Distributed Systems

Thoai Nam

Faculty of Computer Science and Engineering

HCMC University of Technology

Chapter 1: Introduction

- Distributed Systems
- Hardware & software
- Transparency
- Scalability
- Distributed OS

Definition of a Distributed System

- What is a distributed system?
  - Multiple connected CPUs working together
  - A collection of independent computers that appears to its users as a single coherent system [Tanenbaum]
  - One in which components located at networked computers communicate and coordinate their actions by only message passing [Coulouris].

Examples of Distributed Systems

- Parallel machines, networked machines
- ***Cluster***: *"A type of parallel or distributed processing system, which consists of a collection of interconnected stand-alone computers cooperatively working together as a single, integrated computing resource" [Buyya]*
- ***Grid***: *"A type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements" [Buyya]*
- ***Cloud***: *"A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers" [Buyya]*

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

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

Different forms of transparency in a distributed system.

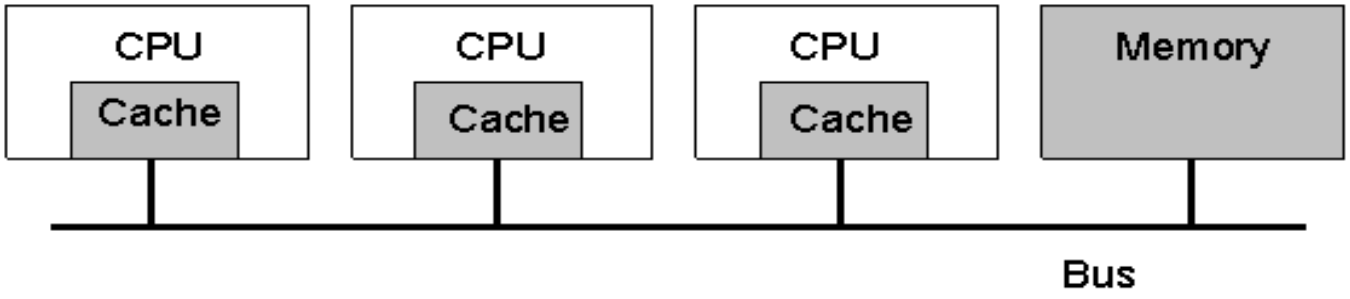
Scalability Problems

Concept	Example
Centralized services	A single server for all users
Centralized data	A single on-line telephone book
Centralized algorithms	Doing routing based on complete information

Examples of scalability limitations

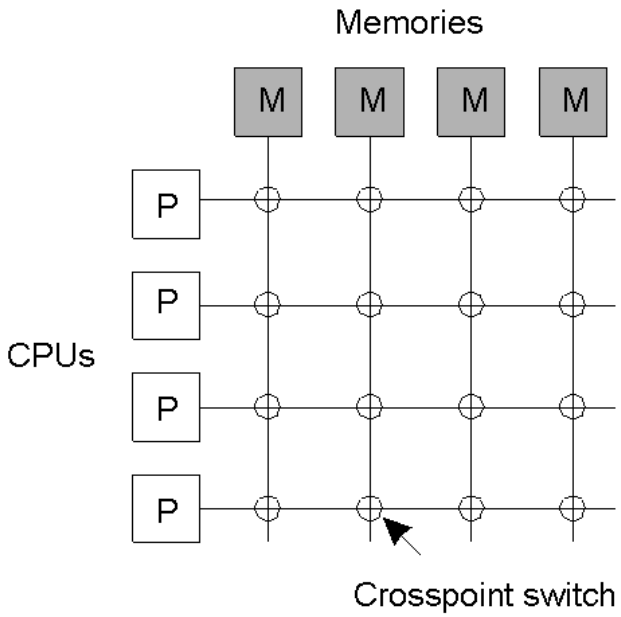
Hardware Concepts: Multiprocessors (1)

