现 代 操 作 系 统

Modern Operating Systems

宋虹 songhong@csu. edu. cn

中南大学计算机学院网络空间安全系

Chapter 7 Multiple Processor Systems

- **7.1 Multiprocessors**
- **7.2 Multicomputers**
- **7.3 Virtualization**
- **7.4 Distributed Systems**
- **7.5 Reading Materials**

Introduction

- The goals of introducing MultiProcessors
  - Want to get more and more computing power, but the solution that makes the CPU clock run faster is beginning to hit some fundamental limits on clock speed.
  - To increase the system throughput
  - To save money
  - To improve system reliability.

Introduction

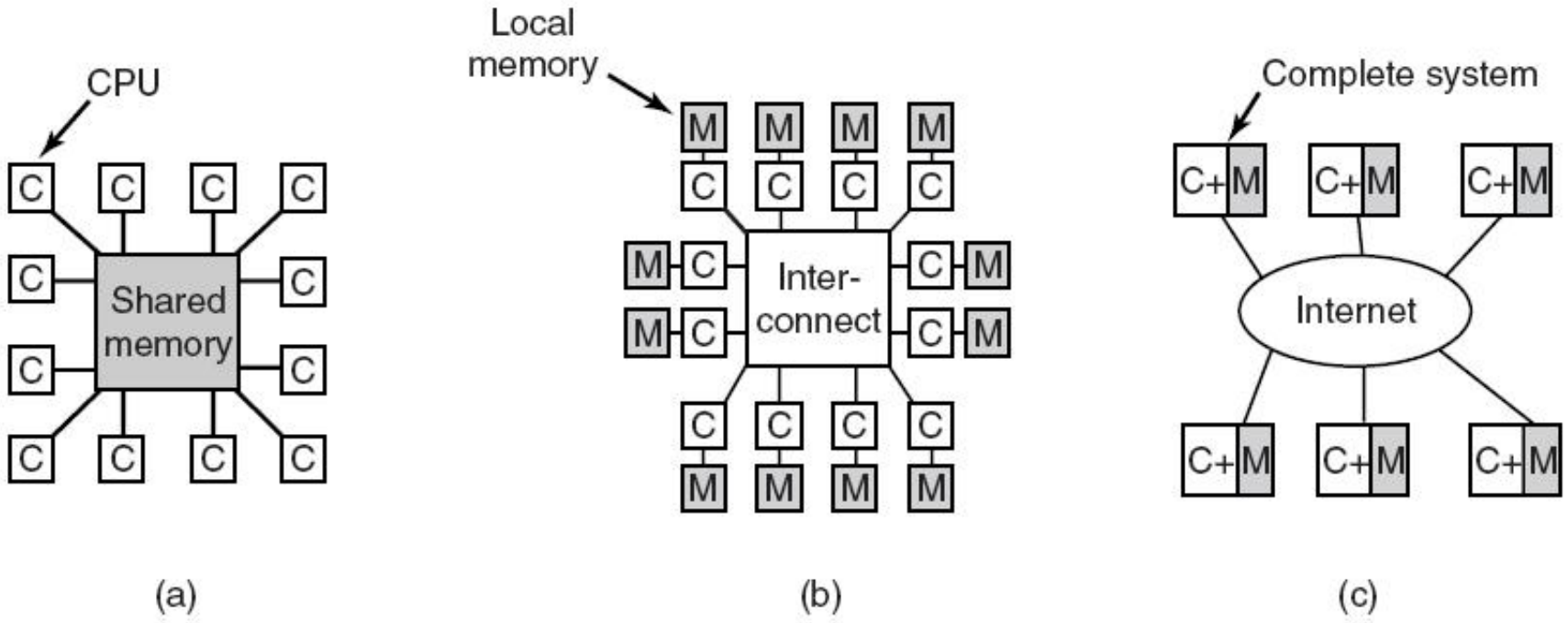


Figure 8-1. (a) A shared-memory multiprocessor. (b) A message-passing multicomputer. (c) A wide area distributed system.

Introduction

- Loosely coupled system---distributed system
  - Communication line or communication channel
  - Each computer can work independently
  - Message passing system
- Tightly coupled system---shared-memory system & message-passing multiomputer
  - high-speed interconnect or high-speed bus
  - Sharing memory and I/O devices

7.1 MultiProcessors

- Shared-memory multiprocessor---multiprocessor
  - A computer system in which two or more CPUs share full access to a common RAM
  - Regular operating system
  - Care its unique features in some areas
    - Process synchronization
    - Resource management
    - scheduling
- Hardware----UMA & NUMA
  - UMA: Uniform Memory Access
    - three structures
  - NUMA: Nonuniform Memory Access

7.1.1 UMA Multiprocessors with Bus-Based Architectures

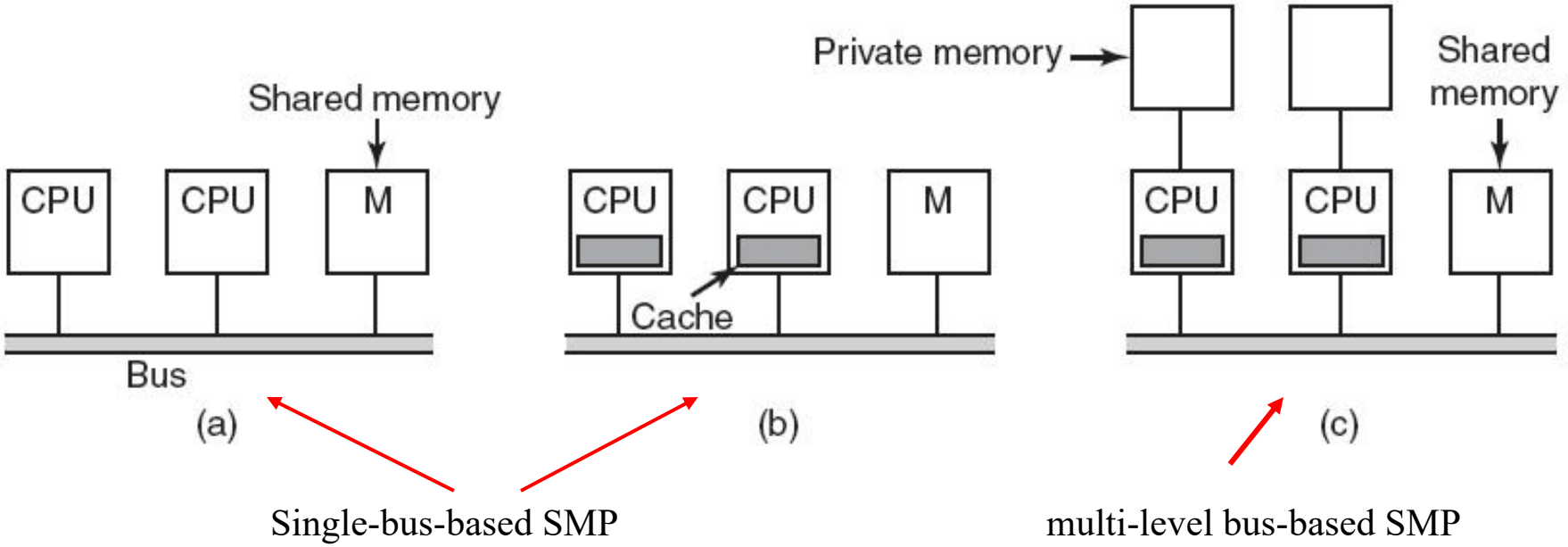


Figure 8-2. Three bus-based multiprocessors. (a) Without caching. (b) With caching. (c) With caching and private memories.

1. Single-bus based SMP

- Idea—two or more CPUs and one or more memory modules all use the same bus for communication.
- advantages
  - One copy of running OS;
  - Programs for single processor can be portable.
- Disadvantages
  - Limited scalability
  - Bottleneck: the bandwidth of the bus
  - Limited number of CPUs: 4-20
- Solutions: cache

2. multiple-bus based SMP

- Idea—each CPU has not only a cache, but also a local, private memory which it accesses over a dedicated bus.
- advantages
  - Can reduce the bus traffic, and support more CPUs: 16-32
- Disadvantages
  - Need active cooperation from the compiler

7.1.2 Multiprocessors using crossbar switchs

- Idea—use the simplest circuit (crossbar switch) for connecting *n* CPUs to *k* memories.
- advantages
  - Nonblocking network
  - No advance planning
  - Contention for memory can be reduced
- Disadvantages
  - The number of cross points grows large
  - The crossbar design is workable.
  - Can support medium-sized system with 8-16 CPUs

7.1.2 UMA Multiprocessors Using Crossbar Switches

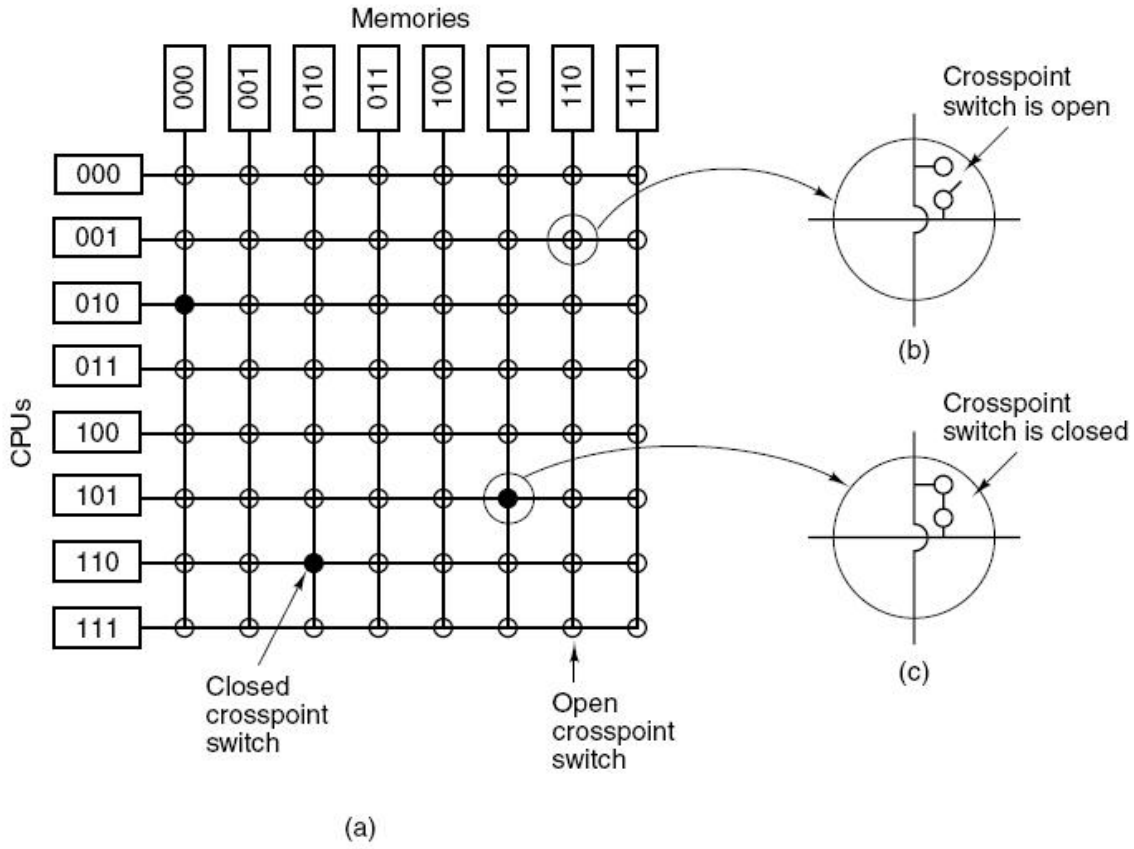


Figure 8-3. (a) An 8 × 8 crossbar switch. (b) An open crosspoint. (c) A closed crosspoint.

7.1.3 UMA Multiprocessors Using Multistage Switching Networks (1)

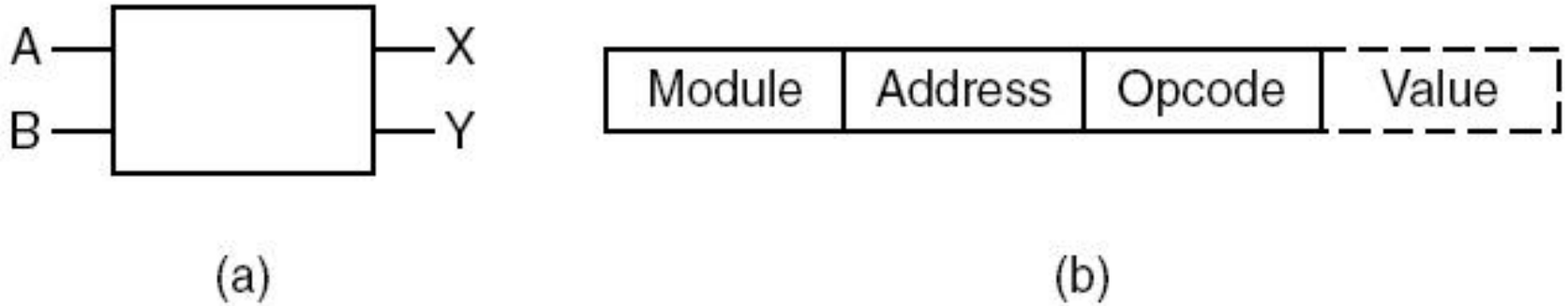


Figure 8-4. (a) A 2 × 2 switch with two input lines, A and B, and two output lines, X and Y. (b) A message format.

7.1.3 UMA Multiprocessors Using Multistage Switching Networks (2)

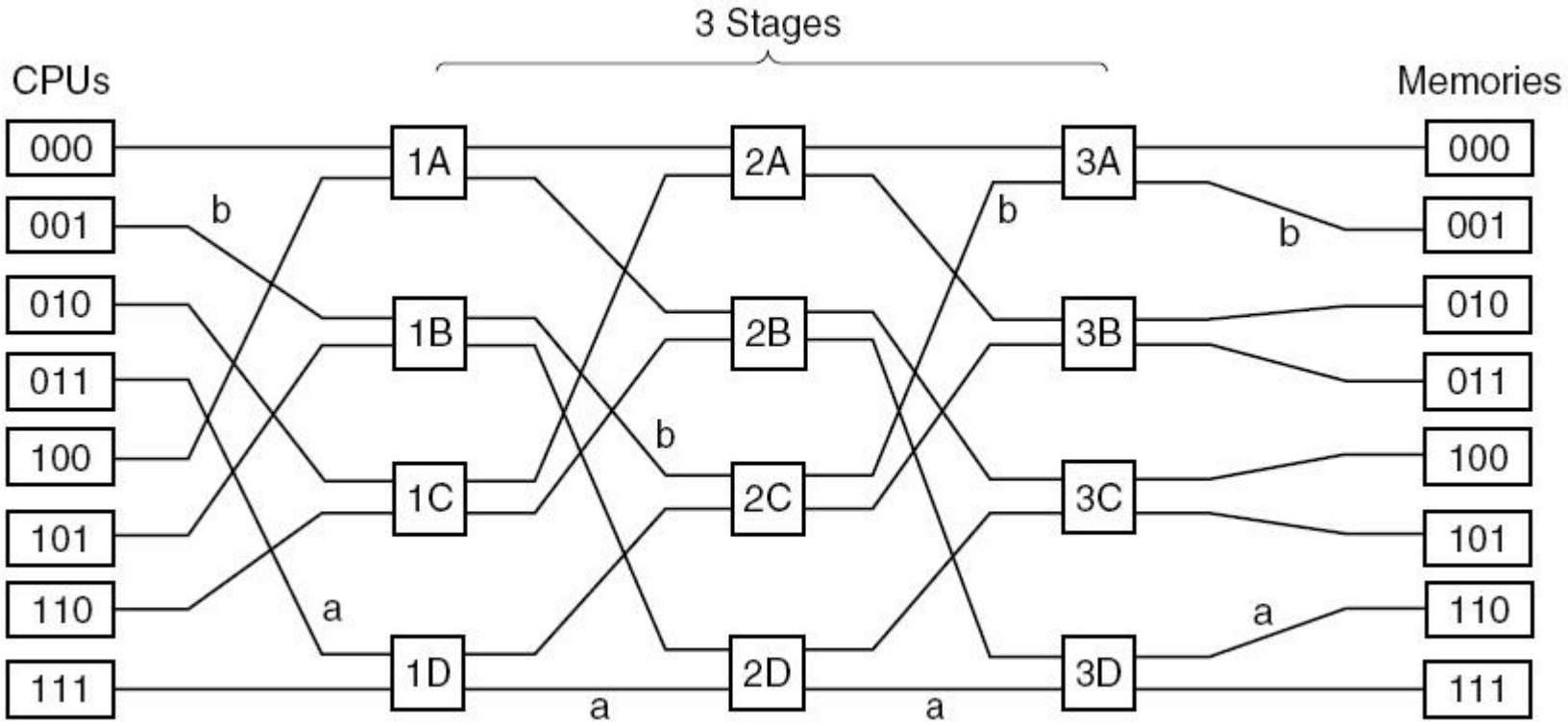


Figure 8-5. An omega switching network.

7.1.3 UMA Multiprocessors Using Multistage Switching Networks (3)

- Idea: larger multistage switching networks.
- Advantages
  - Offer multiple paths from each CPU to each memory module
  - Spread the traffic better
  - Maximize parallelism
- Disadvantages
  - Need a lot of expensive hardware
  - no more than 100 CPUs

7.1.4 NUMA Multiprocessors (1)

- Characteristics of NUMA machines:
  - There is a single address space visible to all CPUs.
  - Access to remote memory is via LOAD and STORE instructions.
  - Access to remote memory is slower than access to local memory.
- Two NUMA systems
  - NC-NUMA(No cache NUMA)
  - CC-NUMA(Cache-Coherent NUMA): directory-based multiprocessor

7.1.4 NUMA Multiprocessors (2)

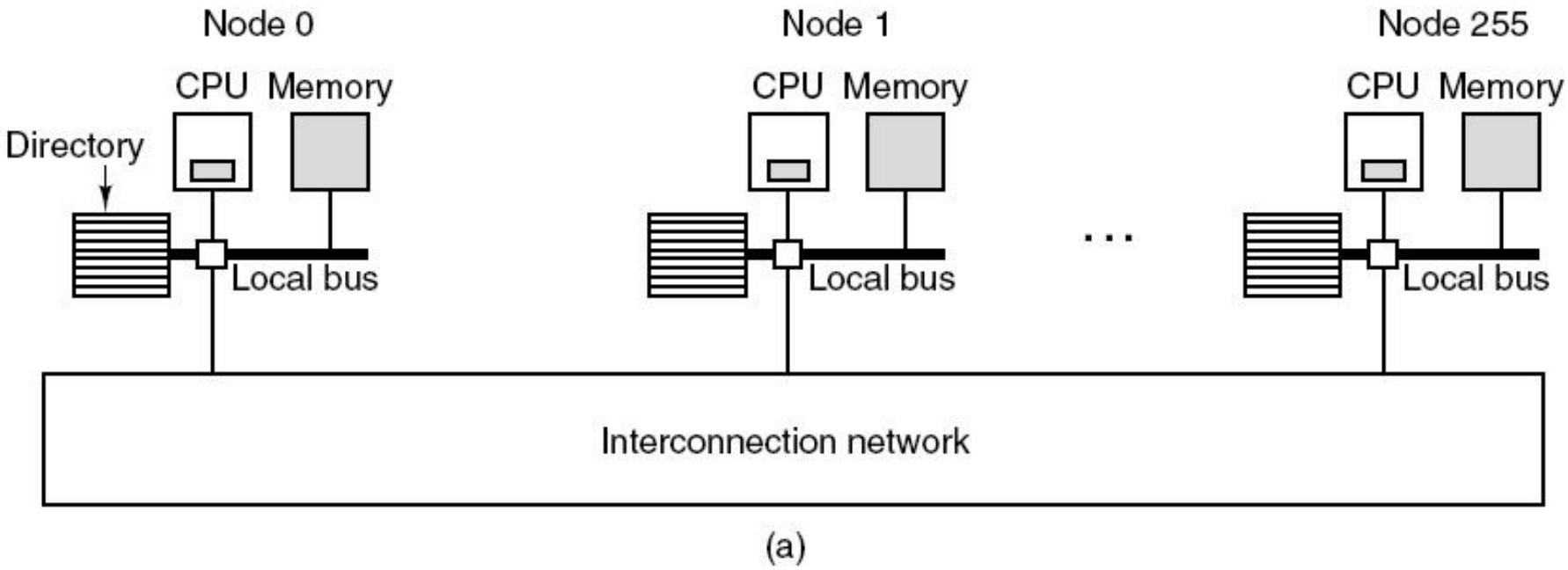
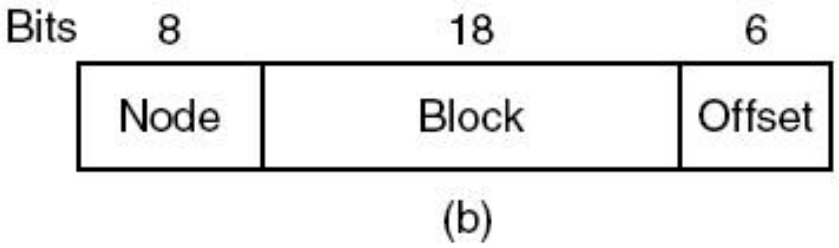


Figure 8-6. (a) A 256-node directory-based multiprocessor.

