

Distributed Systems

Why Distributed Systems?

- Many systems that we use on a daily basis are distributed
 - World wide web, Google
 - Amazon.com
 - Peer-to-peer file sharing systems
 - SETI@Home
 - Grid and cluster computing
 - Modern networked computers
- Useful to understand how such real-world systems work
- Course covers basic principles for designing distributed systems

Definition of a Distributed System

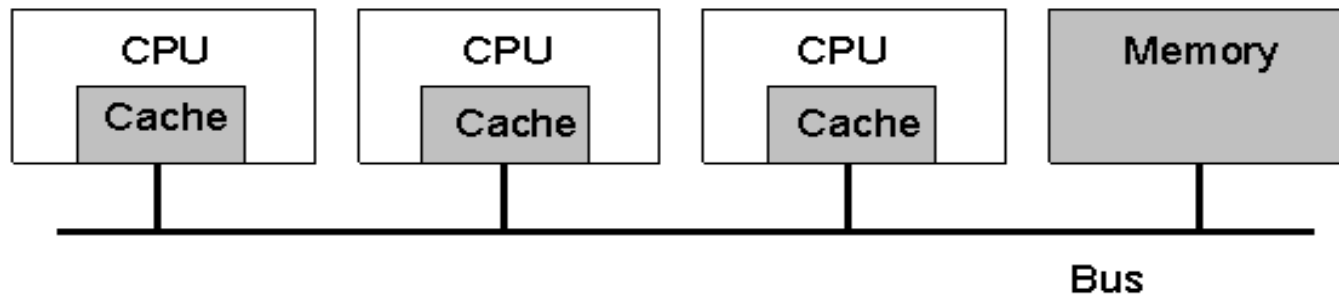
- A distributed system:
 - Multiple connected CPUs working together
 - A collection of independent computers that appears to its users as a single coherent system
- Examples: parallel machines, networked machines

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

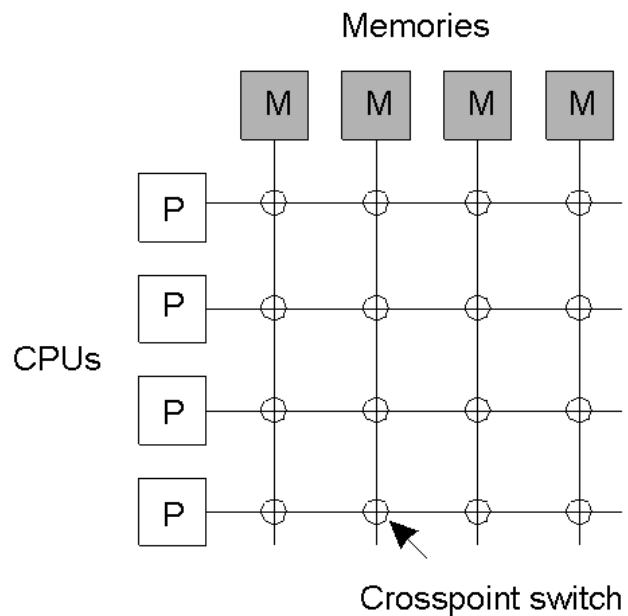
Hardware Concepts: Multiprocessors (1)

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



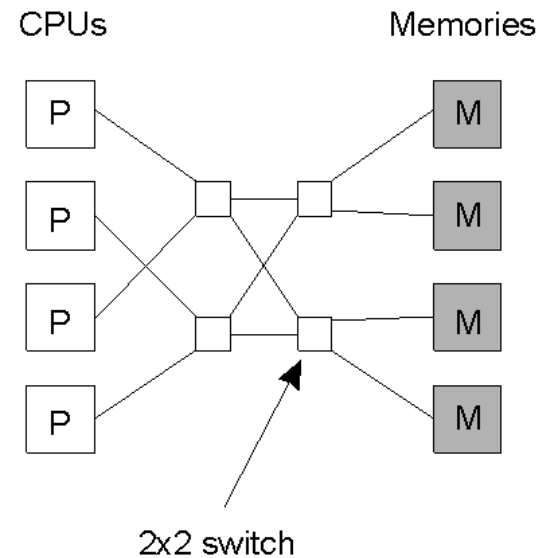
Multiprocessors (2)

a) A crossbar switch



(a)