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

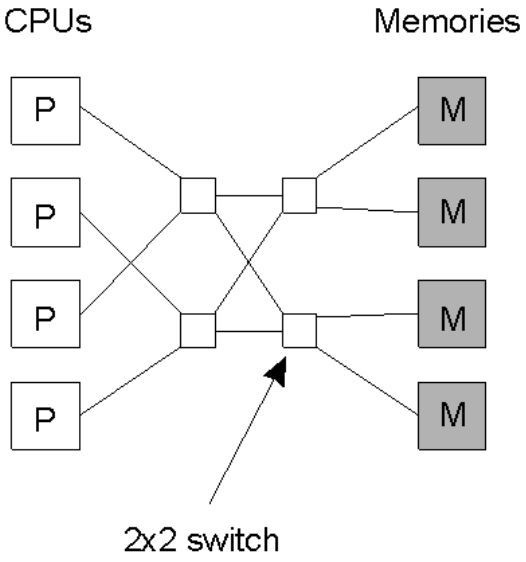


Multiprocessors (2)

a) A crossbar switch b) An omega switching network



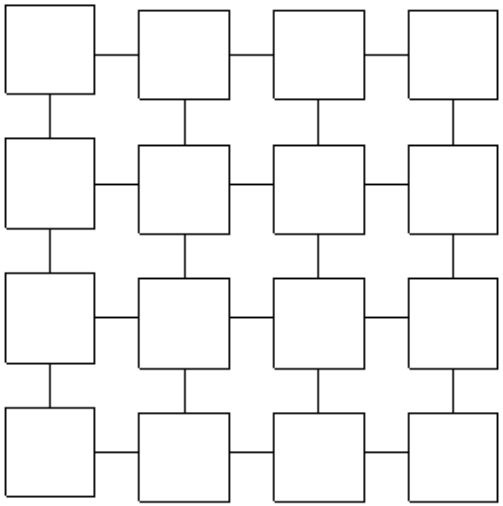
(a)



(b)

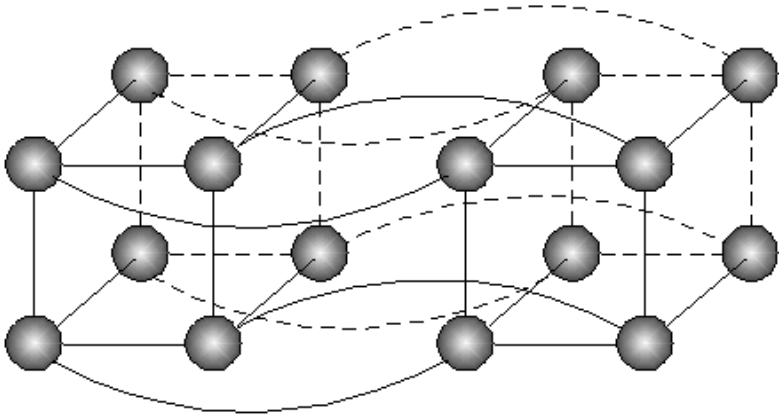
Homogeneous Multicomputer Systems

a) Grid



(a)