7.1.4 NUMA Multiprocessors (3)



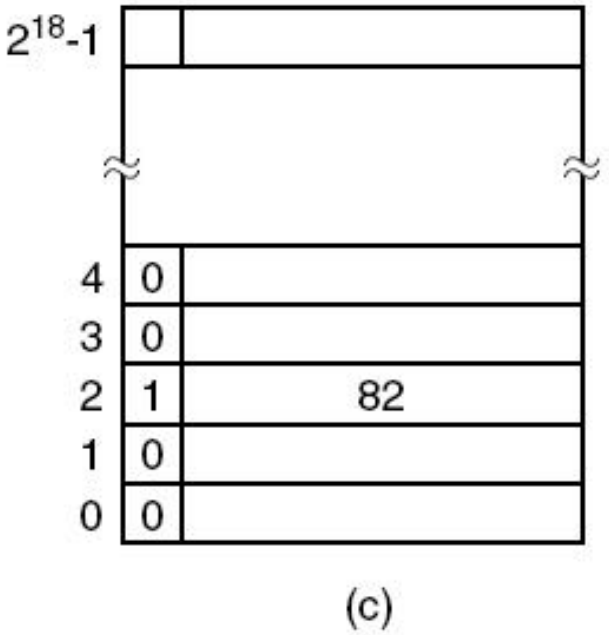


Figure 8-6. (b) Division of a 32-bit memory address into fields.
(c) The directory at node 36.

7.1.4 NUMA Multiprocessors (4)

- Features
  - All shared-memory are distributed in physical view, and be continuous in logical view.
  - Three levels for memory structure
    - Local memory
    - Shared memory
    - Global shared memory or memory for other node
- Disadvantage
  - A line can be cached at only one node, so we need some way of locating all of them.

7.1.5 Multiprocessor Operating System Types

- Features for Multiprocessor Operating System
  - Parallelism, Distribution, Synchronization, Reconfigurability
- Functions
  - Process Management, memory management, file system, System reconstruction
- Three approaches for multiprocessor software
  - Each CPU has its own operating system
  - Master-slave multiprocessors
  - Symmetric multiprocessors

7.1.5 Each CPU Has Its Own Operating System

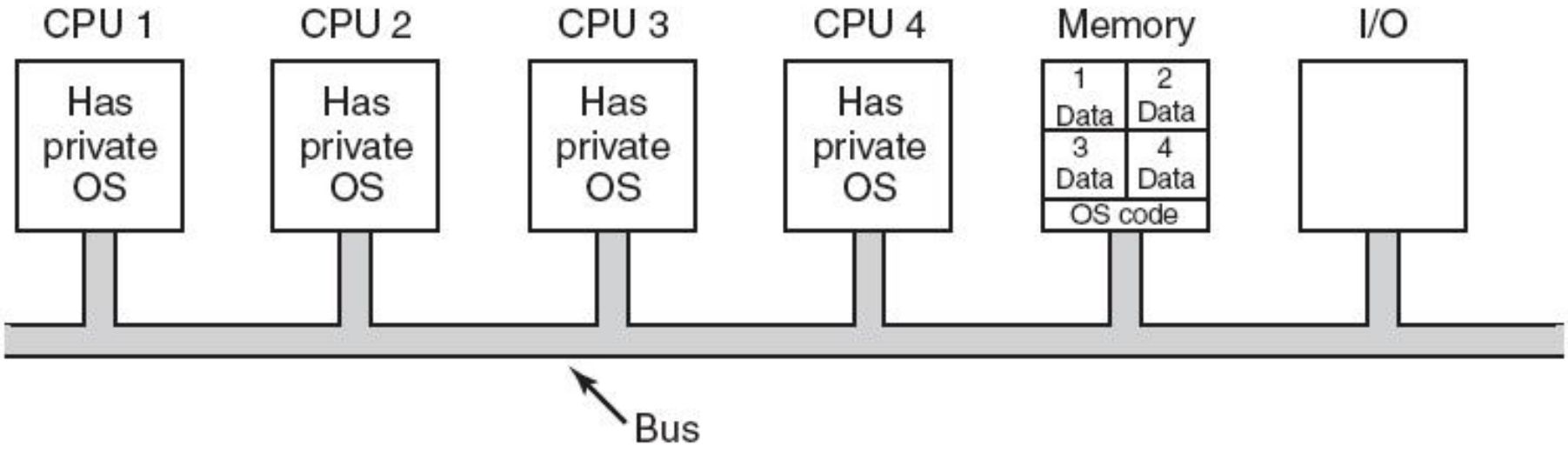


Figure 8-7. Partitioning multiprocessor memory among four CPUs, but sharing a single copy of the operating system code. The boxes marked Data are the operating system's private data for each CPU.

7.1.5 Master-Slave Multiprocessors

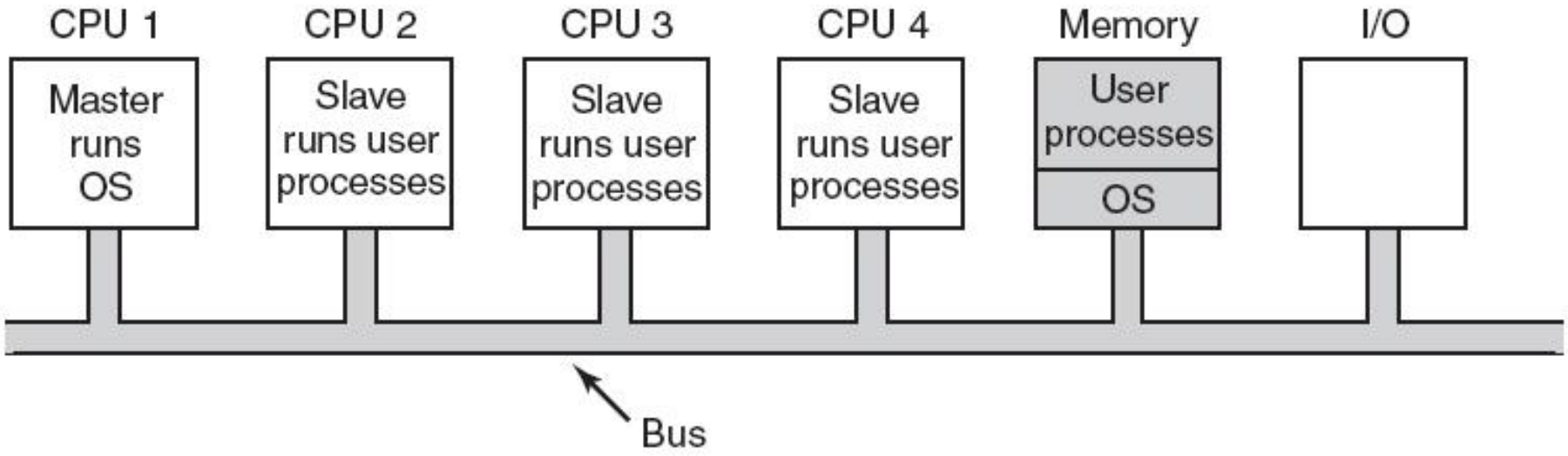


Figure 8-8. A master-slave multiprocessor model.

7.1.5 Symmetric Multiprocessors

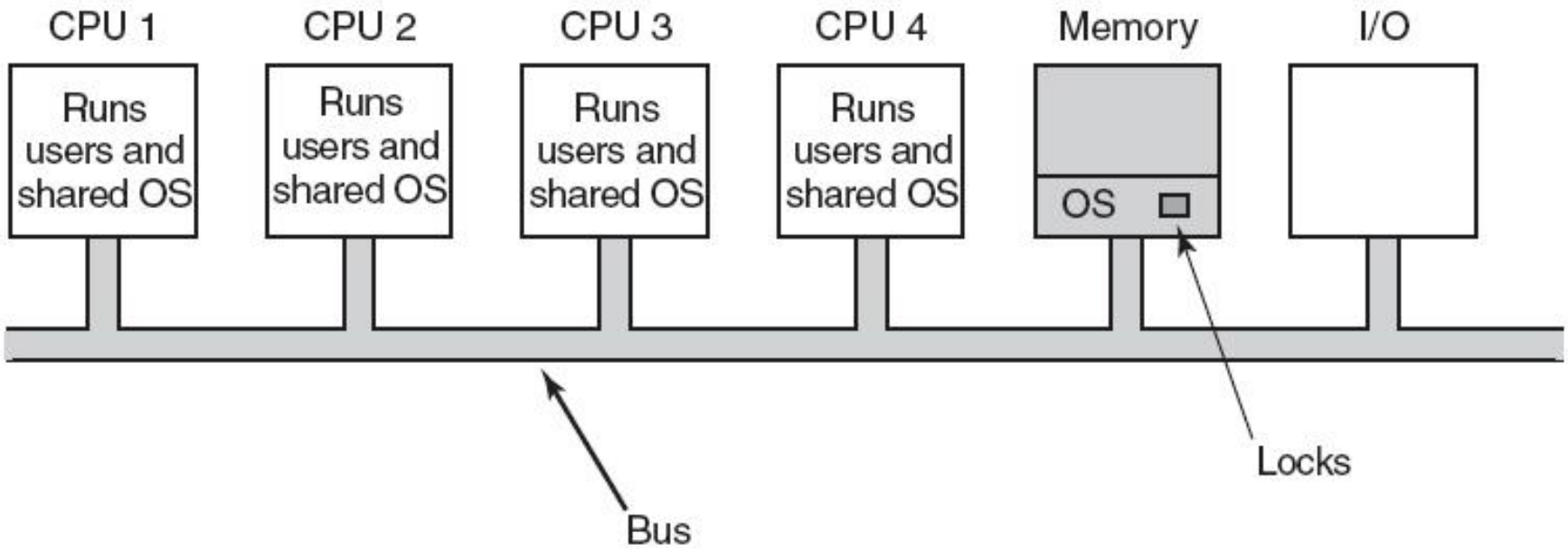


Figure 8-9. The SMP multiprocessor model.

7.1.5 Multiprocessor Operating System Types

Types	Advantages	disadvantages
Each CPU has its own operating system	(1)Memory , disks and other I/O devices can be shared flexibly. (2) Processes can efficiently communicate with one another.	(1) System call is caught and handled on its own CPU. (2) No sharing of processes (3) No sharing of pages (4) Inconsistent results because of using cache.
Master-slave multiprocessors	(1) Single data structure keeps track of ready processes. (2) Load balancing (3) No inconsistent results.	(1)Master will become a bottleneck. (2) can't for large multiprocessors.
Symmetric multiprocessors	(1)Eliminate the asymmetry. (2)Balance processes and memory dynamically. (3)Eliminate the master CPU bottleneck.	(1) Disaster may well result. (2) As bad as the master-slave model

7.1.6 Multiprocessor Synchronization (1)

- Who need
  - Concurrent processes in the same processor
  - Processes in different processors
- Synchronizations include
  - semaphare
  - Spinlocks , RCU lock, event count, centric process
- Three approaches for multiprocessor software
  - Each CPU has its own operating system
  - Master-slave multiprocessors
  - Symmetric multiprocessors

7.1.6 Multiprocessor Synchronization (2)

- synchronization mechanisms
  - centralized synchronization
    - disadvantages
  - Distributed synchronization
    - Five conditions
    - Hard to realized
  - Central process
    - Spin lock
    - read-copy modified lock

7.1.6 Multiprocessor Synchronization (3)

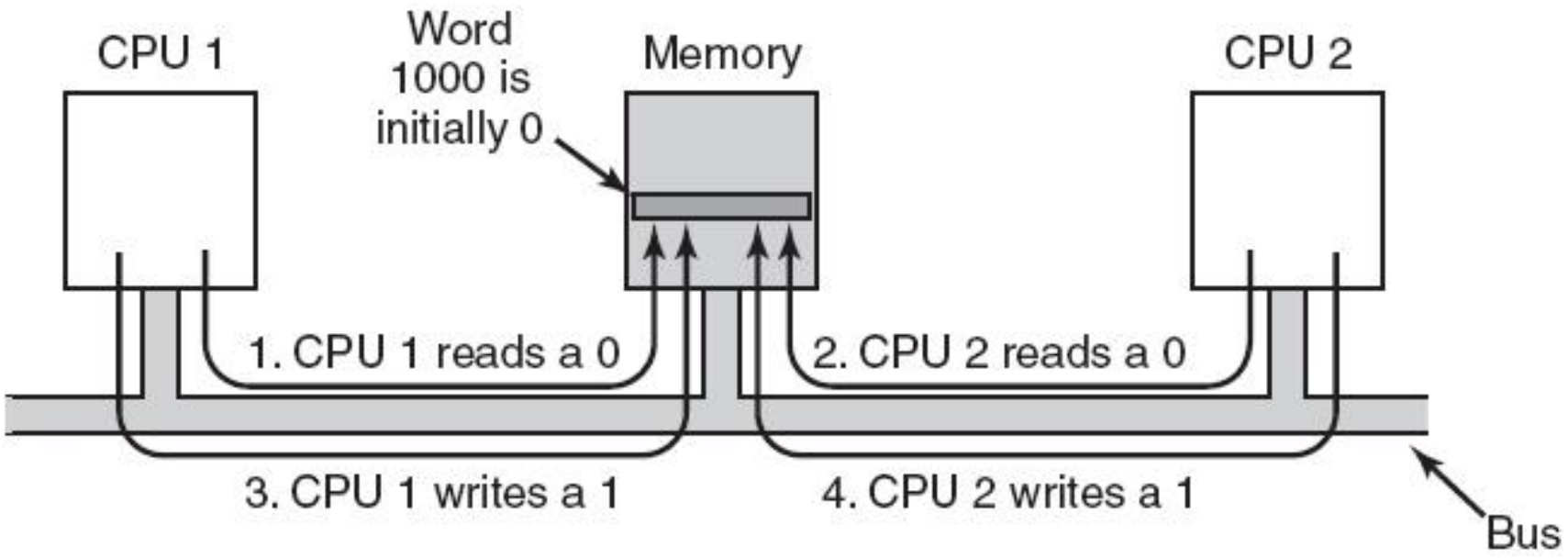


Figure 8-10. The TSL instruction can fail if the bus cannot be locked. These four steps show a sequence of events where the failure is demonstrated.

7.1.6 Multiprocessor Synchronization (4)

- Binary exponential backoff algorithm
  - Idea: delay TSL instruction's execution time
  - Advantage: reduce bus traffic
  - Disadvantage: waste idle lock
- Private lock variables

7.1.6 Multiprocessor Synchronization (5)

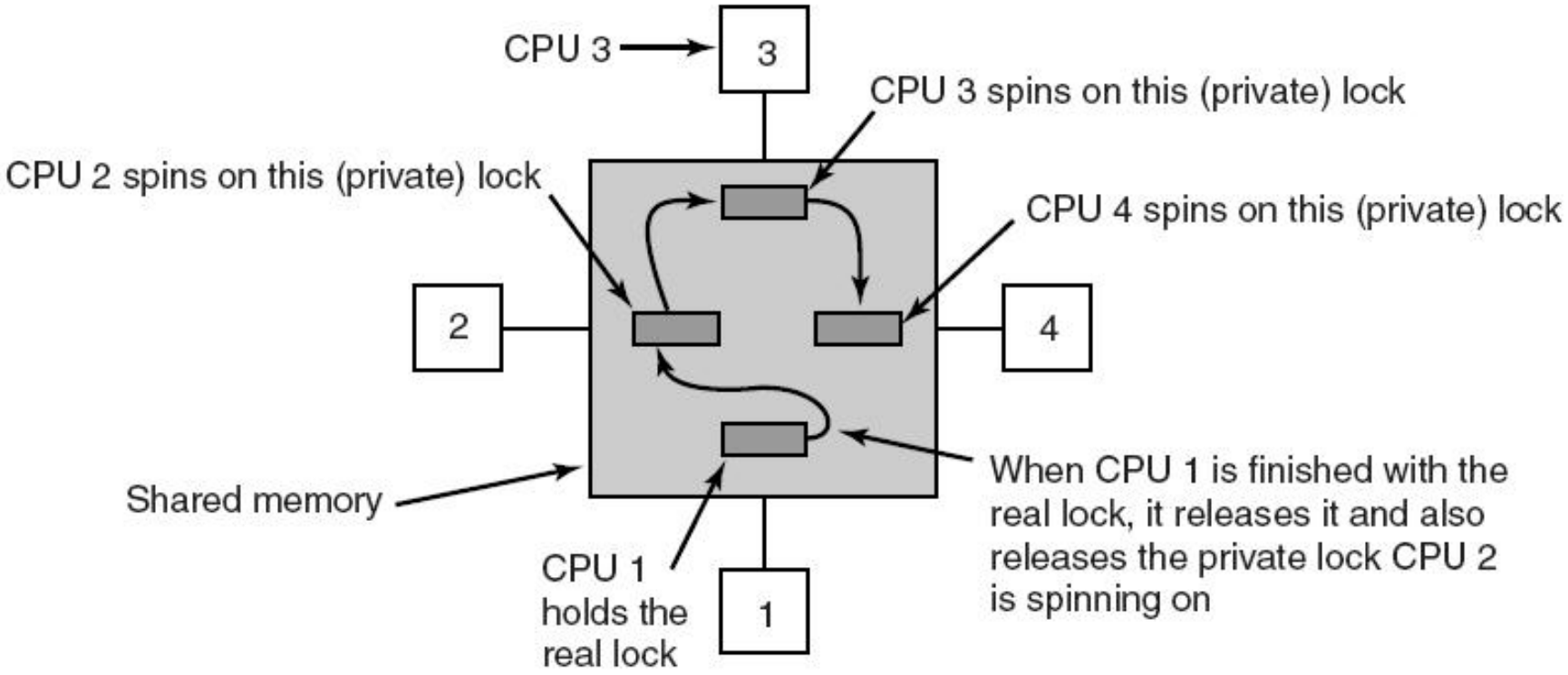


Figure 8-11. Use of multiple locks to avoid cache thrashing.

