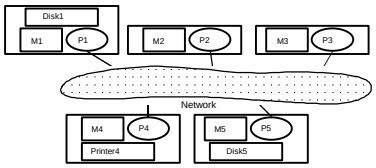
Classification of Distributed Systems
Properties of Distributed Systems

- motivation: advantages of distributed systems
- Classification
  - architecture based
    - on interconnection
    - on memory access
  - design based (OS models)
- Design issues of a distributed system
  - transparency
  - heterogeneity
  - autonomy
  - others

Classification of Operating Systems (cont.)

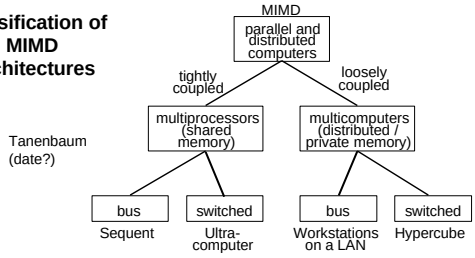
- "True" Distributed Operating System
  - Loosely-coupled hardware
    - No shared memory, but provides the "feel" of a single memory
  - Tightly-coupled software
    - One single OS, or at least the feel of one
  - Machines are somewhat, but not completely, autonomous



Why Use Distributed Systems?
What are the Advantages?

- Price / performance
  - Network of workstations provides more MIPS for less $ than a mainframe does
- Higher performance:
  - *n* processors potentially give *n* times the computational power
- Resource sharing:
  - Expensive (scarce) resources need not be replicated for each processor
- Scalability:
  - Modular structure makes it easier to add or replace processors and resources
- Reliability:
  - Replication of processors and resources yields fault tolerance

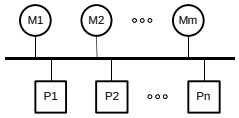
Classification of MIMD Architectures



- Tightly coupled ≈ *parallel processing*
  - Processors share clock and memory, run one OS, communicate frequently
- Loosely coupled ≈ *distributed computing*
  - Each processor has its own memory, runs its own OS (?), communicates infrequently

Classification of Multiprocessors Based on Interconnection Network

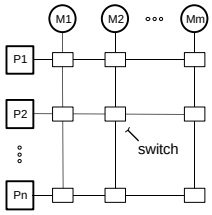
- Three main types of interconnection:
  - Bus
  - Switch (crossbar, multistage switch)



- Bus-based interconnection
  - Simple
  - Bus is a broadcast medium
  - Contention for access to bus (does not scale well)
  - Complicates caches (need *snoopy* cache)

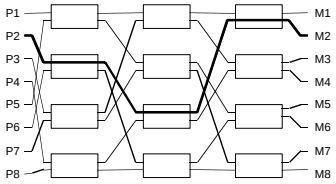
Classification of Multiprocessors Based on Interconnection Network (cont.)

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



Classification of Multiprocessors Based on Interconnection Network (cont.)

- Multistage switch
  - Reduced number of switches
  - Increased communication delay
  - Increased contention for memory access
  - Complex network



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 x n grid
      - Maximum route proportional to n^2
      - 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

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

Distributed System Models

- Minicomputer model
  - Several minicomputers connected to a network, each with several terminals
- Workstation model
  - Many workstations connected to a network
  - Particularly useful if users can use remote workstations (process migration)
- Workstation-server model
  - Same, plus more some machines run as servers: file server, print server, etc.
  - Good resource sharing (printers, etc.), cheap workstations (don't need big disks)
- Processor-pool model
  - Terminals connect to network, pool of processors connect to network

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
    - location transparency
    - user mobility
- *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

Goals of a Distributed System: Support Heterogeneity

- *Heterogeneity* means "consisting of a number of completely different elements"
- *Computer hardware* heterogeneity
  - Different computer architectures (e.g., instruction sets, data representations) of components in distributed systems
- *Network* heterogeneity
  - Different transmission media, signaling, network interfaces, and protocols
- *Software* heterogeneity
  - Different operating systems, application programs

↙ Support for heterogeneity remains a mostly unsolved problem

Goals of a Distributed System: Right Degree of Autonomy

- *Autonomy* is a measure of the independence of the components in a distributed system
- Low degree of autonomy = dependent
  - Inflexible
  - Little robustness in the presence of failures
- High degree of autonomy = independent
  - More flexibility
  - High redundancy
  - May still require some central control
  - Poor resource sharing and coordination
- Determining the right degree of autonomy in a distributed system is a difficult problem

Other design issues of a distributed system

- Fault tolerance:
  - fault avoidance
  - fault tolerance
    - redundancy
    - distributed control
  - fault detection/recovery
- flexibility needed
  - to ease modification
  - ease enhancement
- performance
  - batch if possible
  - cache when possible
  - minimize network traffic
  - parallelize

Other properties of a distributed system (cont.)

- scalability
  - avoid centralized entities
  - avoid centralized algorithms
  - spread the load
- security
  - hard because there is no single point of control/authentication
    - the communicating parties should be sure of each other identity (be able to trust each other)
    - the communicating parties should be sure that the communication is not compromised (altered or eavesdropped)