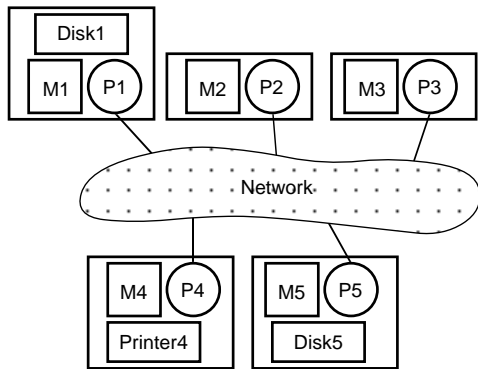


## Classification of Operating Systems (Review)

### ■ “True” Distributed Operating System

- Loosely-coupled hardware
  - No shared memory, but provides the “feel” of a single memory
- Tightly-coupled software
  - One single OS, or at least the feel of one
- Machines are somewhat, but not completely, autonomous

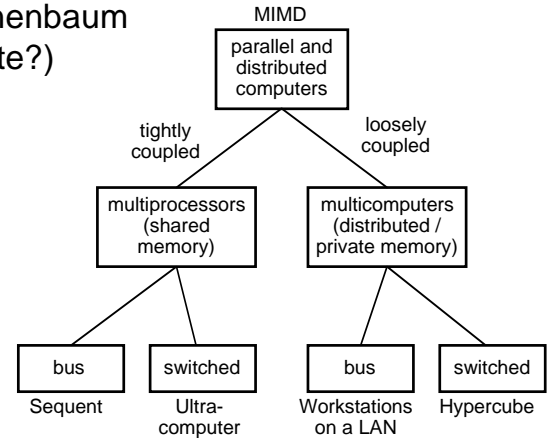


1

Spring 2001, Lecture 02

## Classification of MIMD Architectures (Review)

### ■ Tanenbaum (date?)



### ■ Tightly coupled $\approx$ parallel processing

- Processors share clock and memory, run one OS, communicate frequently

### ■ Loosely coupled $\approx$ distributed computing

- Each processor has its own memory, runs its own OS (?), communicates infrequently

2

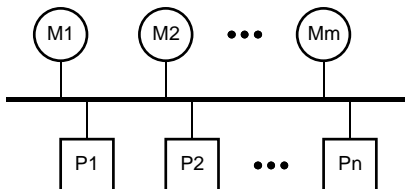
Spring 2001, Lecture 02

## Classification of Multiprocessors, Based on Interconnection Network

### ■ Three main types of interconnection:

- Bus
- Switch (crossbar, multistage switch)

### ■ Bus-based interconnection



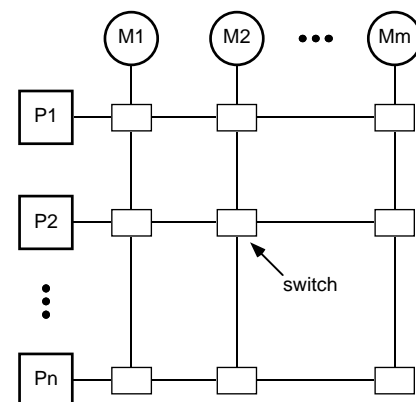
- ✓ Simple
- ✓ No need for routing — just broadcast
- ✗ Contention for access to bus (does not scale well)
- ✗ Limited to 32, maybe 64, processors

3

Spring 2001, Lecture 02

## Classification of Multiprocessors, Interconnection Network (cont.)

### ■ Crossbar switch



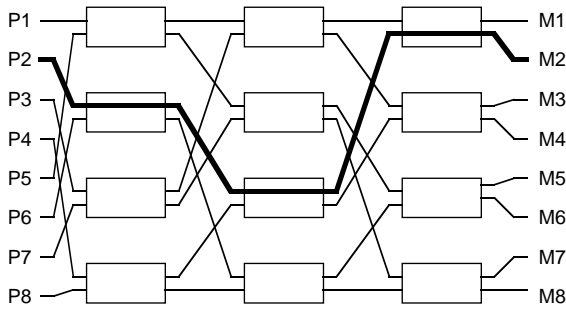
- ✓ Usually no contention for memory access — multiple memories can be accessed in parallel
- ✓ Simple routing
- ✗ Number of crossbar switches grows quadratically

4

Spring 2001, Lecture 02

## Classification of Multiprocessors, Interconnection Network (cont.)

### ■ Multistage switch



- ✓ Reduced number of switches
- ✗ Increased communication delay
- ✗ Increased contention for memory access
- ✗ Complex network

5

Spring 2001, Lecture 02

## Classification of Multicomputers, Based on Interconnection Network

### ■ Two main types of interconnection:

- Switching network
- LAN (local area network)