7.1.6 Multiprocessor Synchronization (6)

- Spinning versus Switching
  - Scene: if some thread on a CPU needs to access the file system buffer cache and it is currently locked, the CPU can decide to switch to a different thread instead of waiting.
  - Using hindsight algorithm to get the better selection

7.1.7 Multiprocessor Scheduling

- Performance Parameters of scheduling
  - task flow time
  - Scheduling flow time
  - Average flow time
  - CPU utilization
  - speedup ratio
  - throughput

1. Timesharing

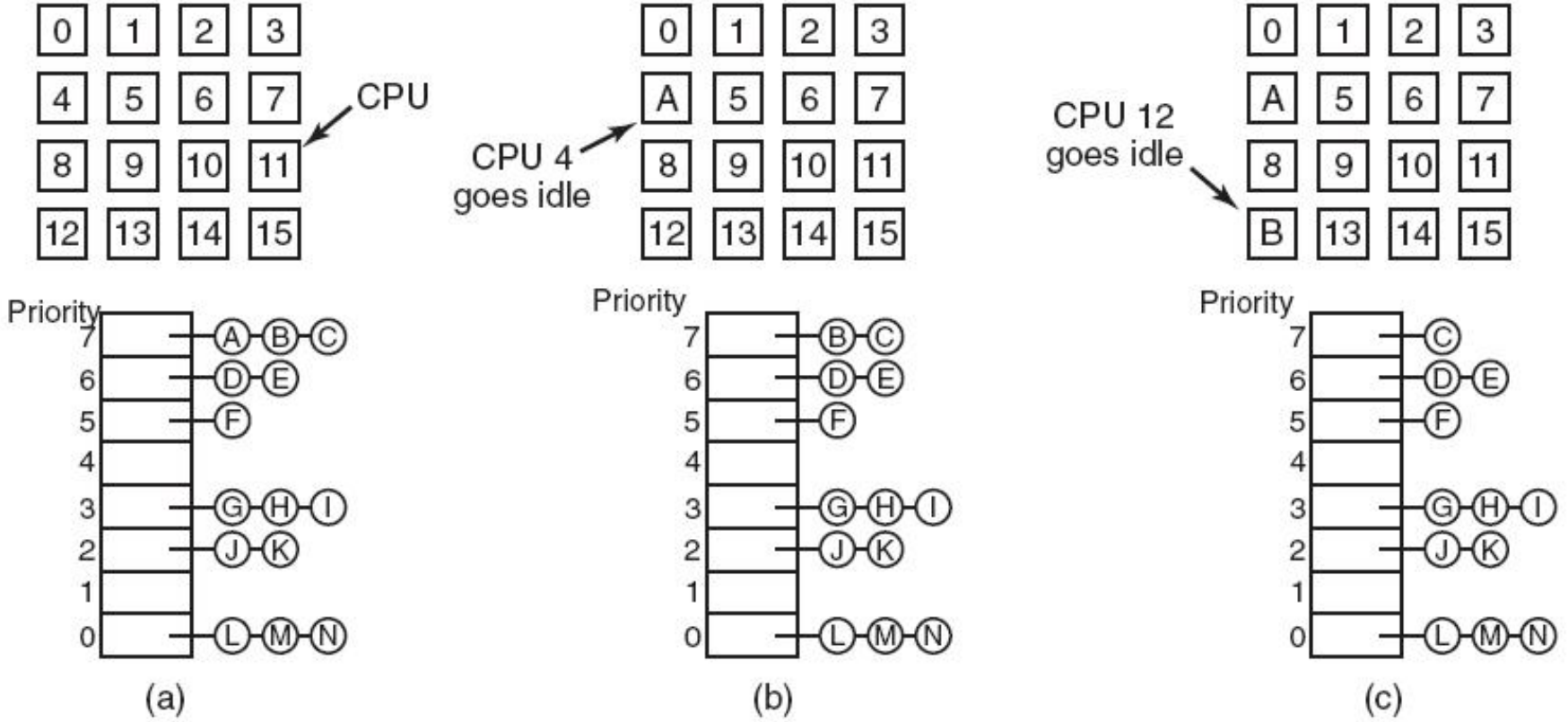


Figure 8-12. Using a single data structure for scheduling a multiprocessor.

1. Timesharing

- Problems
  - the potential contention for the scheduling data structure as the number of CPUs grows
  - the usual overhead in doing a context switch when a thread blocks for I/O.
- Solutions
  - Smart scheduling
  - Affinity scheduling---two level algorithm
    - load balancing
    - advantage of cache affinity
    - Contention be minimized

2. Space sharing

- Scheduling multiple threads at the same time across multiple CPUs is called space sharing.
  - One CPU for a thread
  - Scheduling mechanism: SJF, FCFS
- Advantages
  - elimination of multiprogramming
  - eliminates the context switching overhead
- Disadvantages
  - time waste

2. Space Sharing

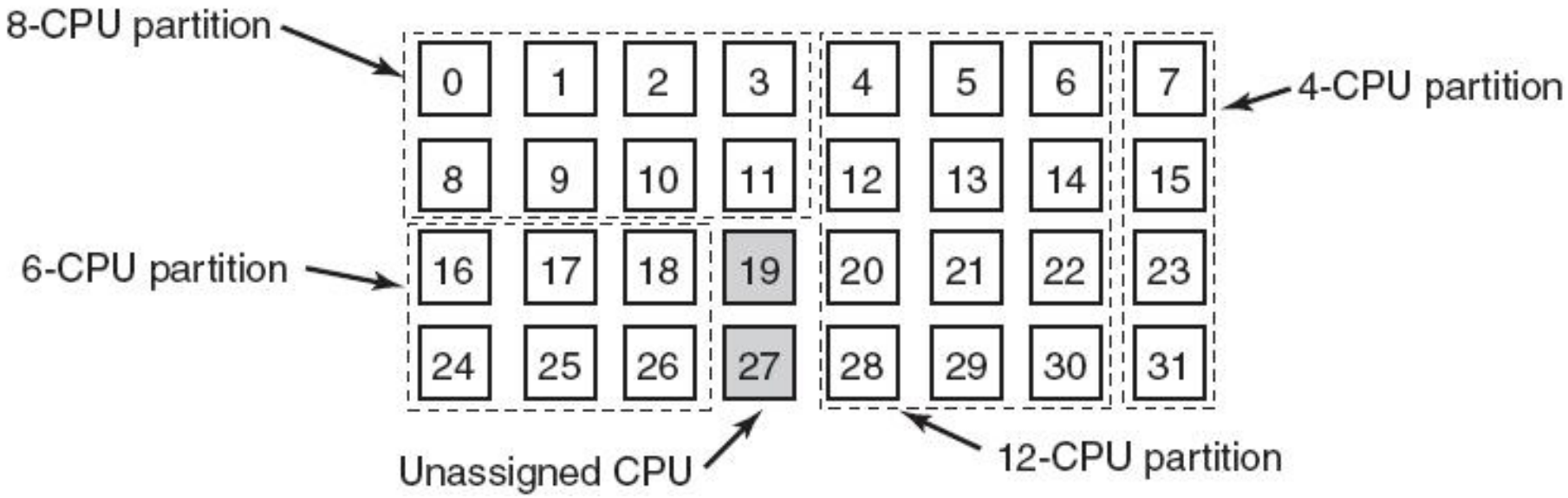


Figure 8-13. A set of 32 CPUs split into four partitions, with two CPUs available.

3. Gang Scheduling (1)

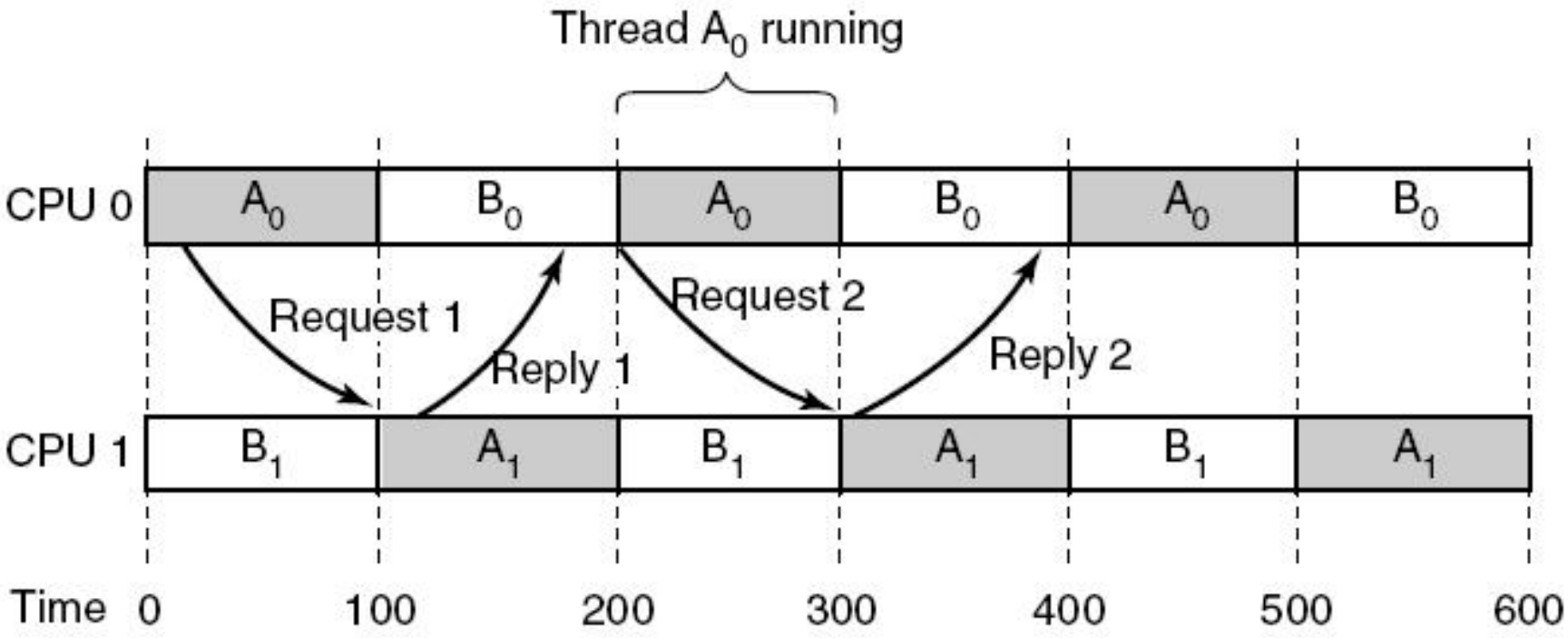


Figure 8-14. Communication between two threads belonging to thread A that are running out of phase.

3. Gang Scheduling (2)

The three parts of gang scheduling:

- Groups of related threads are scheduled as a unit, a gang.
- All members of a gang run simultaneously, on different timeshared CPUs.
- All gang members start and end their time slices together.

3. Gang Scheduling (3)

Time slot \ CPU	0	1	2	3	4	5
0	A_0	A_1	A_2	A_3	A_4	A_5
1	B_0	B_1	B_2	C_0	C_1	C_2
2	D_0	D_1	D_2	D_3	D_4	E_0
3	E_1	E_2	E_3	E_4	E_5	E_6
4	A_0	A_1	A_2	A_3	A_4	A_5
5	B_0	B_1	B_2	C_0	C_1	C_2
6	D_0	D_1	D_2	D_3	D_4	E_0
7	E_1	E_2	E_3	E_4	E_5	E_6

Figure 8-15. Gang scheduling.

7.2 MultiComputers

- Two types of MultiComputers
  - Cluster computers
  - COWS: Clusters of Workstations
- Features of Multicomputers
  - Tighly couples CPU
  - NOT share memory
- Component of multicoputers
  - Stripped-down PC WITH the addition of a high-performance network interface card.

7.2.1 Multicomputer Hardware
1. Interconnection Technology (1)

- topology
  - Star
  - Ring
  - Grid or mesh
  - Double torus
  - Cube
  - 4D hypercube

1. Interconnection Technology (2)

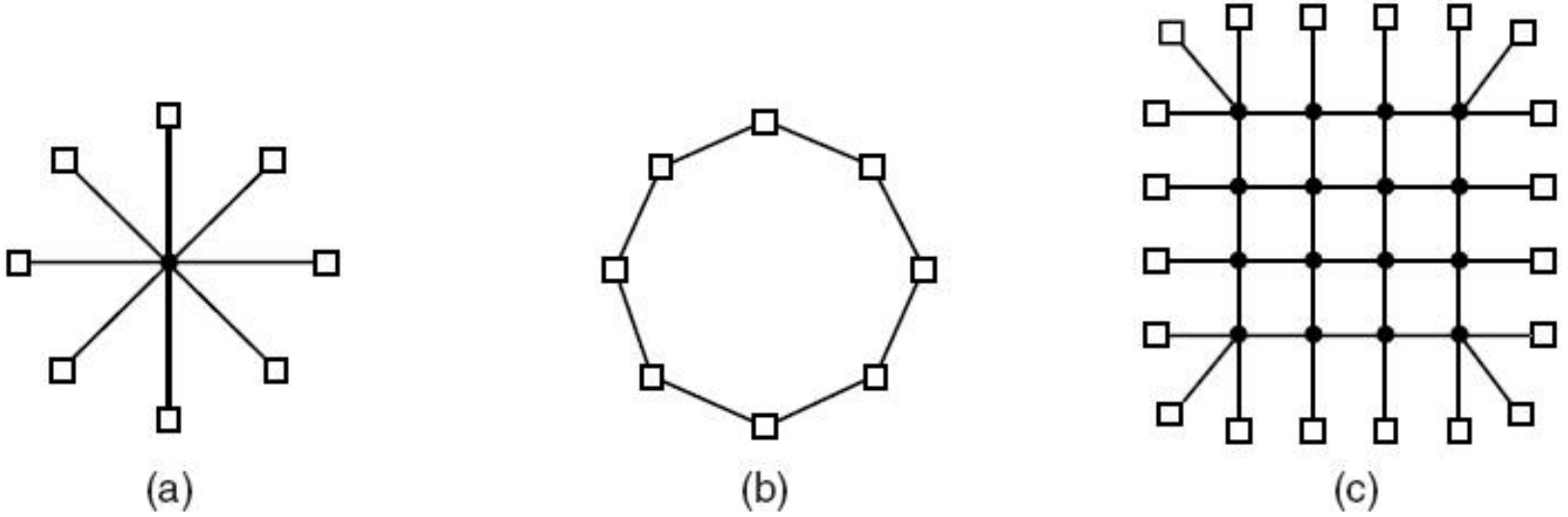


Figure 8-16. Various interconnect topologies.
(a) A single switch. (b) A ring. (c) A grid.

1. Interconnection Technology (3)

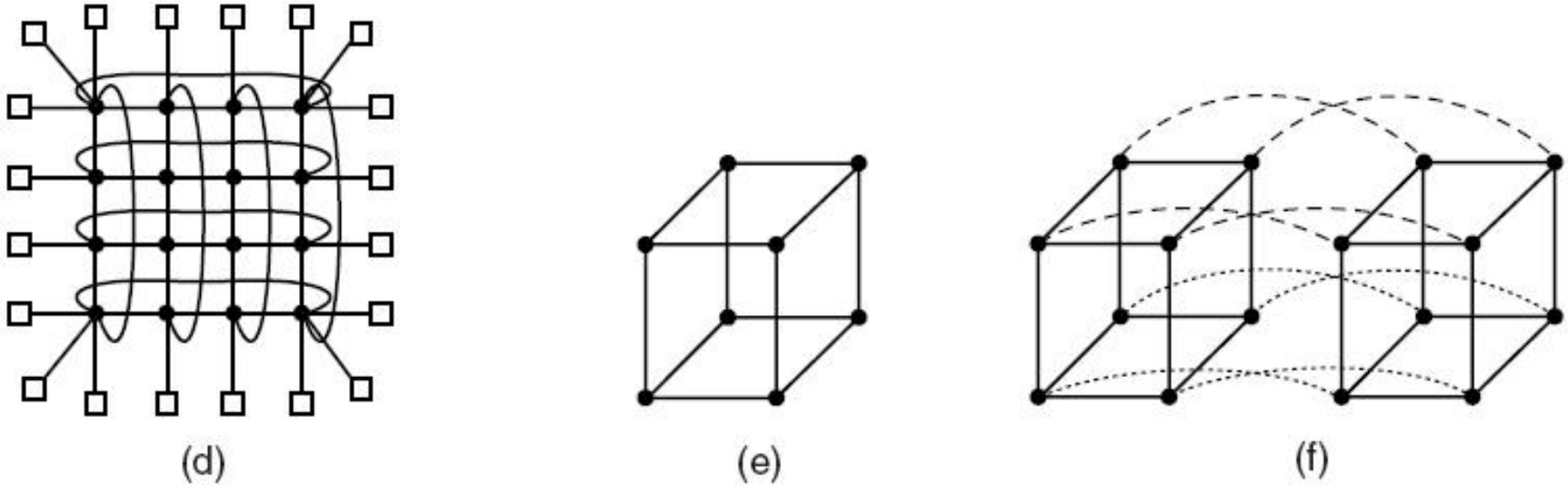


Figure 8-16. Various interconnect topologies.
(d) A double torus. (e) A cube. (f) A 4D hypercube.

1. Interconnection Technology (4)

- switching schemes
  - Store-and-forward packet switching
    - Advantages: flexible and efficient
    - Disadvantage: increasing latency (delay)
  - circuit switching
    - the first switch first establishing a path through all the switches to the destination switch.
    - no intermediate buffering
    - requires a setup phase and path-torn down phase
    - Another variation: wormhole routing

1. Interconnection Technology (5)

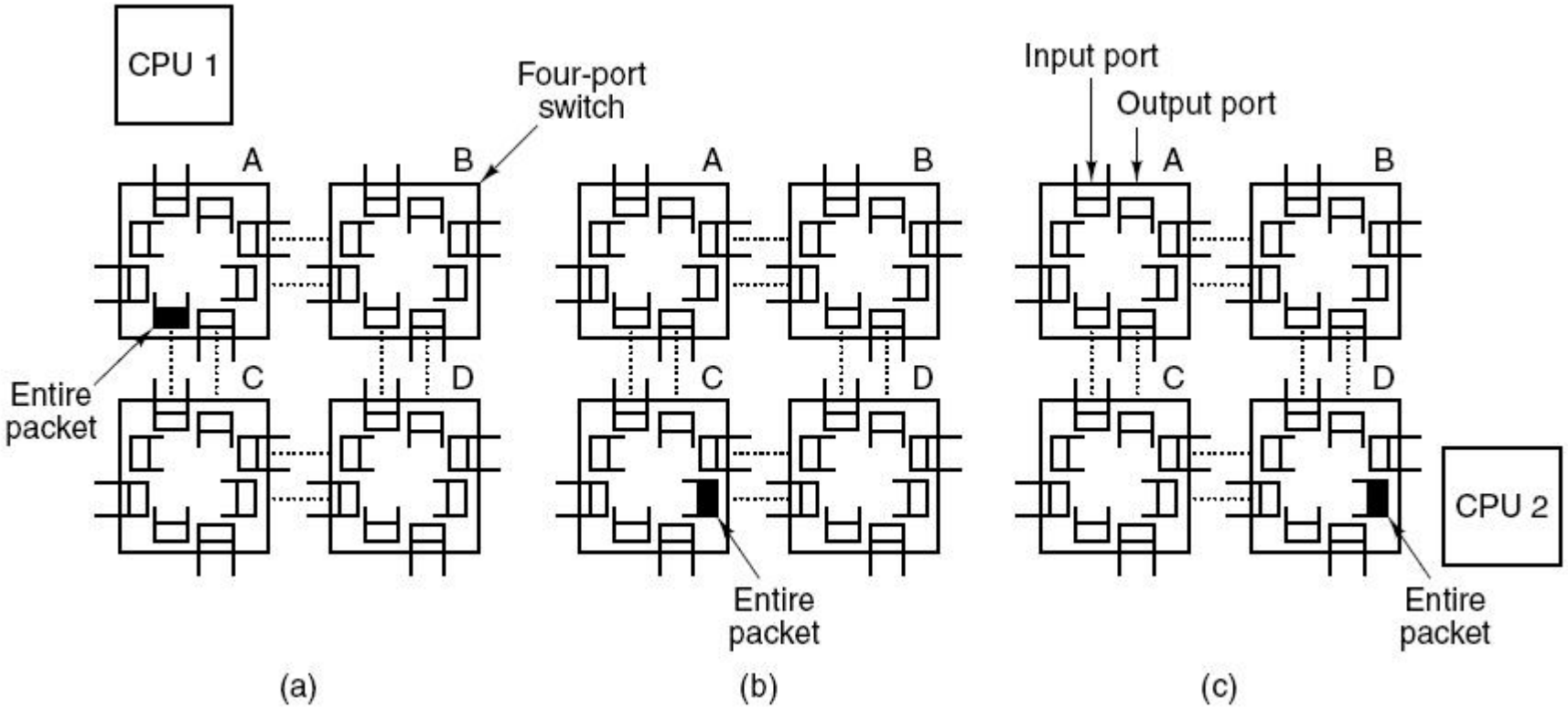


Figure 8-17. Store-and-forward packet switching.

