Chapter 1
Introduction

Driving Forces

- Improvement in computer technology
  - Processor speed
  - Cost
- High-speed computer networks (LANs and WANs)
Definition of a Distributed System (1)

• A distributed system is:

  A collection of independent computers that appears to its users as a single coherent system

• Characteristics
  ➢ Heterogeneity hidden from users
  ➢ Consistent, uniform interaction regardless of location and time
  ➢ Expandable and scalable
  ➢ Availability

• Examples of distributed systems

Definition of a Distributed System (2)

Figure 1-1. A distributed system organized as middleware. The middleware layer extends over multiple machines, and offers each application the same interface.
Goals

• Connecting users and resources
  - Remote resources, sharing, and security

• Transparency
  - Hide the distributed nature of the system
  - Implications on performance

• Openness
  - Offers services according to standard rules that describe syntax and semantics of services
  - Interfaces and IDL (completeness and neutrality → Interoperability and portability)
  - Flexibility → Separating policy from mechanism

• Scalability
  - Size (more users and resources)
  - Geographical scalability (communication in LANs (synchronous) versus WANs)
  - Administrative scalability (conflicting policies across domains)

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 is replicated</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>
</tbody>
</table>

Figure 1-2. Different forms of transparency in a distributed system (ISO, 1995).
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>

Figure 1-3. Examples of scalability limitations.

Scalability Problems

Characteristics of decentralized algorithms:
- No machine has complete information about the system state.
- Machines make decisions based only on local information.
- Failure of one machine does not ruin the algorithm.
- There is no implicit assumption that a global clock exists.
Scaling Techniques 1/3

- Hiding communication latencies
  - Asynchronous communication
  - Interactive applications? (push front-end processing to client)
- Distribution
  - DNS
  - WWW
- Replication
  - Availability and load balancing
  - Consistency problems

Scaling Techniques 2/3

Figure 1-4. The difference between letting (a) a server or (b) a client check forms as they are being filled.
Scaling Techniques 3/3

Figure 1-5. An example of dividing the DNS name space into zones.

Pitfalls when Developing Distributed Systems

False assumptions made by first time developer:

- The network is reliable.
- The network is secure.
- The network is homogeneous.
- The topology does not change.
- Latency is zero.
- Bandwidth is infinite.
- Transport cost is zero.
- There is one administrator.
## Software Concepts

<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 multiprocessors 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>

## Multicomputer Operating Systems

![Multicomputer Operating Systems Diagram]
Network Operating System

Distributed System as Middleware

- Scalability and openness of NOS
- Transparency of DOS
- Mask heterogeneity of lower levels
Middleware Services

- High-level communication facilities
- Transparent access to remote data
- Naming (sharing and look up)
- Persistence
- Distributed transactions
- Security

Middleware and Openness

In an open middleware-based distributed system, the protocols used by each middleware layer should be the same, as well as the interfaces they offer to applications.