b) Hypercube



(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

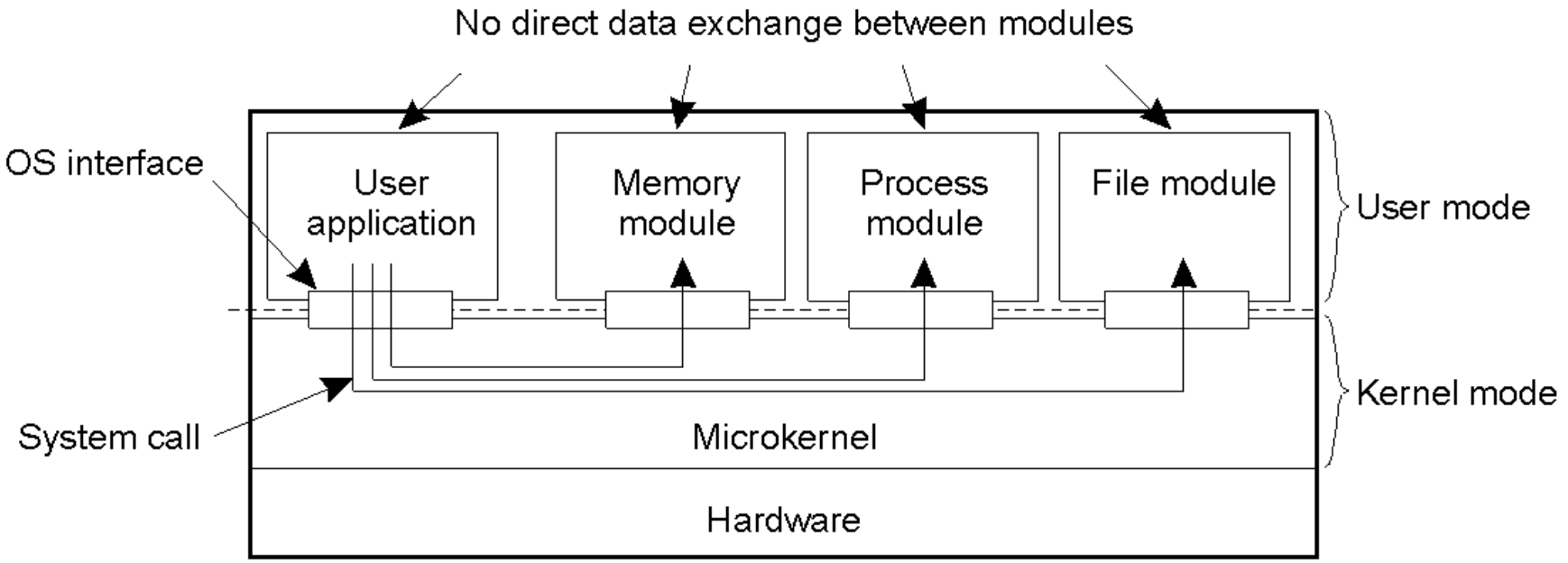
Uniprocessor Operating Systems

- 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



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

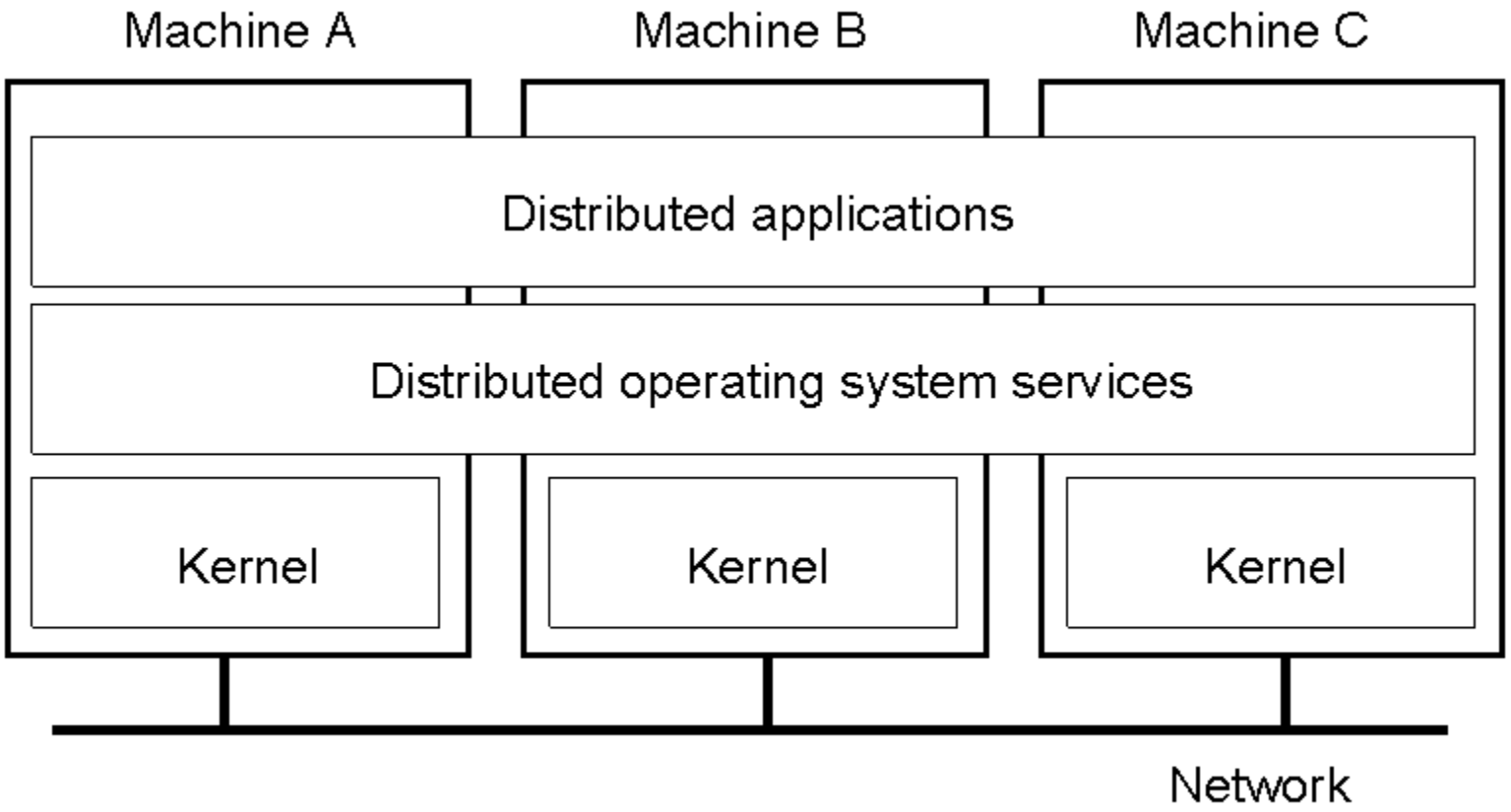
Types of Distributed OSs

System	Description	Main Goal
DOS	Tightly-coupled operating system for multi-processors and homogeneous multicomputers	Hide and manage hardware resources
NOS	Loosely-coupled operating system for heterogeneous multicomputers (LAN and WAN)	Offer local services to remote clients
Middleware	Additional layer atop of NOS implementing general-purpose services	Provide distribution transparency

