Distributed System

THOAI NAM
Chapter 1: Introduction

- Distributed Systems
- Hardware & software
- Transparency
- Scalability
- Distributed OS
Definition of a Distributed System

- A distributed system:
  - Multiple connected CPUs working together
  - A collection of independent computers that appears to its users as a single coherent system
- Examples: parallel machines, networked machines
Advantages and Disadvantages

- Advantages
  - Communication and resource sharing possible
  - Economics – price-performance ratio
  - Reliability, scalability
  - Potential for incremental growth

- Disadvantages
  - Distribution-aware PLs, OSs and applications
  - Network connectivity essential
  - Security and privacy
## Transparency in a Distributed System

<table>
<thead>
<tr>
<th>Transparency</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access</td>
<td>Hide differences in data representation and how a resource is accessed</td>
</tr>
<tr>
<td>Location</td>
<td>Hide where a resource is located</td>
</tr>
<tr>
<td>Migration</td>
<td>Hide that a resource may move to another location</td>
</tr>
<tr>
<td>Relocation</td>
<td>Hide that a resource may be moved to another location while in use</td>
</tr>
<tr>
<td>Replication</td>
<td>Hide that a resource may have many copies</td>
</tr>
<tr>
<td>Concurrency</td>
<td>Hide that a resource may be shared by several competitive users</td>
</tr>
<tr>
<td>Failure</td>
<td>Hide the failure and recovery of a resource</td>
</tr>
<tr>
<td>Persistence</td>
<td>Hide whether a (software) resource is in memory or on disk</td>
</tr>
</tbody>
</table>

**Different forms of transparency in a distributed system.**
Scalability Problems

<table>
<thead>
<tr>
<th>Concept</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized services</td>
<td>A single server for all users</td>
</tr>
<tr>
<td>Centralized data</td>
<td>A single on-line telephone book</td>
</tr>
<tr>
<td>Centralized algorithms</td>
<td>Doing routing based on complete information</td>
</tr>
</tbody>
</table>

Examples of scalability limitations.
Hardware Concepts: Multiprocessors (1)

- Multiprocessor dimensions
  - Memory: could be shared or be private to each CPU
  - Interconnect: could be shared (bus-based) or switched

- A bus-based multiprocessor.
Multiprocessors (2)

a) A crossbar switch     b) An omega switching network

(a) Crosspoint switch

(b) 2x2 switch
Homogeneous Multicomputer Systems

a) Grid

b) Hypercube

(a)

(b)
Distributed Systems Models

- Minicomputer model
  - Each user has local machine
  - Local processing but can fetch remote data (files, databases)

- Workstation model
  - Processing can also migrate

- Client-server Model
  - User has local workstation
  - Powerful workstations serve as servers (file, print, DB servers)

- Processor pool model
  - Terminals are Xterms or diskless terminals
  - Pool of backend processors handle processing
Uniprocessor Operating Systems

- An OS acts as a resource manager or an arbitrator
  - Manages CPU, I/O devices, memory
- OS provides a virtual interface that is easier to use than hardware
- Structure of uniprocessor operating systems
  - Monolithic (e.g., MS-DOS, early UNIX)
    » One large kernel that handles everything
  - Layered design
    » Functionality is decomposed into N layers
    » Each layer uses services of layer N-1 and implements new service(s) for layer N+1
Uniprocessor Operating Systems

Microkernel architecture
• Small kernel
• user-level servers implement additional functionality

No direct data exchange between modules

OS interface

System call

User application
Memory module
Process module
File module

User mode
Kernel mode

Microkernel

Hardware
Distributed Operating System

- Manages resources in a distributed system
  - Seamlessly and transparently to the user
- Looks to the user like a centralized OS
  - But operates on multiple independent CPUs
- Provides transparency
  - Location, migration, concurrency, replication, ...
- Presents users with a virtual uniprocessor
# Types of Distributed OSs

<table>
<thead>
<tr>
<th>System</th>
<th>Description</th>
<th>Main Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>DOS</td>
<td>Tightly-coupled operating system for multi-processors and homogeneous multicomputers</td>
<td>Hide and manage hardware resources</td>
</tr>
<tr>
<td>NOS</td>
<td>Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN)</td>
<td>Offer local services to remote clients</td>
</tr>
<tr>
<td>Middleware</td>
<td>Additional layer atop of NOS implementing general-purpose services</td>
<td>Provide distribution transparency</td>
</tr>
</tbody>
</table>
Multiprocessor Operating Systems

- Like a uniprocessor operating system
- Manages multiple CPUs transparently to the user
- Each processor has its own hardware cache
  - Maintain consistency of cached data
Multicomputer Operating Systems

Machine A

Distributed applications

Distributed operating system services

Kernel

Machine B

Kernel

Network

Machine C

Kernel
Network Operating System (1)

![Diagram showing network operating system](image)

- **Machine A**
  - Network OS services
  - Kernel

- **Machine B**
  - Network OS services
  - Kernel

- **Machine C**
  - Network OS services
  - Kernel

**Distributed applications**

**Network**
Network Operating System (2)

- Employs a client-server model
  - Minimal OS kernel
  - Additional functionality as user processes

![Diagram of client-server model](image-url)
Middleware-based Systems

- General structure of a distributed system as middleware.

![General structure of a distributed system as middleware](image-url)
## Comparison between Systems

<table>
<thead>
<tr>
<th>Item</th>
<th>Distributed OS</th>
<th>Network OS</th>
<th>Middleware-based OS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Multiproc.</td>
<td>Multicomp.</td>
<td></td>
</tr>
<tr>
<td><strong>Degree of transparency</strong></td>
<td>Very High</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Same OS on all nodes</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Number of copies of OS</strong></td>
<td>1</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td><strong>Basis for communication</strong></td>
<td>Shared memory</td>
<td>Messages</td>
<td>Files</td>
</tr>
<tr>
<td><strong>Resource management</strong></td>
<td>Global, central</td>
<td>Global, distributed</td>
<td>Per node</td>
</tr>
<tr>
<td><strong>Scalability</strong></td>
<td>No</td>
<td>Moderately</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Openness</strong></td>
<td>Closed</td>
<td>Closed</td>
<td>Open</td>
</tr>
</tbody>
</table>