2. Network Interfaces

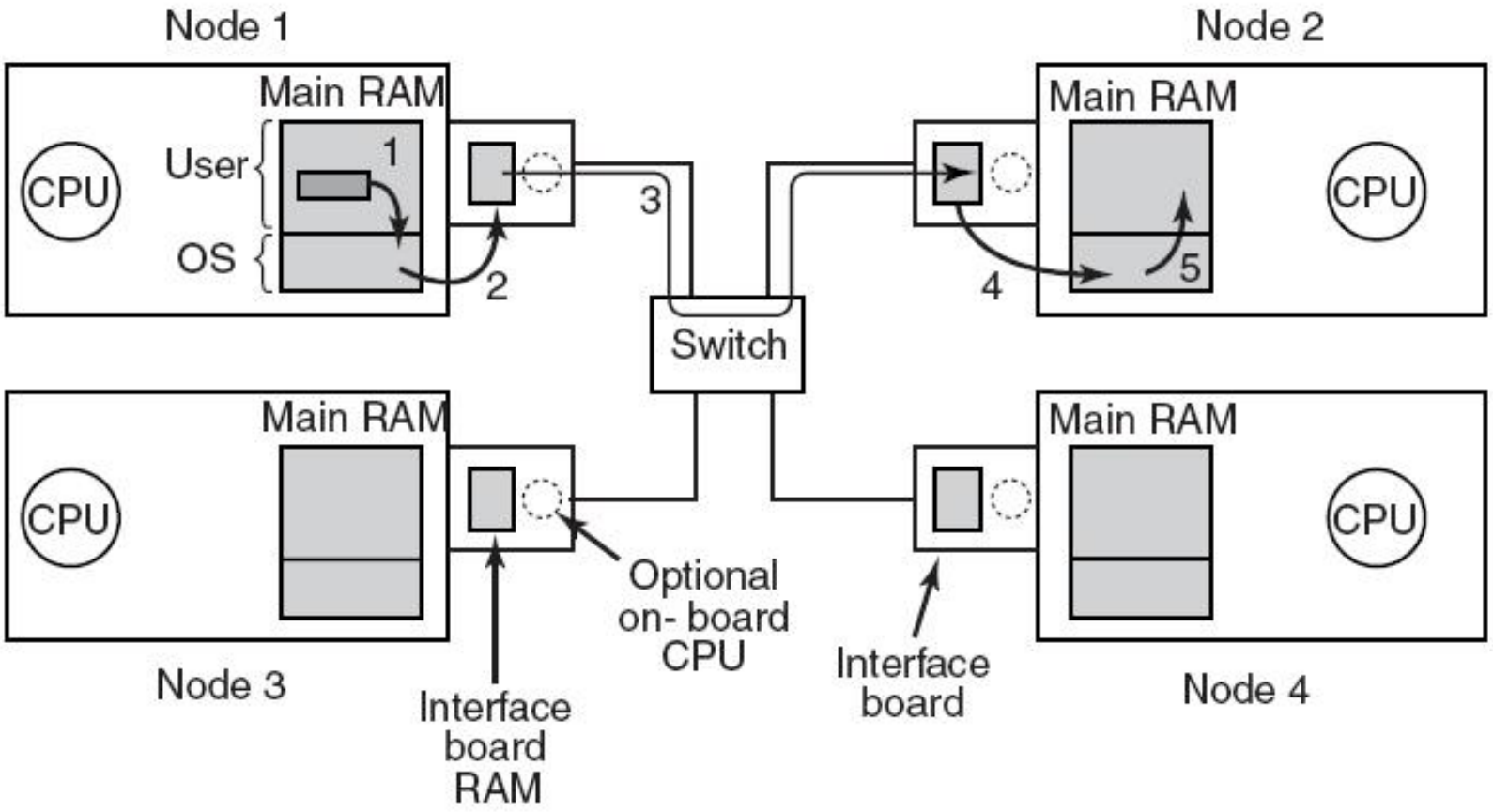


Figure 8-18. Position of the network interface boards in a multicomputer.

7.2.2 Low-Level Communication Software

- Solve three problems
  - what if several processes are running on the node and need network access to send packets? Which one gets the interface board in its address space?
  - kernel may need access to the interconnection network itself, how to differentiate from user process?
  - how to get packets onto the interface board?

7.2.3 User-Level Communication Software

- communication services provided can be reduced to two (library) calls
  - Sending messages
    - send (dest, &mptr)
  - Receiving messages
    - Receive (addr, &mptr)
- Blocking versus Nonblocking calls
  - Blocking calls----synchronous calls
  - Nonblocking calls----asynchronous calls

Blocking versus Nonblocking Calls (1)

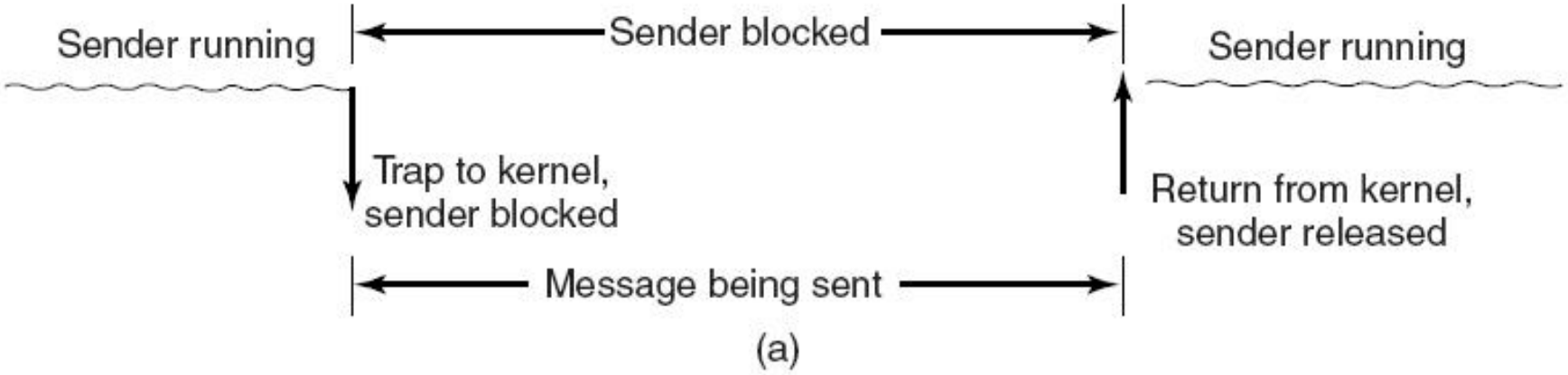


Figure 8-19. (a) A blocking send call.

Blocking versus Nonblocking Calls (2)

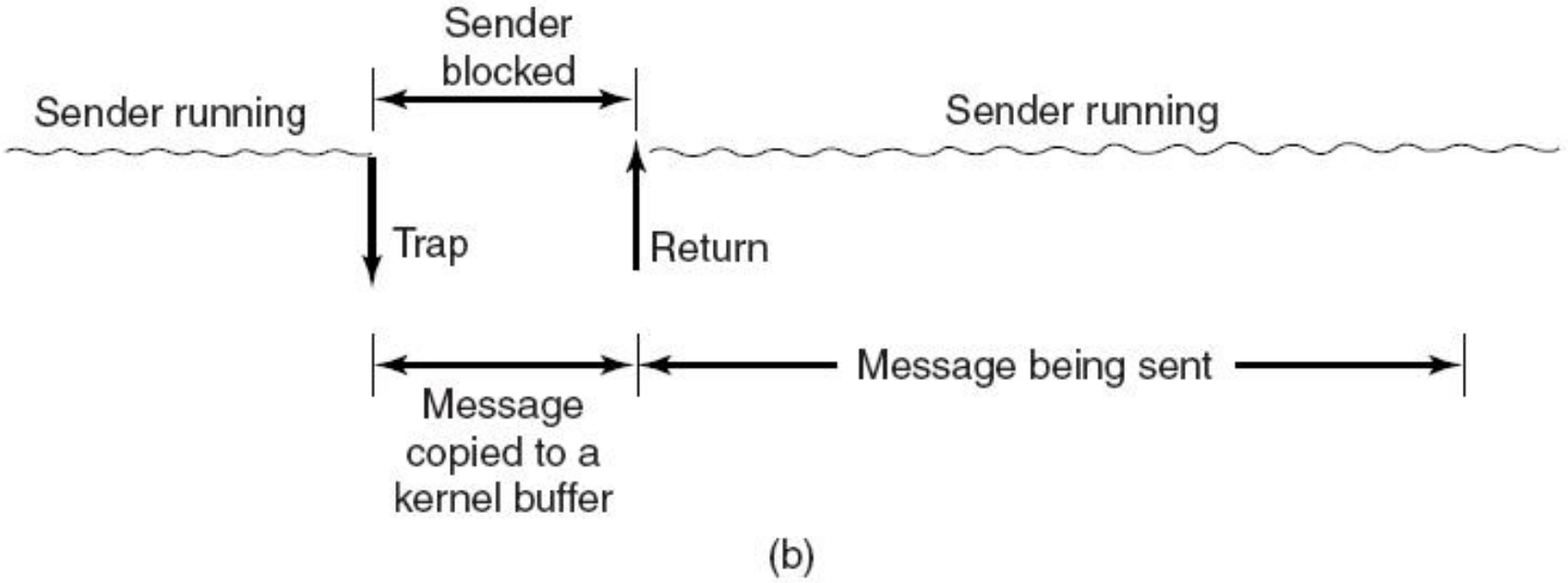


Figure 8-19. (b) A nonblocking send call.

Blocking versus Nonblocking Calls (3)

Choices on the sending side:

- Blocking send (CPU idle during message transmission).
- Nonblocking send with copy (CPU time wasted for the extra copy).
- Nonblocking send with interrupt (makes programming difficult).
- Copy on write (extra copy probably needed eventually).

7.2.4 Remote Procedure Call

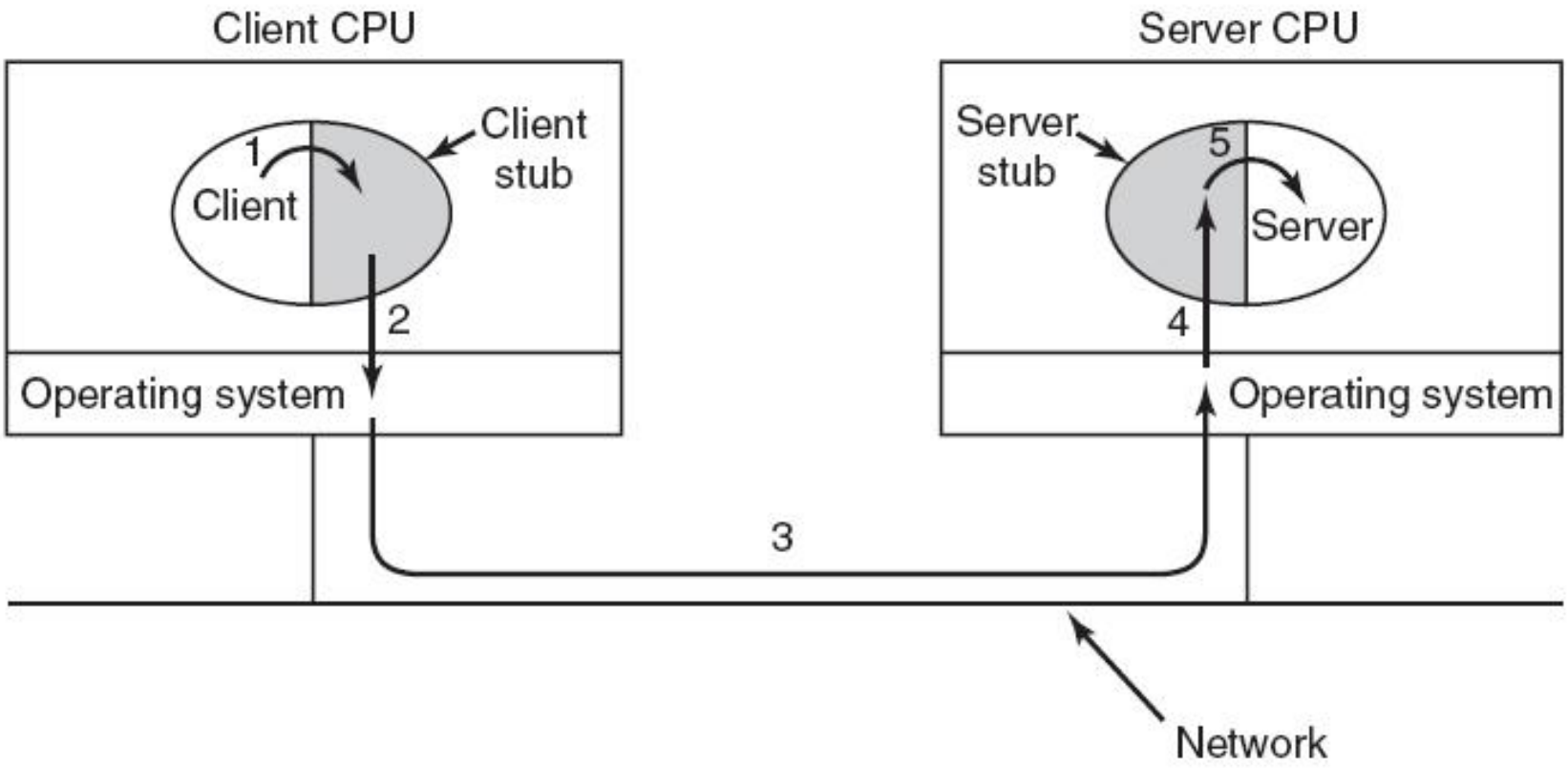


Figure 8-20. Steps in making a remote procedure call. The stubs are shaded gray.

Remote Procedure Call

- Implementation Issues
  - The use of pointer parameters: the client and server are in different address spaces.
    - sloving: method: integer k.
  - weakly typed languages is perfectly legal to write a procedure that computes the inner product of two vectors (arrays), without specifying how large either one is.
  - it is not always possible to deduce the types of the parameters
  - the use of global variables

7.2.5 Distributed Shared Memory (1)

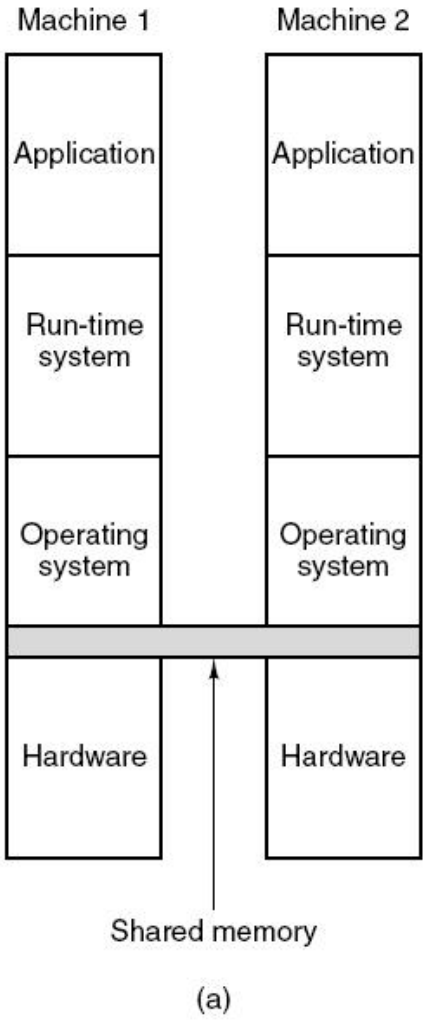


Figure 8-21. Various layers where shared memory can be implemented.
(a) The hardware.

