CS550

• Distributed Operating Systems (Advanced Operating Systems)

• Instructor: Xian-He Sun
  – Email: sun@iit.edu, Phone: (312) 567-5260
  – Office hours: 2:10pm-3:10pm Tuesday, 3:30pm-4:30pm Thursday at SB229C, or by appointment

• TA: TBA
  – Email: tba@iit.edu
  – Office hours: TBA

• Blackboard:
  – http://courseinfo.iit.edu

• Class Web site
Outline

• Course information
• Key issues of distributed operating systems
• Hardware concepts
  – Multiprocessors
  – Multicomputers
  – Distributed systems
• Software concepts
  – Uniprocessor OS
  – Distributed OS
  – Network OS
  – Middleware
What This Course is About

• Understanding the *fundamental concepts* of distributed operating system, and distributed systems in general

• Learning *distributed programming* techniques
  – Multithreading, RPC, RMI, Sockets, etc.

• Understanding the *general principles* of distributed paradigms
  – MPI, JINI, NFS, Web Service, Grid, etc.
Prerequisite

- CS450 “Operating Systems”
- Familiar with
  - Programming in C/C++ or Java
  - UNIX tools and development environment
    - Command
    - Editors (vi, emacs), compilers (gcc), makefiles (GNU make)
  - Networking programming
    - Sockets
    - Multithreaded
    - RPC, Java RMI
  - Basic concepts of computer architecture
Course Materials

• Required:

• Recommended:

• Supplemental readings
Misc. Course Details

• You are expected to attend all of the lectures and presentations
• Grading
  – written and programming assignments (35%): individual work
  – One exam (35%)
  – Final project (30%): individual or group with 2-3 students
• Use the course blackboard
  – Announcements
  – Lecture notes
  – Assignments
  – Discussion
  – ...

X.Sun (IIT) CS550: Distributed OS Lecture 1 Page 6
Policies

• Collaboration
  – Encouraged for high level concepts and understanding the courses materials
  – but ..... 

• Cheating
  – Copying all or part of another student's homework
  – Allowing another student to copy all or part of your homework
  – Copying all or part of code found in a book, magazine, the Internet, or other resource
Any Questions?
Personal Introduction

• Research interests
  – Middleware
  – Performance Analysis and Modeling
  – Pervasive Computing
  – Scientific High Performance Computing

• Research group:
  – Scalable Computing Software Laboratory (SCS)
  – http://www.cs.iit.edu/~scs/
  – Weekly Research seminar
Distributed Computing at SCS

Many workstations are made available for graduate students
Scalable Computing Software (SCS) Lab.

Parallel Computers at SCS

Distributed Optical Testbed (Grid)

Pervasive Computing Environments at SCS
Evolution of Computing

Bigger becomes even bigger
Smaller becomes ever smaller, & connected

Japan’s Earth Simulator
• 640 processor nodes (PNs)
• Each PN is a system with 8 vector-type arithmetic processors (APs)
• Peak performance 40Tflops

approx; 50m x 65m x 17m

1.4m x 1m x 2m
Embedded Systems: What is the new

- Devices become smaller and more powerful
- Devices are coordinated via network
- From “autonomous computing” to coordinated “human-center computing”
Pervasive Computing

MIT’s view of pervasive computing
**Pervasive Computing**

- **Cross network (PSTN phone, internet) Service**
- **Scarlet framework for context aware computing**
- **Mobility**
The IIT HawkTour

Pushing the Boundaries of Pervasive Computing

• Provides tour information based on location and orientation
• Location-aware with WiFi positioning and GPS
• Intelligent content delivery
• Scarlet is used to provide context awareness
HawkTour cont

- Location Awareness
  - Tracking and Web Services
- Application Design
  - User Interface
- Content
  - Information relevant to the HawkTour
Distributed Computing: What is the new

- Supercomputers become ever powerful
- Communities of “Virtual organizations” are formed
- No VO possesses all required skills and resources
- From “community sharing” to “information grid”
Integrated VOs: the Grid

Mimic the electrical power grid

- Higher Quality of Service
- Increased Productivity
- Improved Resiliency
- Increased Efficiency
- Reduced Complexity & Cost
The Challenge of Grid Computing

Virtualization and Resource Management

Many sources of data, services, computation

Registries organize services of interest to a community

Security & policy must underlie access & management decisions

Resource management is needed to ensure progress & arbitrate competing demands

Data integration activities may require access to, & exploration/analysis of data at many locations

Exploration & analysis may involve complex, multi-step workflows

Virtualization and Resource Management

Security service

Policy service

Discovery

Access

R

RM

RM

RM

RM

RM
Virtual Private Grid (VP-Grid)

• A hosting platform where each user can create and operate in a private grid(s), based on the same shared Grid infrastructure, achieving:
  – Virtualization
  – Isolation and Protection
  – Privacy
  – Accountability and QoS
  – On-demand creation and provisioning
Overview of VP-Grid

VP-Grid Host (physical)

VPS

Virtual service node

VPS'

X.Sun (IIT)  CS550: Distributed OS  Lecture 1 Page 22
Virtualization: Key Technique

- **Two-level OS structure**
  - Host OS
  - Guest OS

- **Strong isolation**
  - Administration isolation
  - Installation isolation
  - Fault / attack isolation
  - Recovery, migration, and reconfiguration

- **Virtual service node**
  - VP-Grid Service (VPS)
  - Guest OS
  - Internetworking enabled

One VP-Grid host
Resource Management & Task Scheduling

- VP-Grid provider selection:
  - Among a set of VP-Grid providers, which one should be chosen to host an VPS?

- VPS selection:
  - Among a set of potential tenants (VPSes), which ones to host? (for QoS, resource utilization, security…)

- The Grid Harvest Service (GHS) System
  - A long-term application-level performance prediction and task scheduling system for non-dedicated distributed (Grid) environments
  - Reservation-based versus shared resources
Rescheduling Algorithm

1. Measure the prediction error of the system utilization, $PU(k)$

2. $PU(k) > \text{threshold}?$

3. NO

4. Find the best machine or machine set for task reallocation

5. Calculate the expectation of $T(\text{reassign})$ and $T(\text{original})$: $E(R)$ and $E(O)$

6. $E(O) - E(R) > 0?$

7. NO

8. Task Reallocation

9. Running application until next monitor period

10. NO

11. Measure the prediction error of the system utilization, $PU(k)$
Mobility of VP-Grid

• Mobility is needed for dynamic scheduling of tasks and reconfiguration of the VPS
• Current successes of mobile computing are based on safe-languages such as Java, which is slow and cannot apply to legacy codes
• VP-Grid supports mobility at two-levels, at the virtual machine level and at the code level
• VP-Grid supports mobility of legacy codes written in traditional languages such as Fortran, C, C++
Mobility of Legacy Code

- We have developed novel methodologies and a prototype system, **HPCM**, to migrate codes written in traditional languages such as Fortran, C, C++

- Two level mobility: migrate native codes under Java virtual machine
- General methods: migrate between different computing systems and different virtual organizations.
- Leading technology, strong mobility
Any Questions?