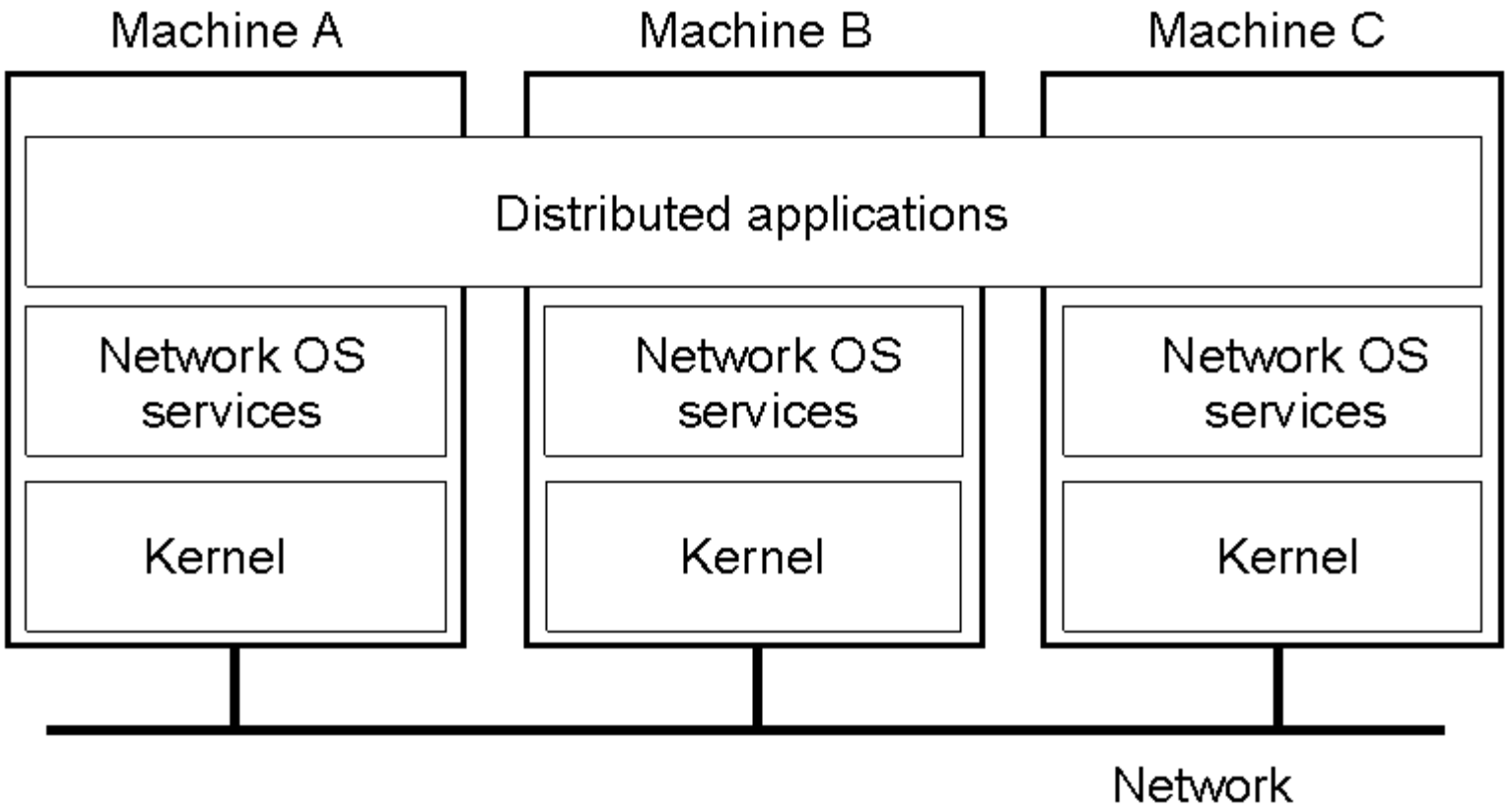
Multiprocessor Operating Systems

- 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

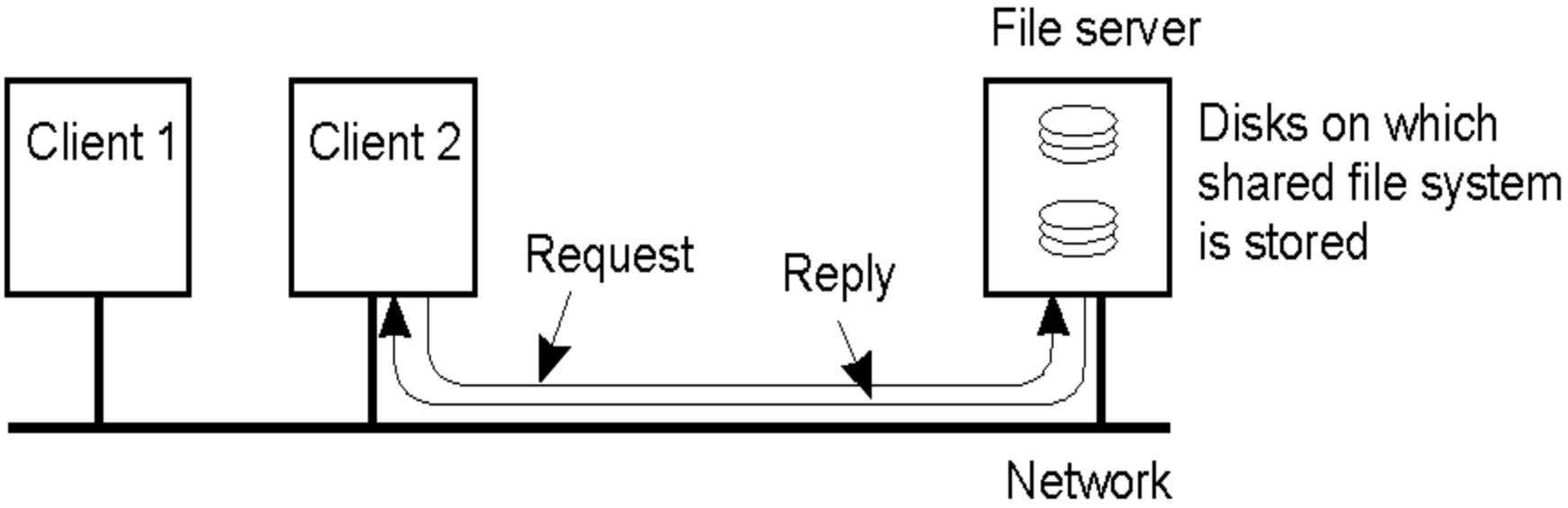


Network Operating System (1)