7.2.5 Distributed Shared Memory (2)

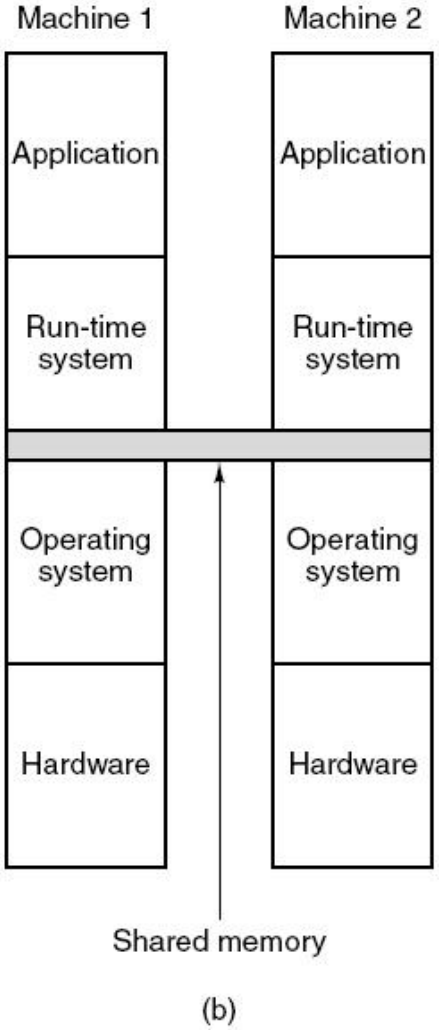


Figure 8-21. Various layers where shared memory can be implemented. (b) The operating system

7.2.5 Distributed Shared Memory (3)

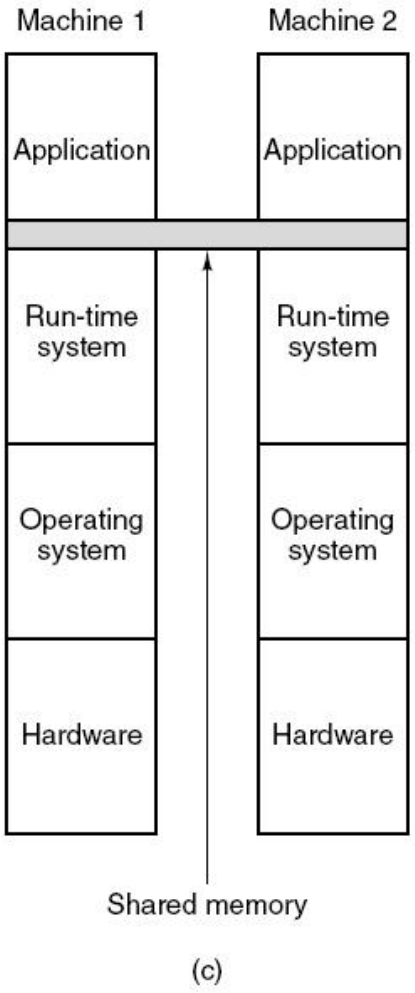


Figure 8-21. Various layers where shared memory can be implemented.
(c) User-level software

7.2.5 Distributed Shared Memory (4)

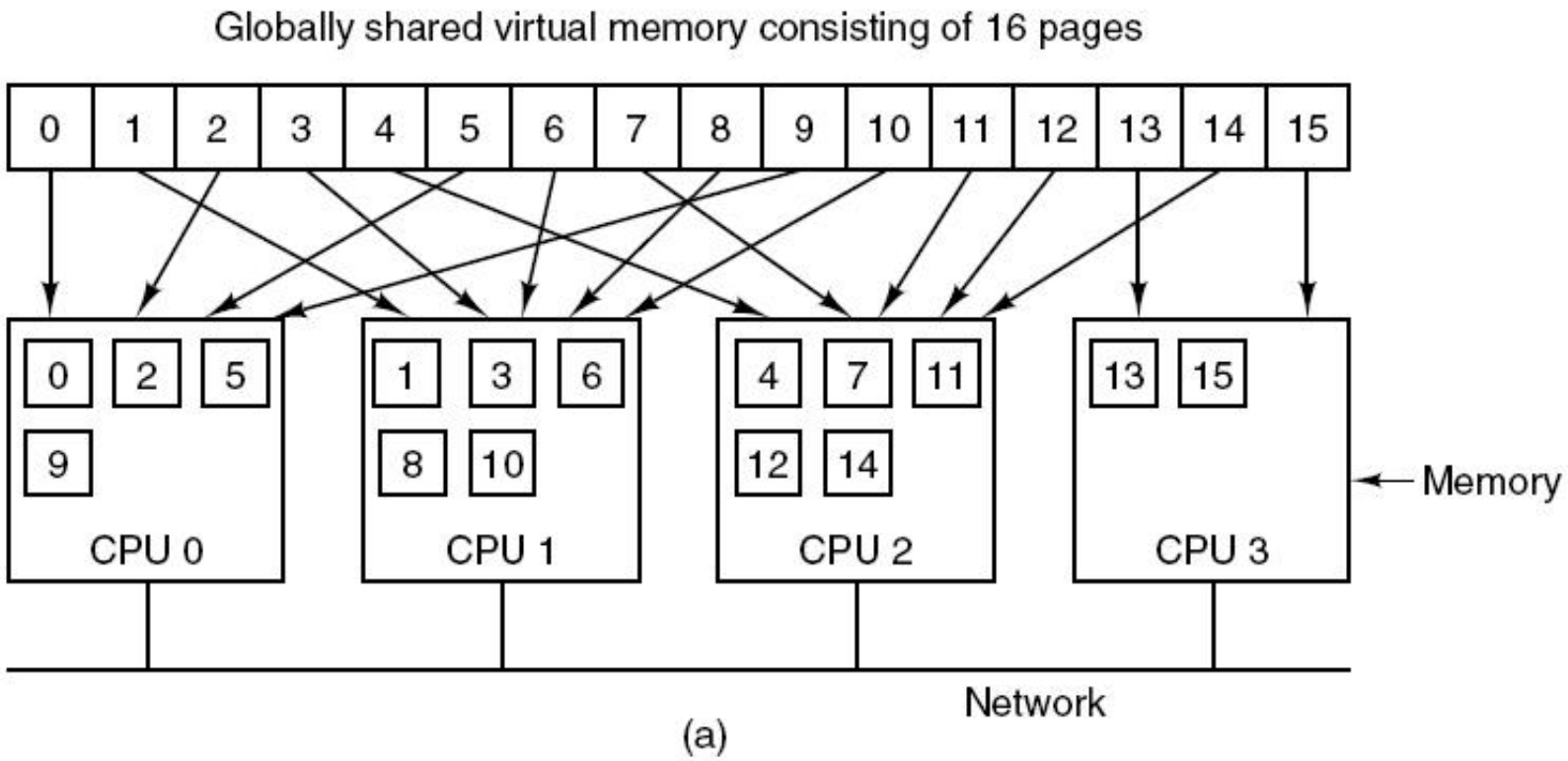


Figure 8-22. (a) Pages of the address space distributed among four machines.

7.2.5 Distributed Shared Memory (5)

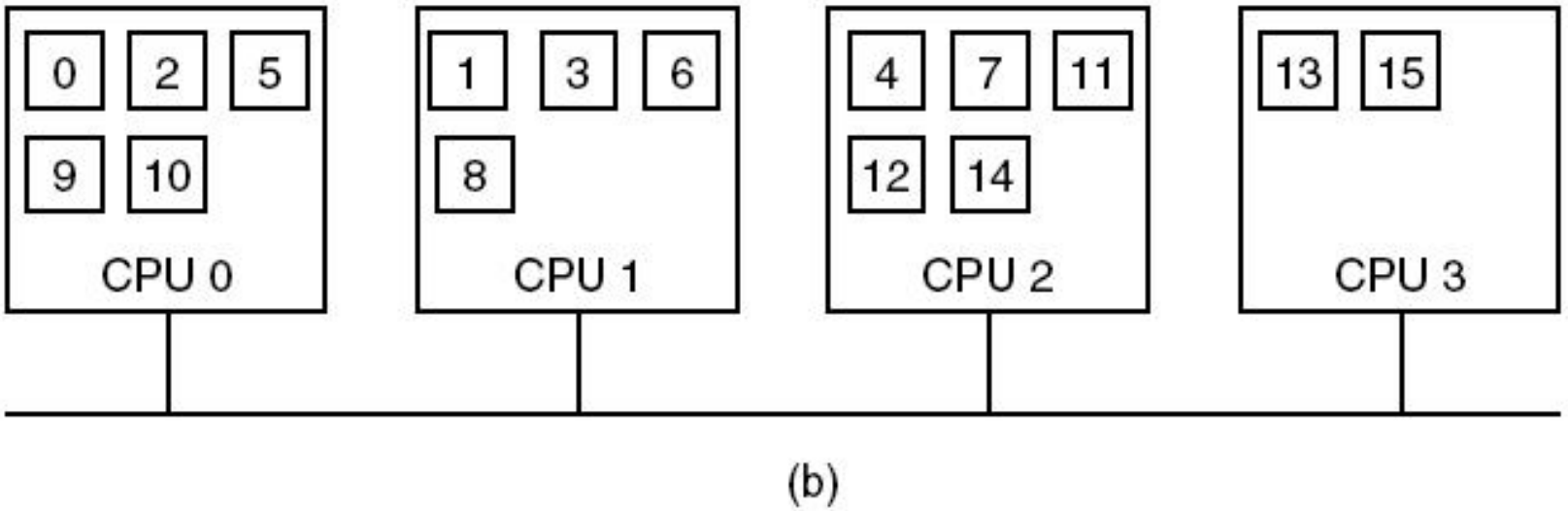


Figure 8-22. (b) Situation after CPU 1 references page 10 and the page is moved there.

7.2.5 Distributed Shared Memory (6)

1. copy

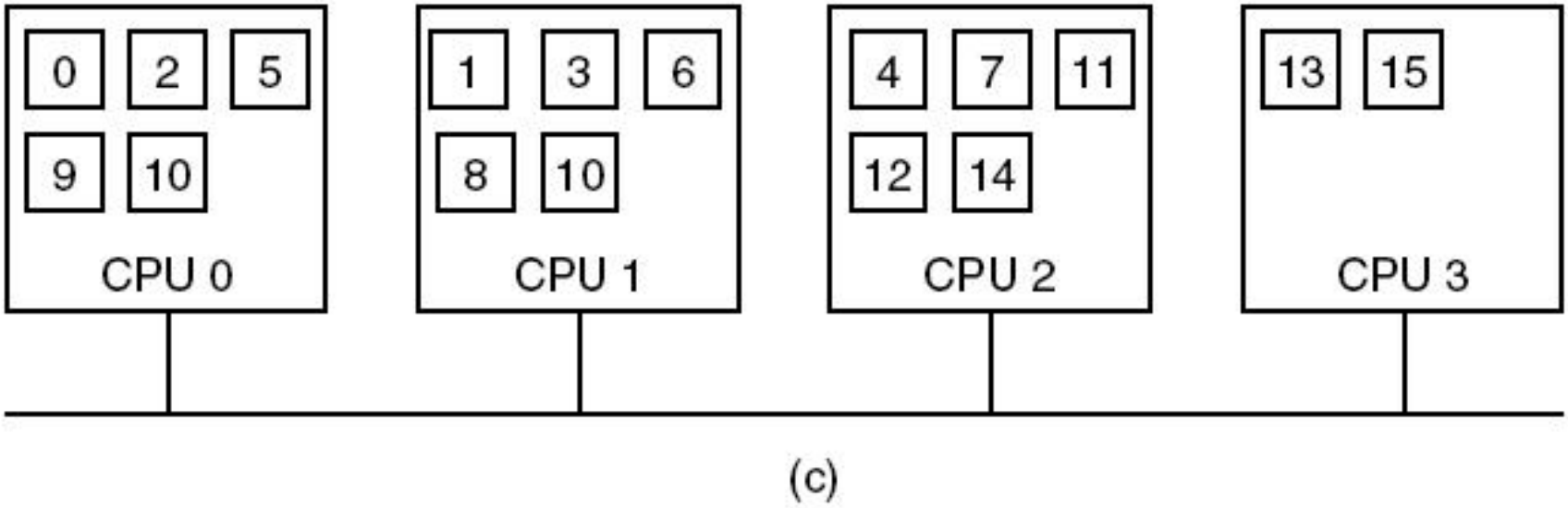


Figure 8-22. (c) Situation if page 10 is read only and replication is used.

2. False Sharing

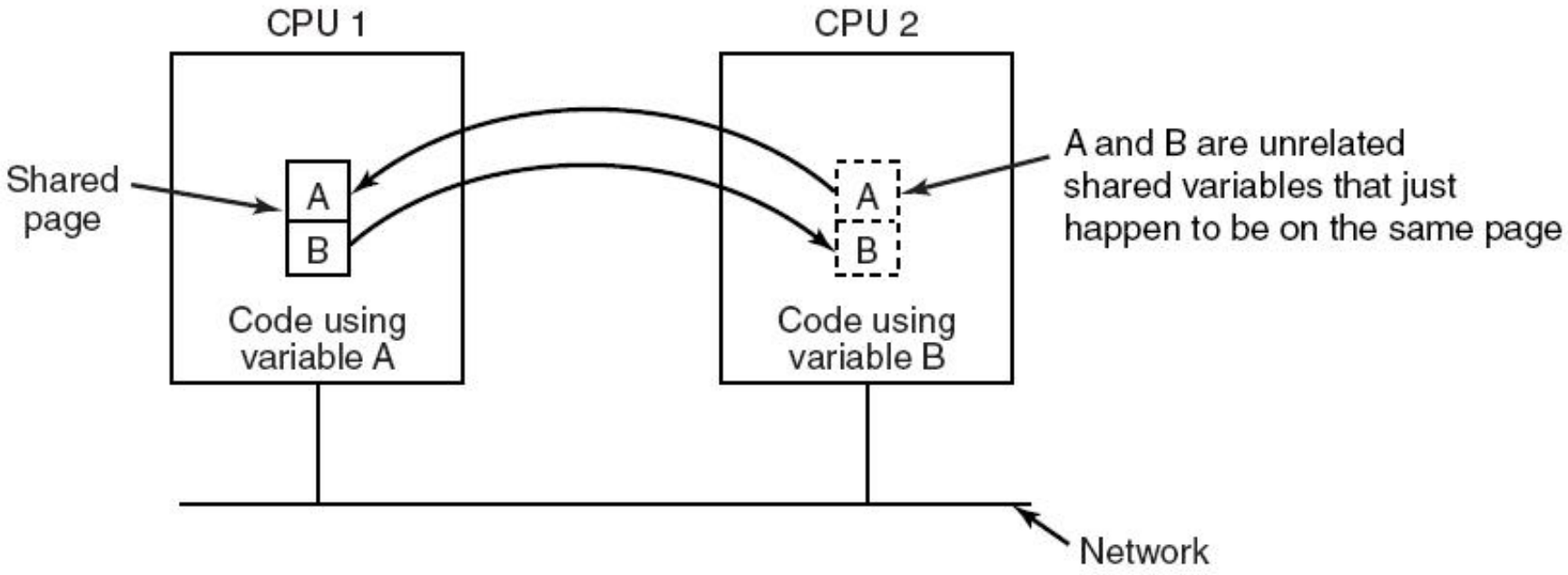


Figure 8-23. False sharing of a page containing two unrelated variables.

7.2.6 Multicomputer Scheduling

- Must coordinate the start of the time slots.
- processor allocation algorithms is worth to looking---how processes be assigned to nodes in an effective way
- goals
  - minimizing wasted CPU cycles due to lack of local work
  - Minimizing total communication bandwidth
  - ensuring fairness to users and processes

7.2.7 A Graph-Theoretic Deterministic Algorithm

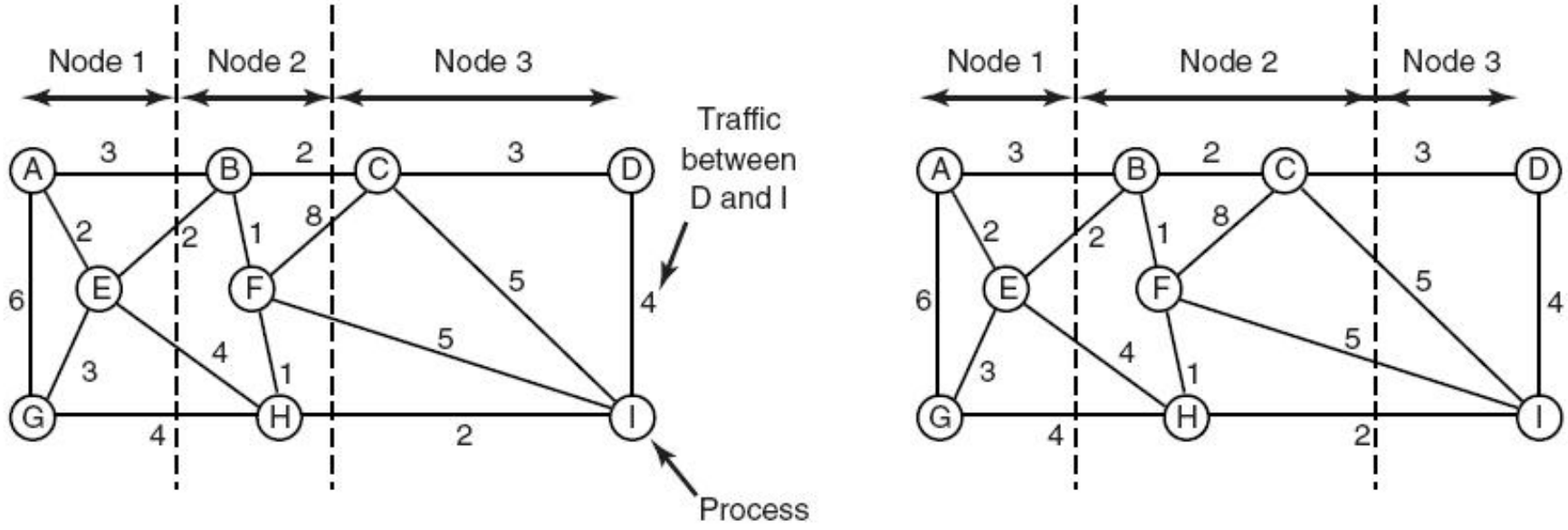


Figure 8-24. Two ways of allocating nine processes to three nodes.

7.2.7 A Sender-Initiated Distributed Heuristic Algorithm

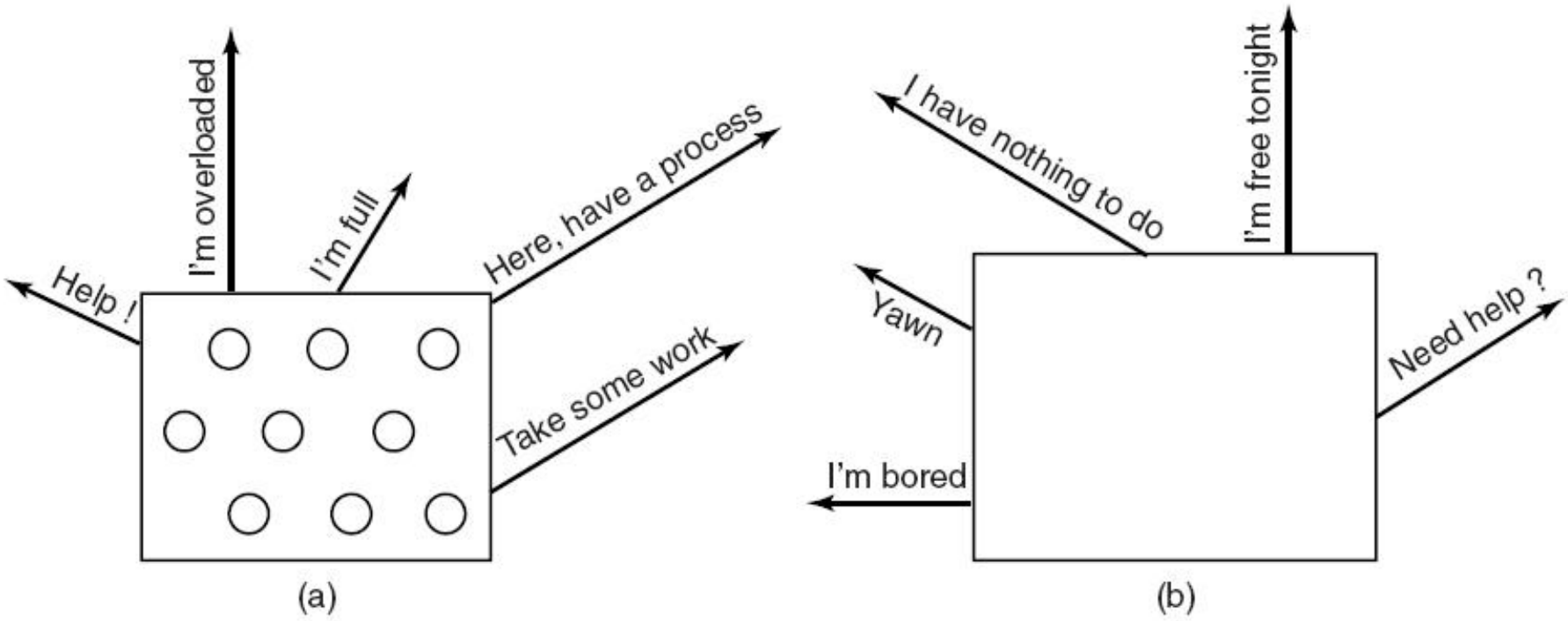
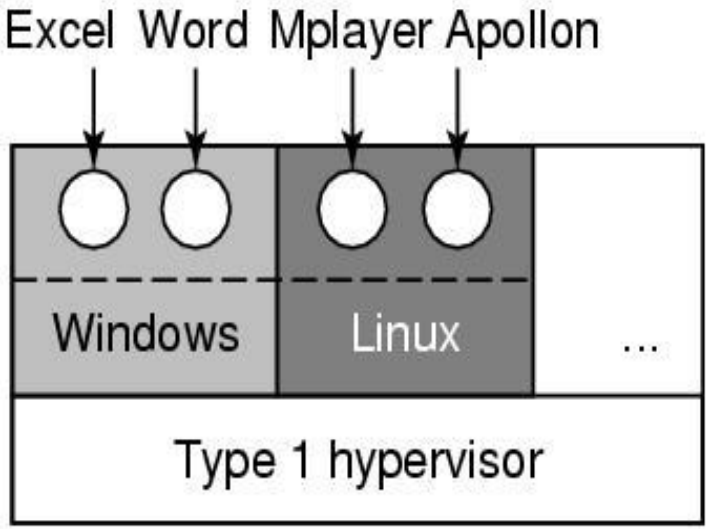


Figure 8-25. (a) An overloaded node looking for a lightly loaded node to hand off processes to. (b) An empty node looking for work to do.