### ■ Switching network

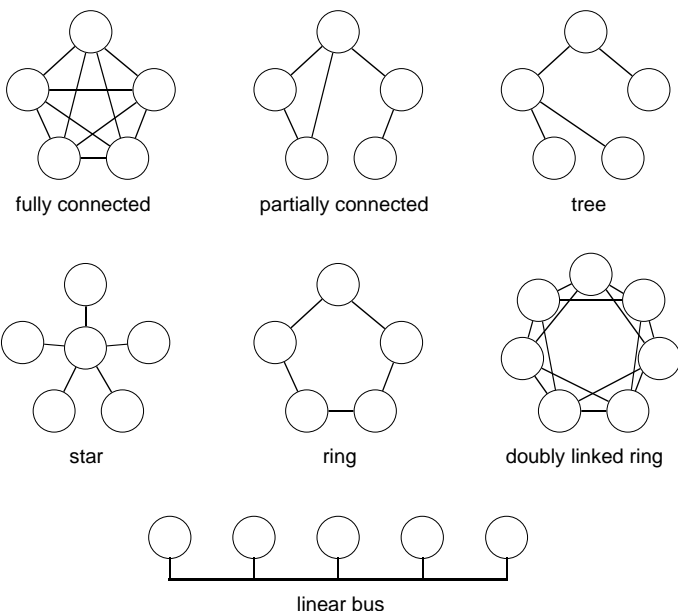
- Grid
  - $n^2$  nodes arranged as an  $n \times n$  grid
  - ✗ Maximum route proportional to  $2n$
  - ✗ Most messages take multiple hops
- Hypercube
  - A  $n$ -degree hypercube ( $n$ -cube) consists of  $2^n$  nodes (processors) arranged in an  $n$ -dimensional cube, where each node is connected to  $n$  other nodes
  - ✓ Maximum route proportional to  $n$
  - ✗ Most messages take multiple hops

6

Spring 2001, Lecture 02

## Classification of Multicomputers, Based on Interconnection Network

### ■ LAN-based interconnection



7

Spring 2001, Lecture 02

## Classification of Multiprocessors and Multicomputers, Based on Memory Access

### ■ UMA — Uniform Memory Access

- Main memory is at a central location

### ■ NUMA — Non-Uniform Memory Access

- Main memory is physically partitioned, with each partition attached to a different processor
- Each processor can access its own memory (fast), or the memory of another processor (slow)

### ■ NORMA — No Remote Memory Access

- Main memory is physically partitioned, with each partition attached to a different processor
- A processor can not access the memory of another processor

8

Spring 2001, Lecture 02

## Distributed System Models

- **Minicomputer model**
  - Several minicomputers connected to a network, each with several terminals
- **Workstation-server model**
  - Specialized workstations running servers: file server, print server, etc.
  - Good resource sharing (printers, etc.), cheap workstations (don't need big disks)
- **Workstation model**
  - Many workstations on a network
  - System automatically transfers processes to idle workstations (process migration)
- **Processor-pool model**
  - Terminals connect to network, pool of processors connect to network

9

Spring 2001, Lecture 02

## Goals of a Distributed System: Transparency

- **Access transparency**
  - User is unaware whether a resource is local or remote
- **Location transparency**
  - User is unaware of physical location of hardware or software resources
- **Migration transparency**
  - User is unaware if OS moves processes or resources (e.g., files) move to a different physical locations
- **Replication transparency**
  - Resource duplication is invisible to users
- **Concurrency transparency**
  - Resource sharing is invisible to users

10

Spring 2001, Lecture 02

## Other Goals of a Distributed System

- **Performance**
  - Cache data, minimize copying data, minimize network traffic, use threads to take advantage of parallelism
- **Scalability**
  - System should adapt well to increased load (avoid central control, do as much work locally as possible)
- **Flexibility**
  - System should be easy to modify and enhance (use microkernel and user-level processes providing services)
- **Reliability**
  - Must avoid, tolerate, detect, and recover from *faults* — mechanical or algorithmic defects that can generate an error

11

Spring 2001, Lecture 02

## Why Use Distributed Systems? What are the Advantages?

- **Natural programming model**
  - Some applications (database in large company) are inherently distributed
- **Resource sharing**
  - Expensive (scarce) resources need not be replicated for each processor
- **Price / performance**
  - Network of workstations provides more MIPS for less \$ than a mainframe does
- **Reliability**
  - Replication of processors and resources yields fault tolerance
- **Scalability**
  - Modular structure makes it easier to add or replace processors and resources

12

Spring 2001, Lecture 02