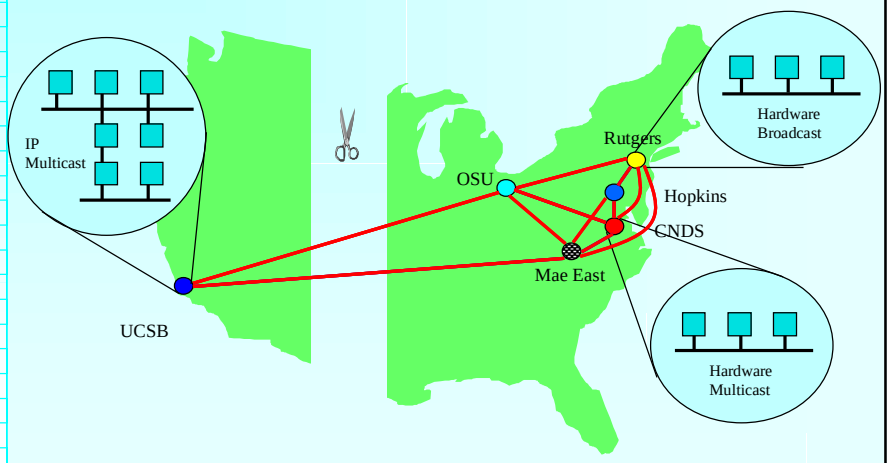



- One Spread daemon in each machine
- Multiple destination groups per message



# Deploying Spread

- Overlay networks concept with smart protocols



## Distributed Databases

When data is sparse over several database servers, there should be a way to coordinate transactions so that they will be:

- Atomic - either all effects take place, or none.
- Consistent - correct.
- Isolated - as if there was one serial database.
- Durable - effects are not lost.



## Replication

Considerations:

- Improve availability
- Improve performance for queries (higher throughput, lower latency).
- Cost

How to do it efficiently?

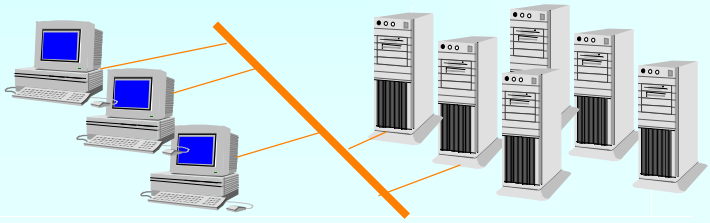
# Distributed Operating Systems

- File system
  - **Stateless / Stateful.**
  - **How available.**
  - **What is the semantics.**
- Processes
  - **Load balancing (static/dynamic).**
- Communication
  - **RPC.**
  - **Group communication.**



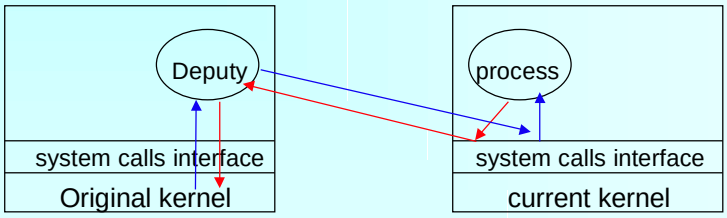
# Cluster Computing with Mosix

- Mosix (from Hebrew U): Kernel level enhancement to Linux that provides dynamic load balancing in a network of workstations.
- Dozens of PC computers connected by local area network (Fast-Ethernet or Myrinet).
- Any process can migrate anywhere anytime!!
- Transparent to the user!!



# Mosix Architecture for Process Migration

**The process migrated from the original computer to the current computer and now is initiating a system call.**

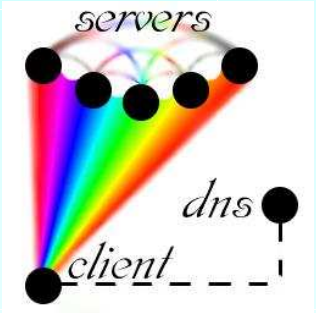



- Each process maintains a deputy in the original initiating computer that helps resolve most system calls.

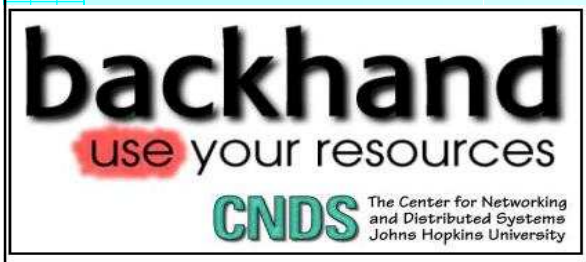


# Managing a Web Cluster with mod_backhand

- Developed here at Hopkins.
- Resource allocation according to the Cost-Benefit framework .
- The necessary "machinery" is implemented as an Apache module to minimize overhead.



# Backhand in Action




**Local Machine Name:** ice4.cnds.jhu.edu
**Apache Version String:**
Apache/1.3.9 (Unix)
balanced_by_mod_backhand/1.0.8
**Server built:** Jan 29 2000 19:29:45
**REMOTE_ADDR:** 128.220.221.24

| Entry | Hostname | Age | Address | Total Mem | Avail Mem | # ready servers/ # total servers | ~ms/req [#req] | Arriba | # CPUs | Load/HWM | CPU Idle |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | ice4.cnds.jhu.edu | 0 | 128.220.221.24:80 | 130764800 | 65651904 | 15/20 | 6695 [957] | 734832 | 1 | 3.700000/512 | 1.000000 |
| 1 | ice11.cnds.jhu.edu | 0 | 128.220.221.31:80 | 130764800 | 104056000 | 25/27 | 8122 [918] | 727283 | 1 | 4.630000/512 | 1.000000 |
| 2 | ice8.cnds.jhu.edu | 0 | 128.220.221.28:80 | 64380928 | 43451584 | 16/19 | 5459 [985] | 324316 | 1 | 3.470000/512 | 0.871000 |
| 3 | ice7.cnds.jhu.edu | 258949 | 128.220.221.27:80 | 130764800 | 74777792 | 0/48 | 6968 [313] | 840811 | 1 | 7.312000/512 | 0.000000 |
| 4 | ice12.cnds.jhu.edu | 0 | 128.220.221.32:80 | 130764800 | 67667136 | 0/19 | 8631 [952] | 475079 | 1 | 3.520000/512 | 0.000000 |
| 5 | ice6.cnds.jhu.edu | 0 | 128.220.221.26:80 | 64380928 | 50058432 | 10/10 | 7994 [987] | 308168 | 1 | 1.570000/512 | 1.000000 |
| 6 | ice5.cnds.jhu.edu | 0 | 128.220.221.25:80 | 263479296 | 151475392 | 15/37 | 4441 [580] | 1074345 | 2 | 5.810000/512 | 0.954000 |
| 7 | ice10.cnds.jhu.edu | 0 | 128.220.221.30:80 | 197156864 | 165516480 | 14/26 | 3805 [388] | 714895 | 2 | 4.760000/512 | 0.980000 |
| 8 | ice9.cnds.jhu.edu | 0 | 128.220.221.29:80 | 130764800 | 110281920 | 10/10 | 7637 [989] | 321603 | 1 | 1.980000/512 | 0.990000 |