Introduction to Distributed Systems

- **Stand-alone computing systems**
  - Independent computers
  - Independent tasks

- **Networked computing systems**
  - Interconnected independent computers
  - Processes of independent tasks can communicate

- **Distributed computing systems**
  - Loosely-coupled computers
  - Processes of individual tasks transparently share resources

- **Parallel computing systems**
  - Tightly-coupled processing units
  - Several processes cooperate on a single task
A New Perspective

Computing systems are evolving to a merge
- Embedded systems were once stand-alone
  - Now modern limousines are distributed systems on wheels
- Workstations were once networked systems
  - Now they use parallel hardware (processors and SMPs)
  - Now transparency is being increased (Gnutella)
- Distributed systems were once local
  - Now the web is the computer (SETI@Home)
- Parallel systems were once built on multiprocessors
  - Now clusters are made of off-the-shelf computers with high-speed buses and networks

Operating systems are being challenge
- Light enough to support a stand-alone system
- Powerful enough to support a distributed system
Distributed Systems

- Set of loosely coupled computers interconnected by a network
- Each computer has its own local resources plus remote resources from other computers in the set
- Processes on a distributed system access resources independently of whether they are local or remote (location transparency)

Process models
- Client-Server
  - Server has a resource that is used by the client
- Peer-to-Peer
  - Both partner processes share some of their resources
Motivation

- **Resource sharing**
  - Remote file sharing, printing, access to special devices (scanner, CD writer, etc)
  - Distributed databases

- **Computation speedup**
  - Tasks can be partitioned and distributed

- **Reliability**
  - The failure of a node does not necessarily disrupts the system

- **Scalability**
  - New nodes can be aggregated to the system on demand

- **Pitfalls:** complexity and security
Transparency

- Location transparency
  - Local and remote objects look just the same
  - No need to specify location
- Migration transparency
  - Objects change location, their names are preserved
- Replication transparency
  - Objects can be automatically replicated (consistency)
- Concurrency transparency
  - Objects can be concurrently manipulated without explicit synchronization
- Parallelism transparency
  - Automatic parallelization
Remote Procedure Call (RPC)