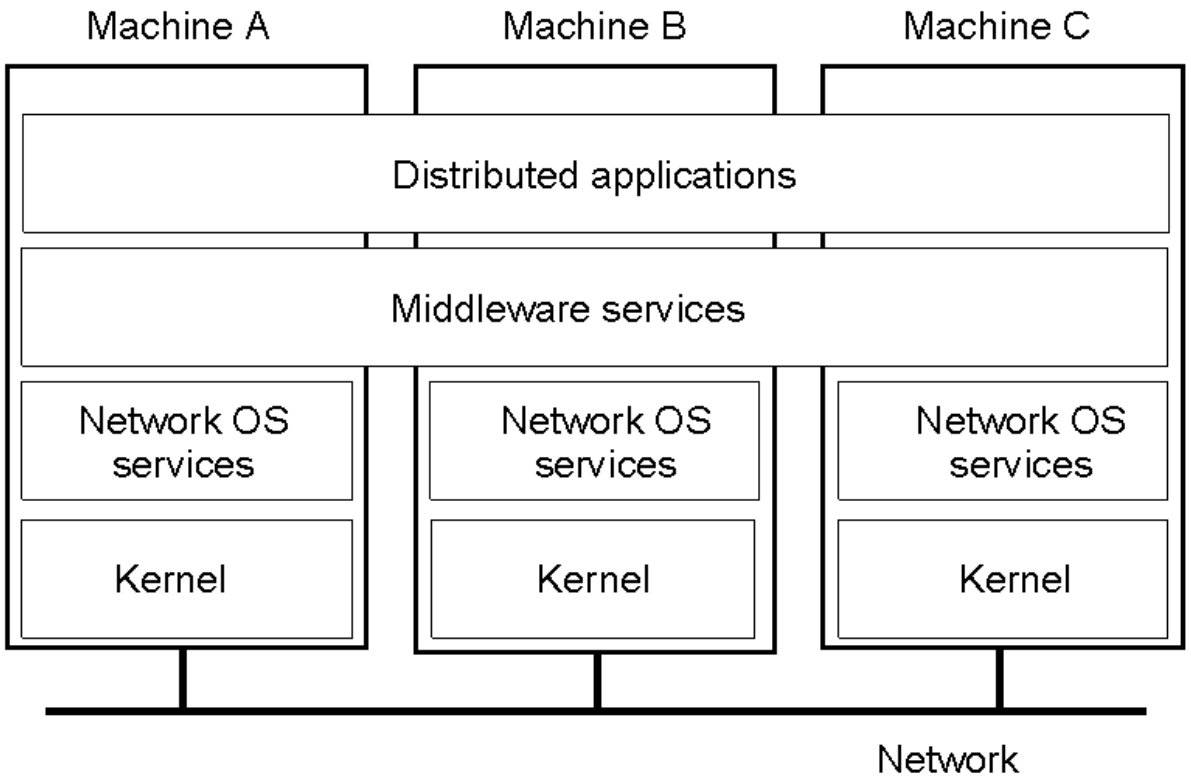
Network Operating System (2)

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



Middleware-based Systems

- General structure of a distributed system as middleware.



Comparison between Systems

Item	Distributed OS		Network OS	Middleware-based OS
	Multiproc.	Multicomp.		
Degree of transparency	Very High	High	Low	High
Same OS on all nodes	Yes	Yes	No	No
Number of copies of OS	1	N	N	N
Basis for communication	Shared memory	Messages	Files	Model specific
Resource management	Global, central	Global, distributed	Per node	Per node
Scalability	No	Moderately	Yes	Varies
Openness	Closed	Closed	Open	Open