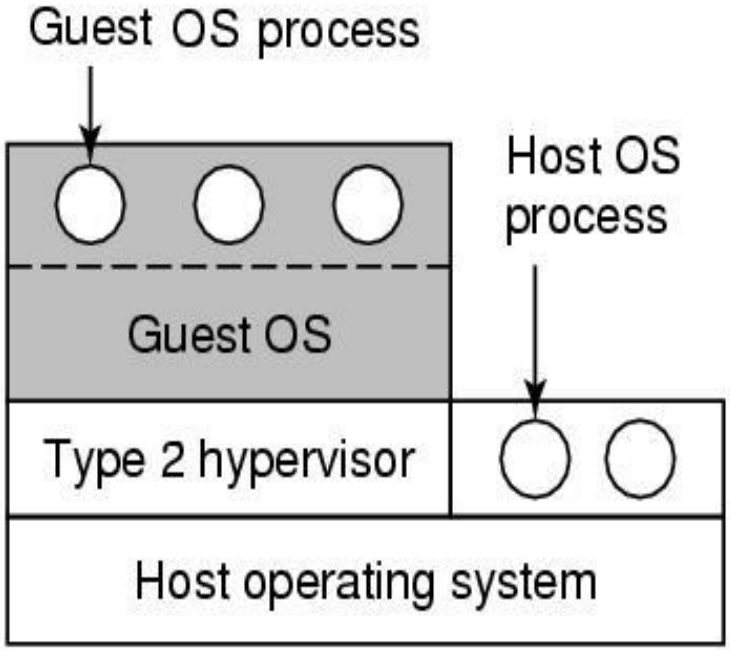
7.3 Virtualization

- Advantages
  - failure in one virtual machine does not automatically bring down any others
  - much lower cost :fewer physical machines saves money on hardware and electricity and takes up less office space
  - easier maintainability & strong isolation
  - checkpointing and migrating virtual machines is much easier than migrating processes running on a normal operating system.
  - to run legacy applications on operating no longer supported or which do not work on current hardware
  - software development

7.3 Virtualization



(a)



(b)

7.3 Virtualization

- Types
  - (a) type 1 hypervisor : virtual machine monitor
  - (b) type 2 hypervisor
- Requirements for Virtualization
  - sensitive instructions
  - privileged instructions
  - a machine is virtualizable only if the sensitive instructions are a subset of the privileged instructions

7.3 Type 1 Hypervisors

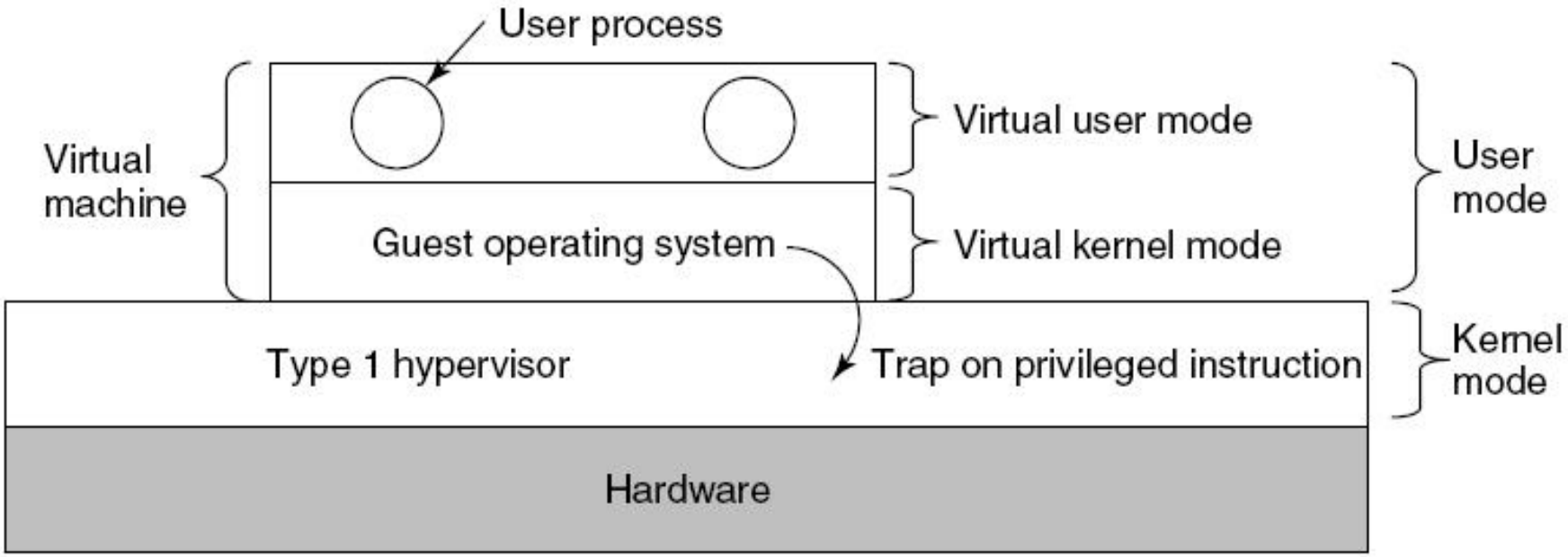


Figure 8-26. When the operating system in a virtual machine executes a kernel-only instruction, it traps to the hypervisor if virtualization technology is present.

7.3 Virtualization

- Paravirtualization
  - modify the source code of the guest operating system so that instead of executing sensitive instructions at all, it makes hypervisor calls
  - difference between true virtualization and paravirtualization is illustrated as follows.
  - some problems must be solved in paravirtualization
    - If the sensitive instructions are replaced with calls to the hypervisor, how can the operating system run on the native hardware?
    - what if there are multiple hypervisors available in the marketplace all with somewhat different hypervisor APIs
    - How can the kernel be modified to run on all of them?

7.3 Paravirtualization (1)

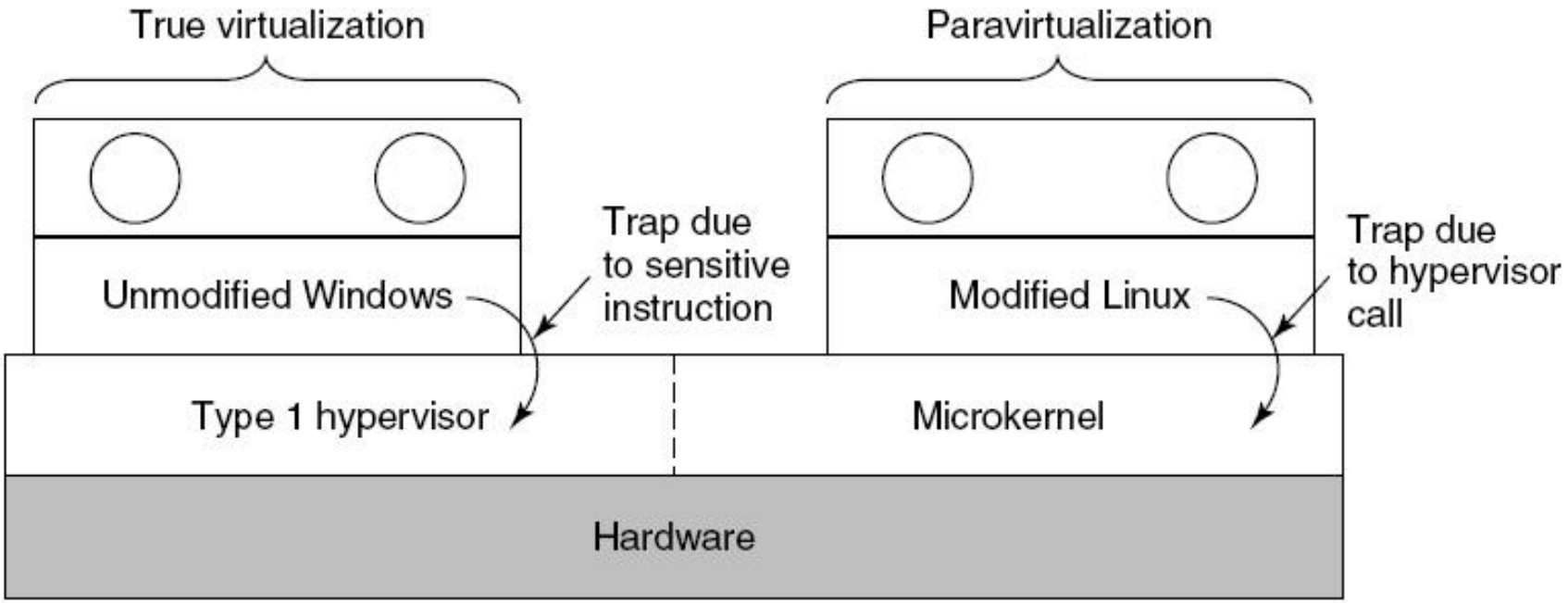


Figure 8-27. A hypervisor supporting both true virtualization and para virtualization.

7.3 Paravirtualization (2)

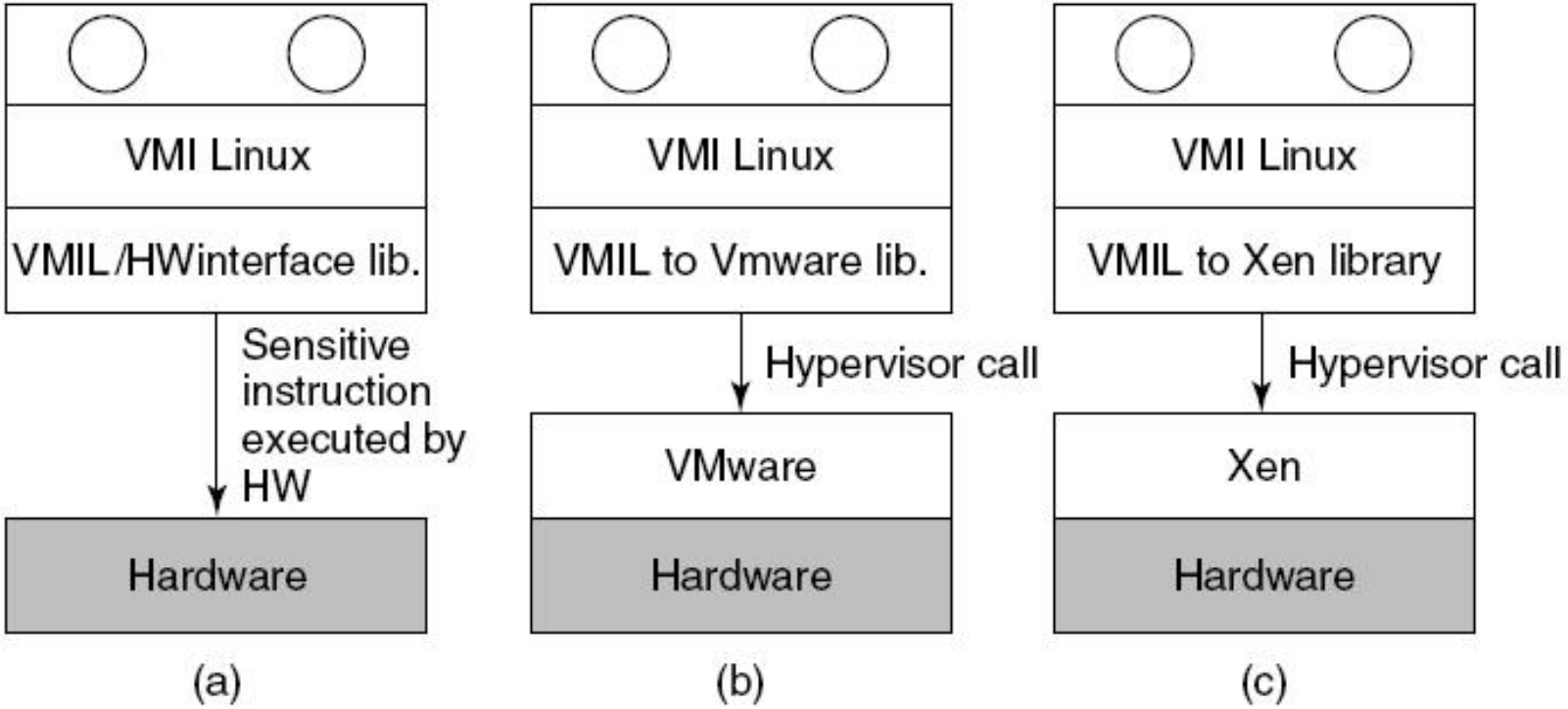


Figure 8-28. VMI Linux running on (a) the bare hardware (b) VMware (c) Xen.

7.3 Virtualization

- Memory Virtualization
  - Shadow page table
  - Virtual Technology---two-level mapping in the hardware
- I/O Virtualization
  - Each guest OS thinks it owns an entire disk partition
  - the use of DMA---absolute memory addresses---I/O MMU
  - standard operating system and reflect all I/O calls---Xen
- Virtual appliance
- Virtual Machines on Multicore CPUs

7.4 Distributed Systems (1)

Item	Multiprocessor	Multicomputer	Distributed System
Node configuration	CPU	CPU, RAM, net interface	Complete computer
Node peripherals	All shared	Shared exc. maybe disk	Full set per node
Location	Same rack	Same room	Possibly worldwide
Internode communication	Shared RAM	Dedicated interconnect	Traditional network
Operating systems	One, shared	Multiple, same	Possibly all different
File systems	One, shared	One, shared	Each node has own
Administration	One organization	One organization	Many organizations

Figure 8-29. Comparison of three kinds of multiple CPU systems.

7.4 Distributed Systems (2)

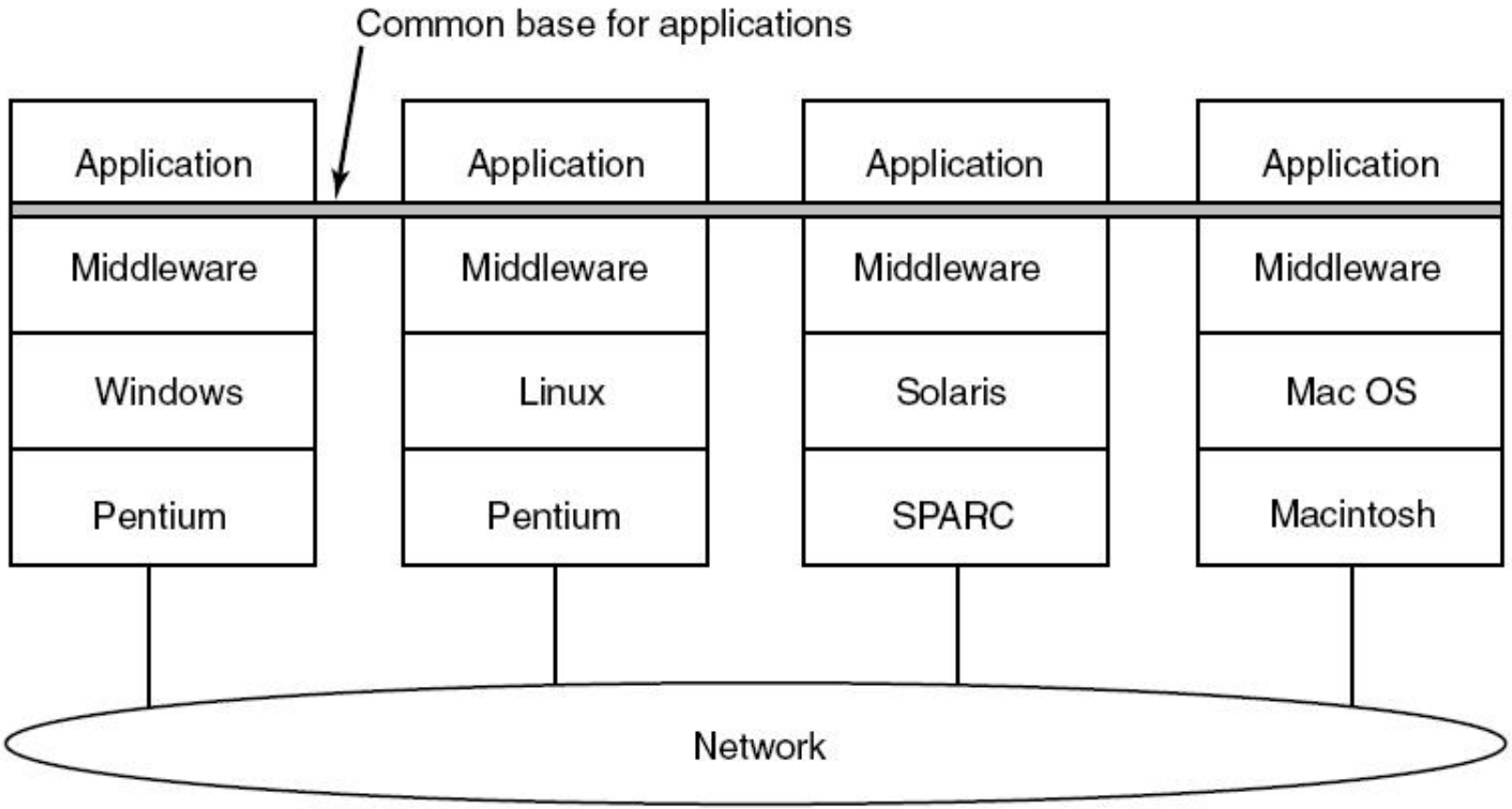


Figure 8-30. Positioning of middleware in a distributed system.

7.4.1 Ethernet

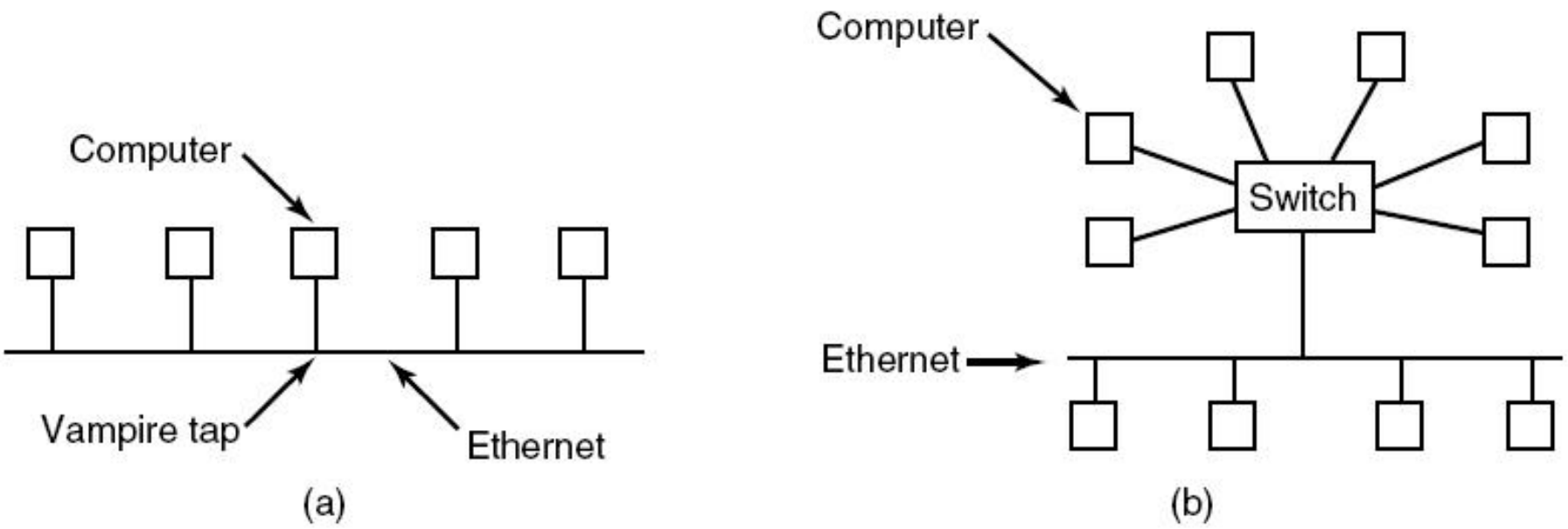


Figure 8-31. (a) Classic Ethernet. (b) Switched Ethernet.

