Distributed systems

- A collection of autonomous computers linked by a network, with software designed to produce an integrated computing facility
  - A well designed distributed system should be perceived as a single, integrated computing facility
- The introduction of LANs at the beginning of the 1970s triggered the development of distributed systems

Key Characteristics

- There are 6 key characteristics of a distributed system
  - Support for resource sharing
  - Openness
  - Concurrency
  - Scalability
  - Fault Tolerance
  - Transparency

Resource Sharing

- Resource
  - The range of things that can be shared usefully in a distributed system
  - Includes things like
    - Printers, disks, cdroms, etc.
    - Data
- A resource manager is a software module that manages a set of resources of a particular type
Openness

• The characteristic that a system can be extended in various ways
  – Hardware extensions
  – Software extensions
• Historically, computer systems were largely closed
  – UNIX broke that mold

Concurrency

• Concurrency in a distributed system does not necessarily mean concurrency within a single program
  – Many users invoke similar commands
  – Many different server processes may be running
• Synchronization, of course, is a problem

Scalability

• Distributed systems operate at many different scales
  – Two workstations and a file server
  – The CS computers…
• Often the more important question is not can you scale, but can you scale well
  – Consider the Internet
Fault Tolerance

- Fault tolerance is the ability of a system to recover when failures occur
  - Required in some systems
  - Nuisance value in others
- Fault tolerant design is based on two approaches
  - Hardware redundancy
  - Software recovery

Hardware Redundancy

- Two computers are employed for a single application, one acting as a standby
  - Very costly, but often very effective, solution
- Redundancy can be planned at a finer grain
  - Individual servers can be replicated
  - Redundant hardware can be used for non-critical activities when no faults are present

Software Redundancy

- Software must be designed so that the state of permanent data can be recovered or “rolled back” when a fault is detected
  - Transaction processing
Transparency

• The concealment of the separation of components in a distributed system
  – Access transparency
  – Location transparency
  – Concurrency transparency
  – Replication transparency
  – Failure transparency
  – Migration transparency
  – Performance transparency
  – Scaling transparency

Transparency

• The two most important forms of transparency are
  – Access
  – Location
• They are sometimes referred to as network transparency
  – rlogin command
  – ptt@cs.rit.edu

Basic Design Issues

• Design issues that arise specifically from the distributed nature of systems include
  – Naming
    • How do you assign names to resources
  – Communication
  – Software structure
    • Openness
  – Workload allocation
How is this different?

- Resource sharing – yes
- Openness – yes
- Concurrency – absolutely
- Scalability – would love it
- Fault tolerance – maybe
- Transparency – definitely not