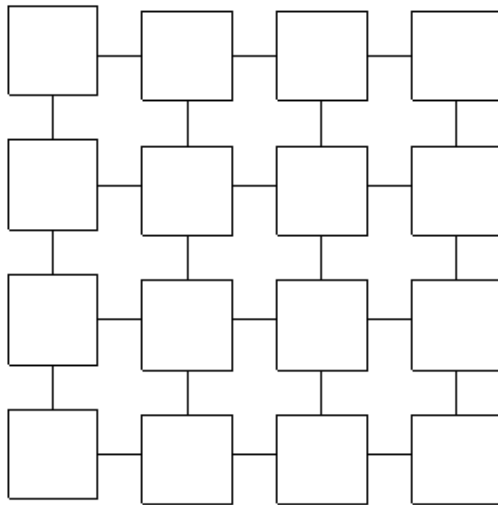
b) An omega switching network



(b)

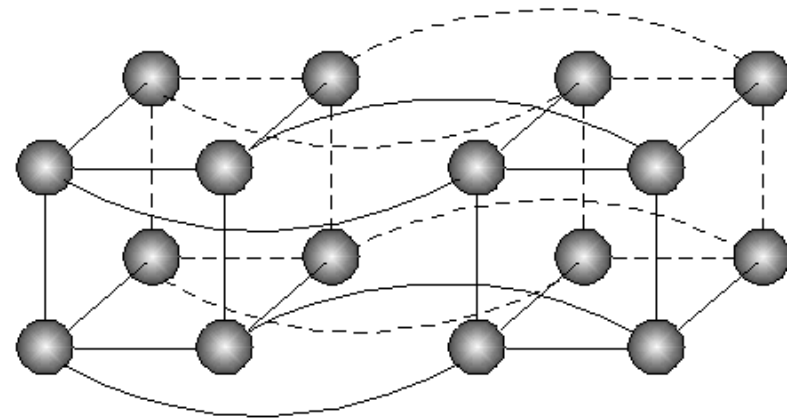
Homogeneous Multicomputer Systems

a) Grid



(a)

b) Hypercube



(b)

Scaling Techniques

- *Principles* for good decentralized algorithms
 - No machine has complete state
 - Make decision based on local information
 - A single failure does not bring down the system
 - No global clock
- *Techniques*
 - Asynchronous communication
 - Distribution
 - Caching and replication

Distributed Systems Models

- Minicomputer model (e.g., early networks)
 - Each user has local machine
 - Local processing but can fetch remote data (files, databases)
- Workstation model (e.g., Sprite)
 - Processing can also migrate
- Client-server Model (e.g., V system, world wide web)
 - User local workstation
 - Powerful workstations serve as servers (file, print, DB servers)
- Processor pool model (e.g., Amoeba, Plan 9)
 - Terminals are Xterms or diskless terminals
 - Pool of backend processors handle processing

Distributed System Models (contd)

- Cluster computing systems / Data centers
 - LAN with a cluster of servers + storage
 - Linux, Mosix, ..
 - Used by distributed web servers, scientific applications, enterprise applications
- Grid computing systems
 - Cluster of machines connected over a WAN
 - SETI @ home
- WAN-based clusters / distributed data centers
 - Google, Amazon, ...
- Virtualization and data centers

Emerging Models

- Distributed Pervasive Systems
 - “smaller” nodes with networking capabilities
 - Computing is “everywhere”
 - Home networks: TiVO, Windows Media Center, ...
 - Mobile computing: smart phones, iPODs, Car-based PCs
 - Sensor networks
 - Health-care: personal area networks