7.4.1 The Internet

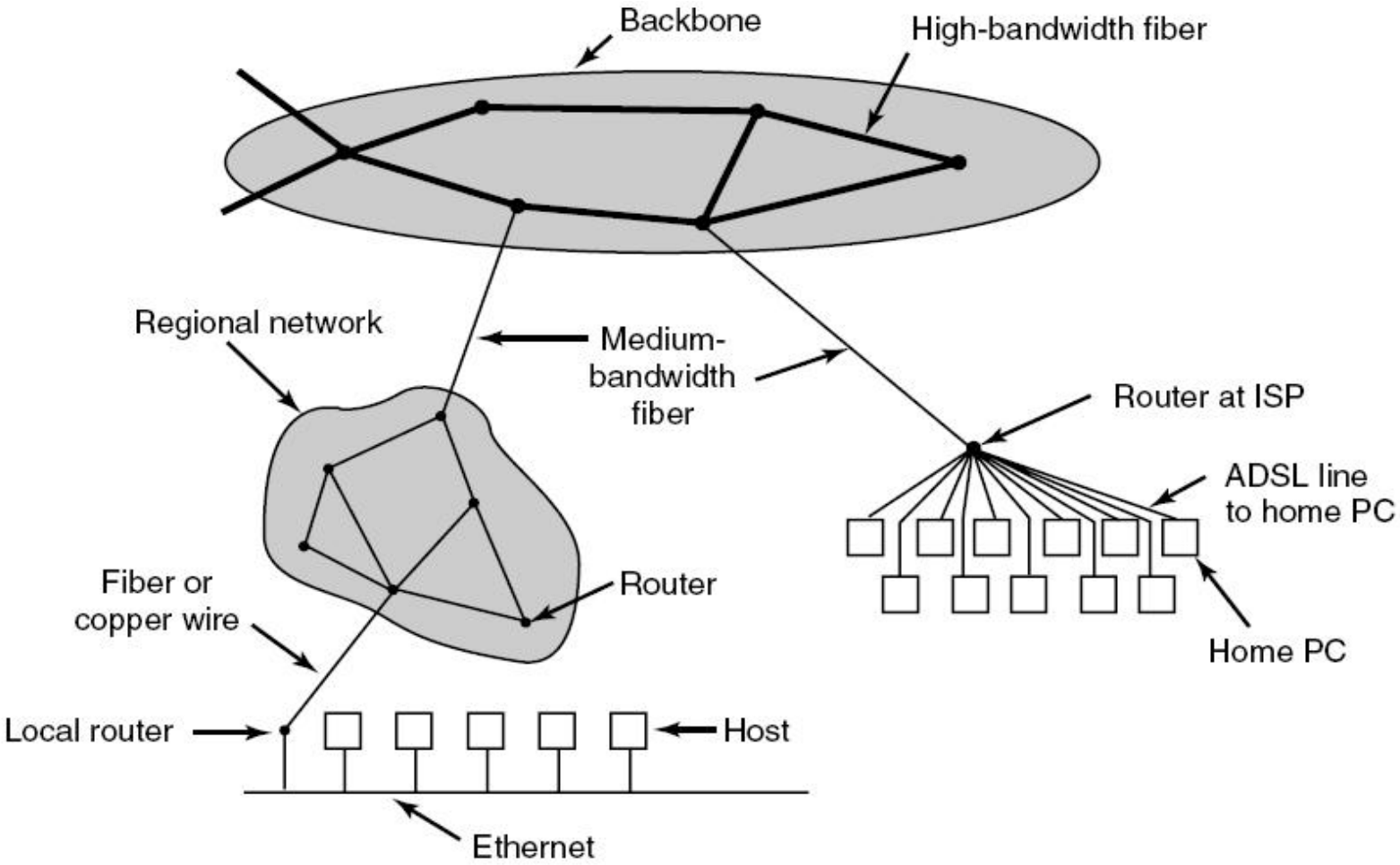


Figure 8-32. A portion of the Internet.

7.4.2 Network Protocols (1)

	Service	Example
Connection-oriented	Reliable message stream	Sequence of pages of a book
	Reliable byte stream	Remote login
	Unreliable connection	Digitized voice
Connectionless	Unreliable datagram	Network test packets
	Acknowledged datagram	Registered mail
	Request-reply	Database query

Figure 8-33. Six different types of network service.

7.4.2 Network Protocols (2)

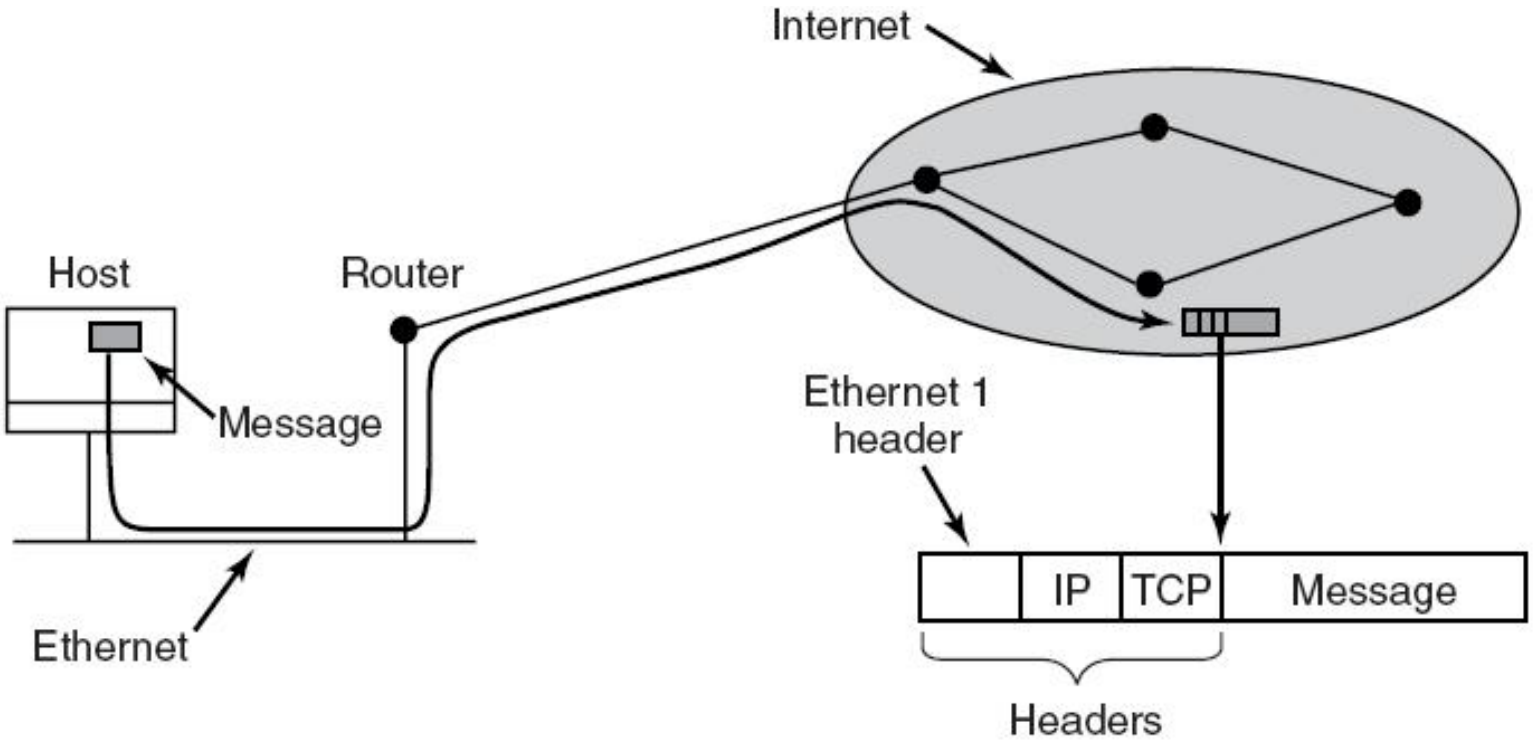


Figure 8-37. Accumulation of packet headers.

7.4.3Document-Based Middleware (1)

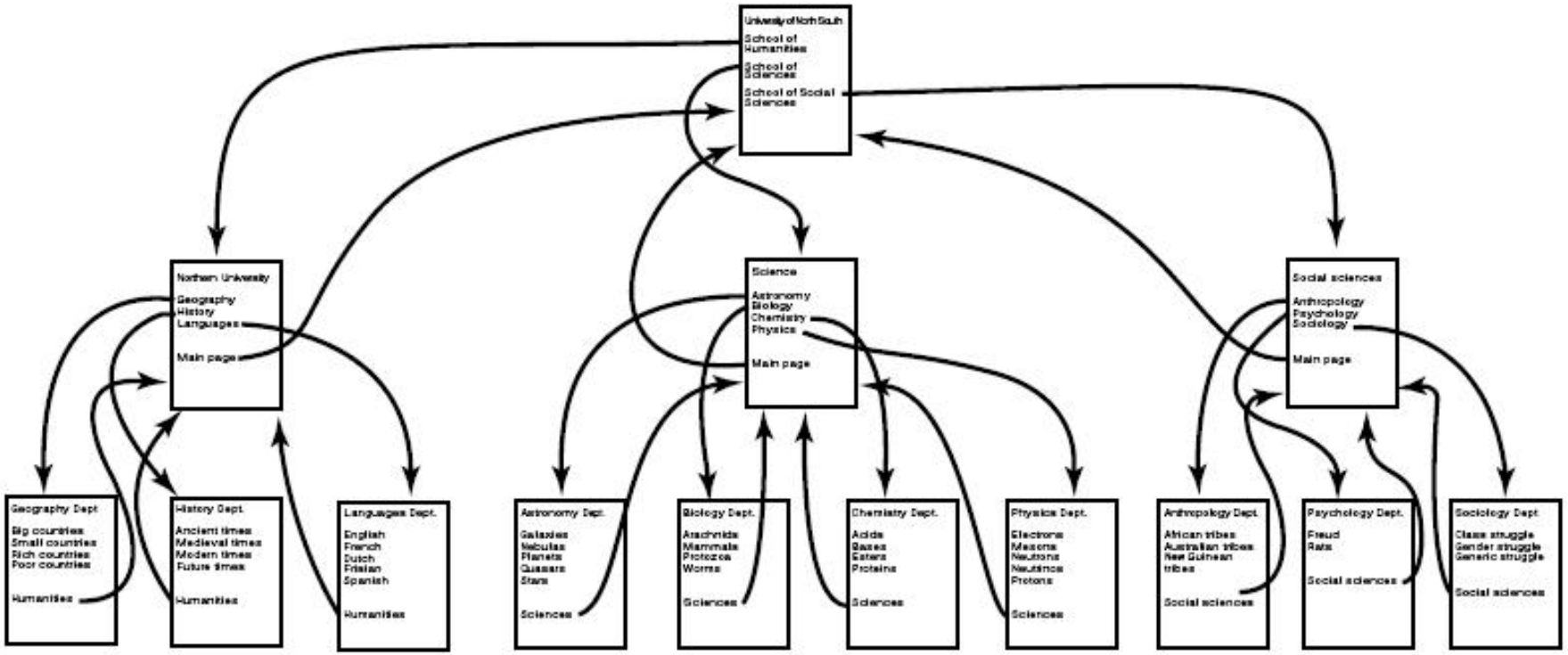


Figure 8-35. The Web is a big directed graph of documents.

7.4.3 Document-Based Middleware (2)

When the browser gets the page *http://www.minix3.org/doc/faq.html.*

- The browser asks DNS for the IP address of www.minix3.org.
- DNS replies with 130.37.20.20.
- The browser makes a TCP connection to port 80 on 130.37.20.20.
- It then sends a request asking for the file doc/faq.html.

. . .

7.4.3 Document-Based Middleware (3)

. . .

- The www.acm.org server sends the file doc/faq.html.
- The TCP connection is released.
- The browser displays all the text in doc/faq.html.
- The browser fetches and displays all images in doc/faq.html.

7.4.4 File-System-Based Middleware Transfer Model

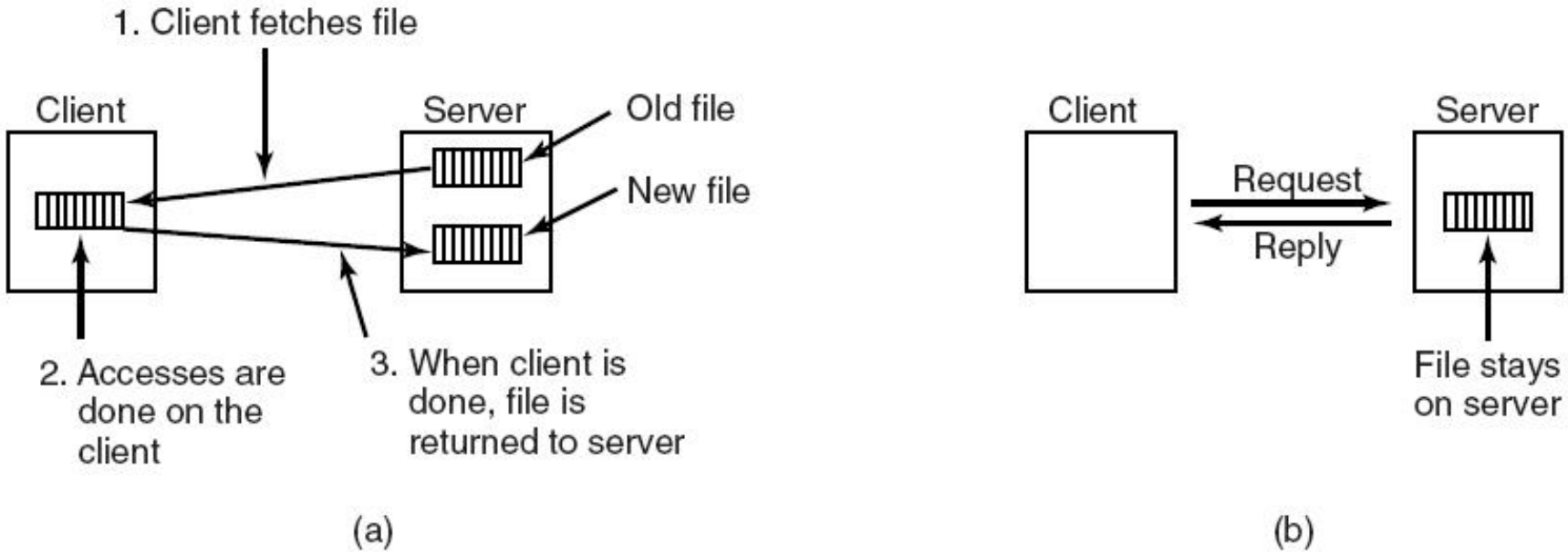


Figure 8-36. (a) The upload/download model. (b) The remote access model.

7.4.4 The Directory Hierarchy (1)

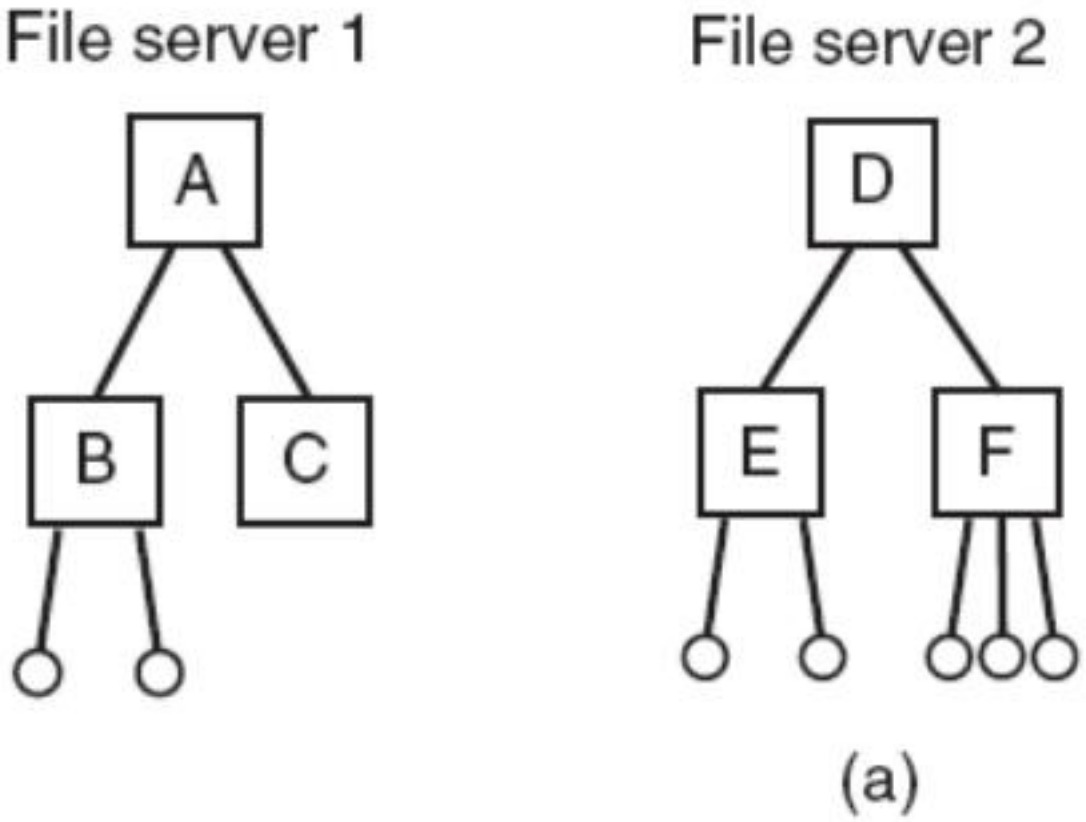


Figure 8-37. (a) Two file servers. The squares are directories and the circles are files.

7.4.4 The Directory Hierarchy (2)

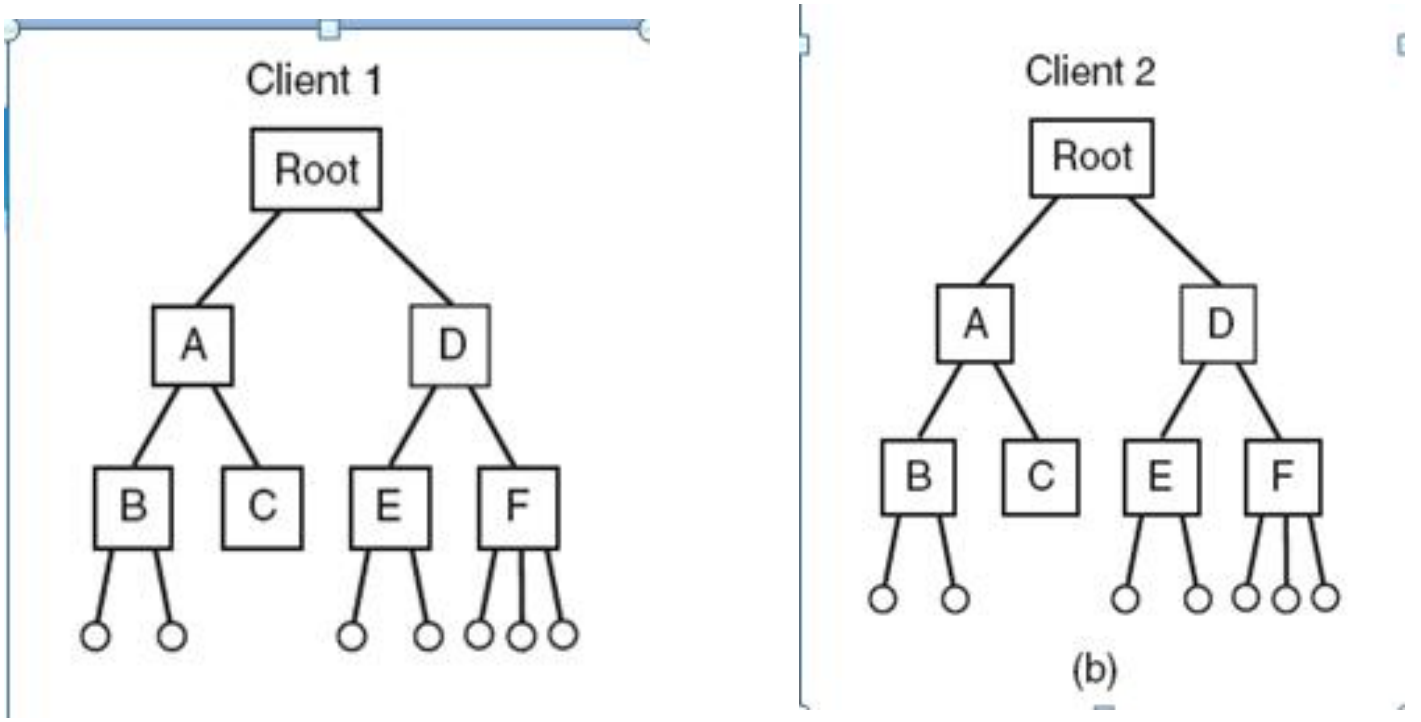


Figure 8-37. (b) A system in which all clients have the same view of the file system.

7.4.4 The Directory Hierarchy (3)

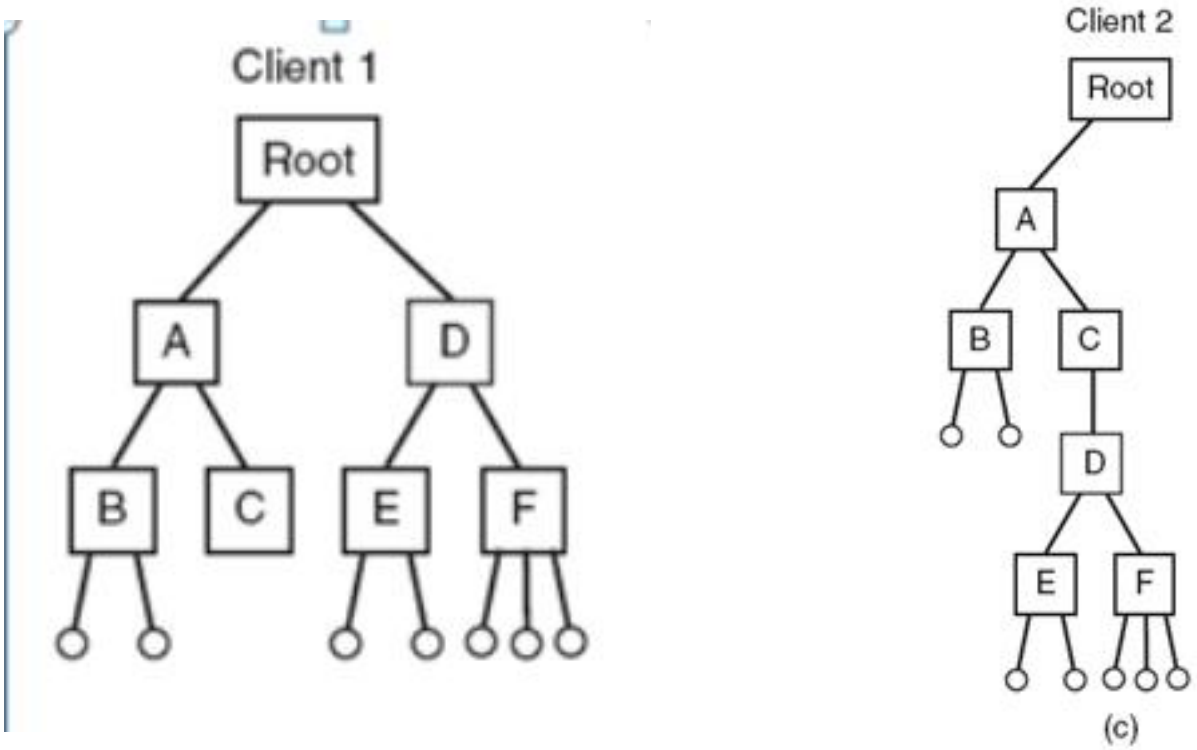


Figure 8-37. (c) A system in which different clients may have different views of the file system.

7.4.4 Naming Transparency

Three common approaches to file and directory naming in a distributed system:

- Machine + path naming, such as */machine/path* or *machine:path*.
- Mounting remote file systems onto the local file hierarchy.
- A single name space that looks the same on all machines.

7.4.4 Semantics of File Sharing(1)

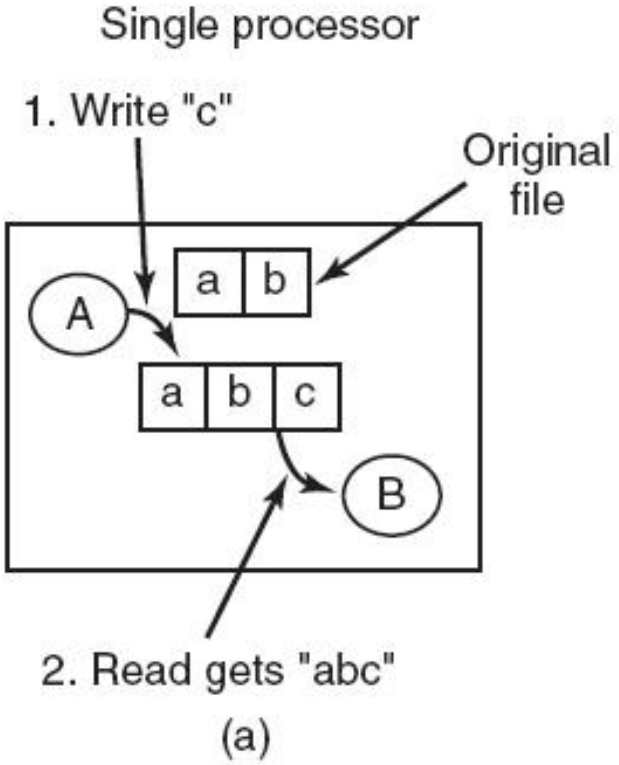


Figure 8-38. (a) Sequential consistency.

7.4.4 Semantics of File Sharing(2)

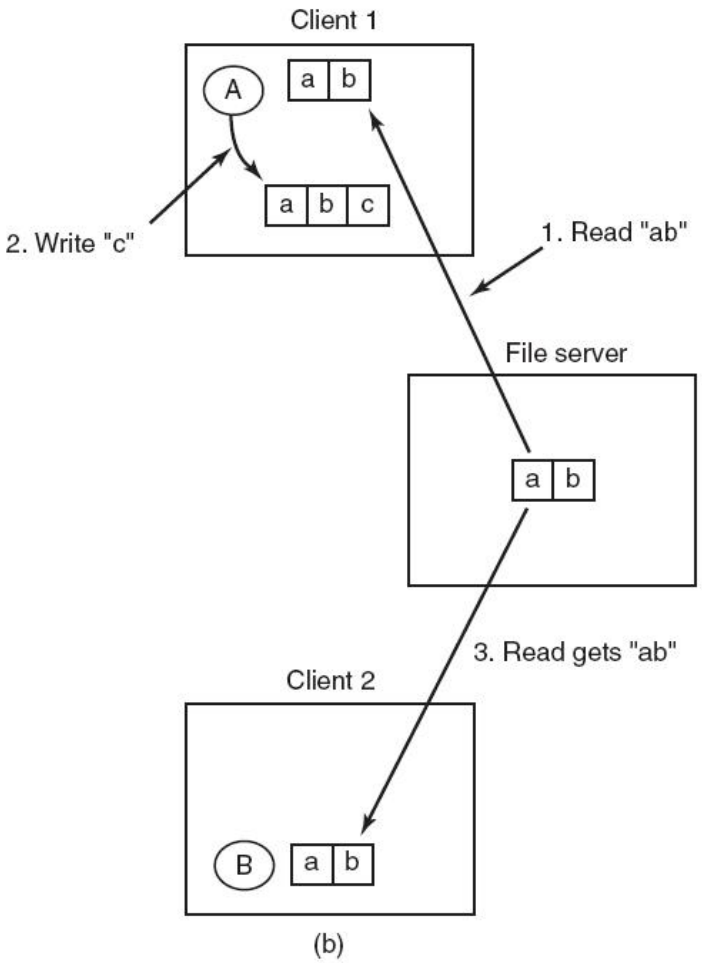


Figure 8-38. (b) In a distributed system with caching, reading a file may return an obsolete value.

7.4.5 Object-Based Middleware

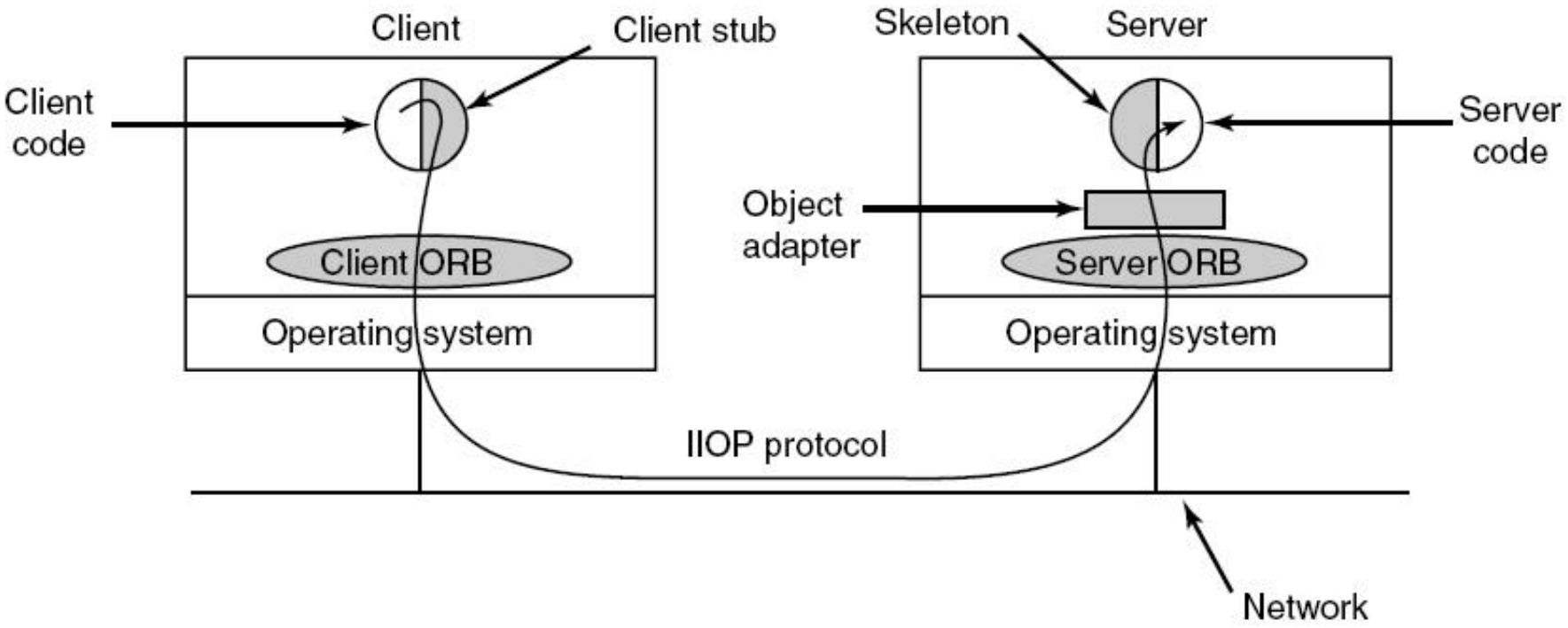


Figure 8-39. The main elements of a distributed system based on CORBA. The CORBA parts are shown in gray.

7.4.6 Coordination-Based Middleware (1)

Linda

- A system for communication and synchronization
- Independent processes communicate via an abstract tuple space
- A tuple is a structure of one or more fields, each of which is a value of some type supported by the base language

```
("abc", 2, 5)
("matrix-1", 1, 6, 3.14)
("family", "is-sister", "Stephany", "Roberta")
```

Figure 8-40. Three Linda tuples.

7.4.6 Matching Tuples

A match occurs if the following three conditions are all met:

- The template and the tuple have the same number of fields.
- The types of the corresponding fields are equal.
- Each constant or variable in the template matches its tuple field.

7.4.6 Publish/Subscribe

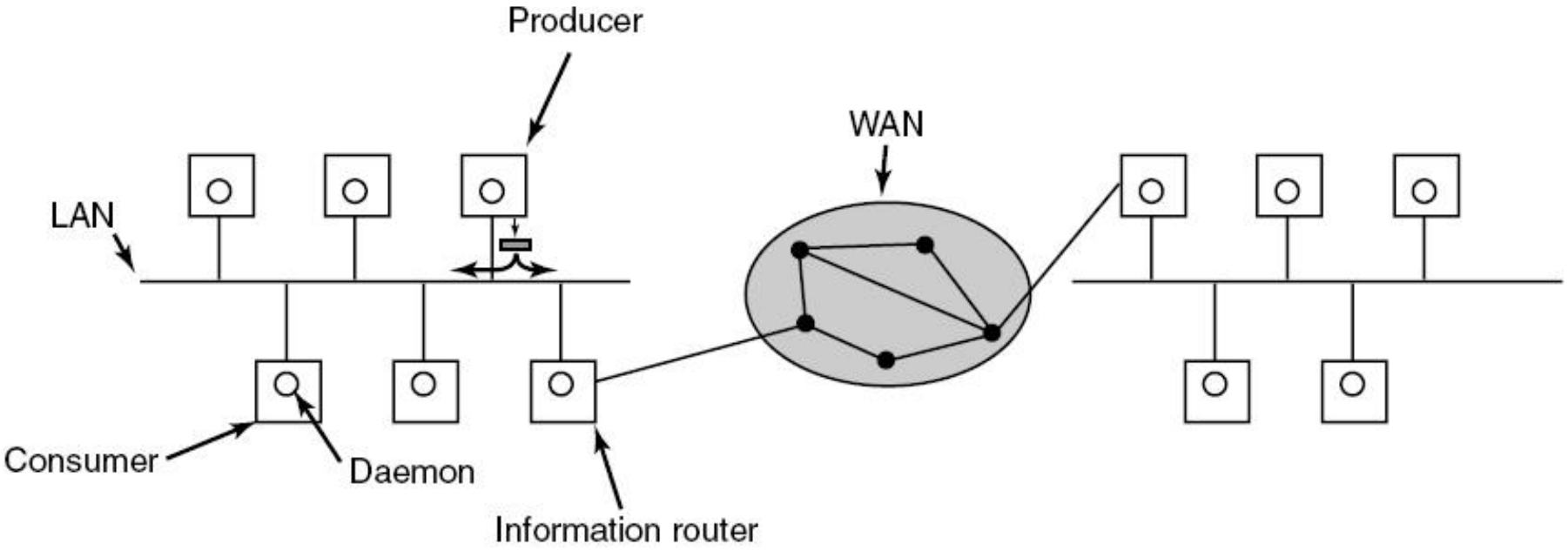


Figure 8-41. The publish/subscribe architecture.

7.4.6 Jini

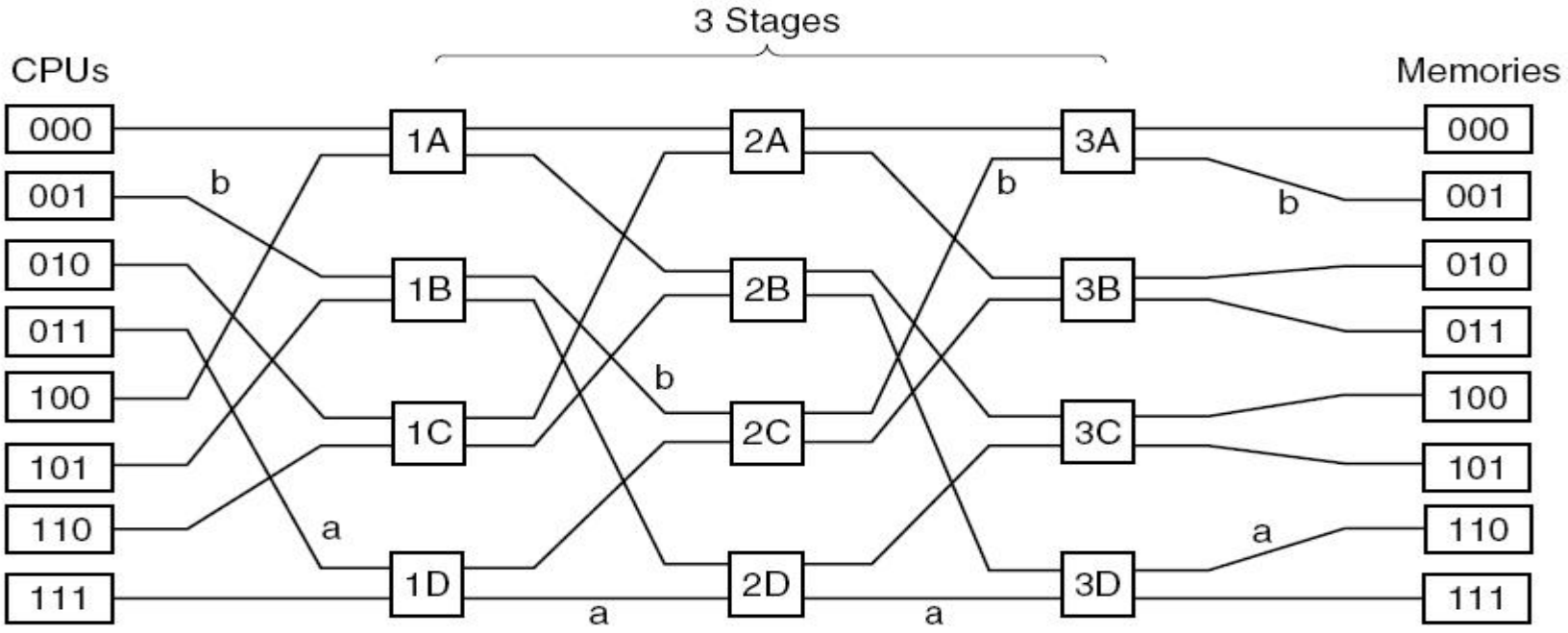
Jini clients and services communicate and synchronize using JavaSpaces.

Methods defined in a JavaSpace:

- **Write**: put a new entry into the JavaSpace.
- **Read**: copy an entry that matches a template out of the JavaSpace.
- **Take**: copy and remove an entry that matches a template.
- **Notify**: notify the caller when a matching entry is written.

Problem

- Suppose that the wire between switch 2A and switch 3B in the omega network of the figure breaks. Who is cut off from whom?



Reading Materials

- The model of run time

谢 谢！