Electrical Engineering and Computer Science

EECS 358 - INTRODUCTION TO PARALLEL COMPUTING

Lecture 3

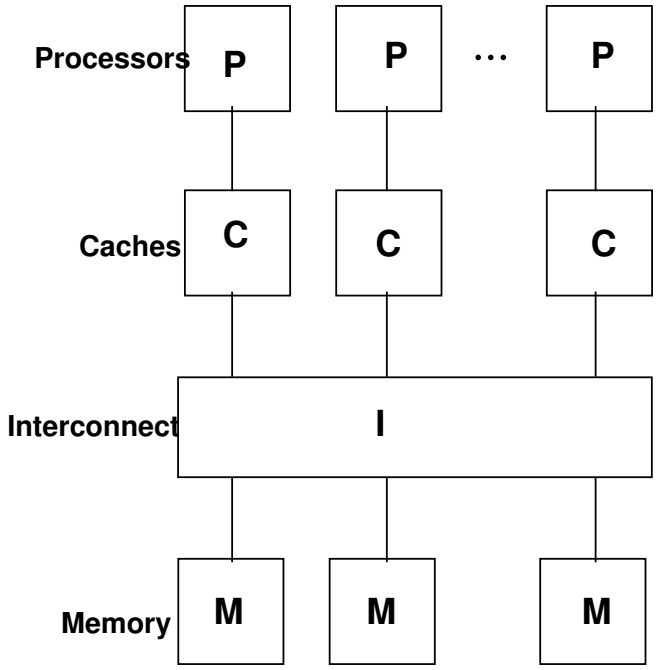
Architectural Features

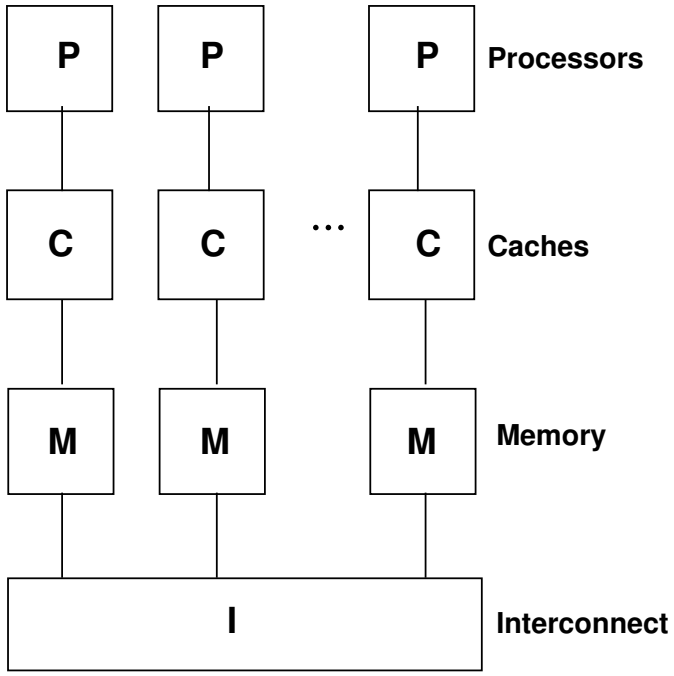
Outline

- Overview of shared memory architectures
- Memory organizations
- Interconnect organizations
- Case studies

Shared Address Space Machines

- All processors share a single global address space
- Single address space facilitates a simple programming model





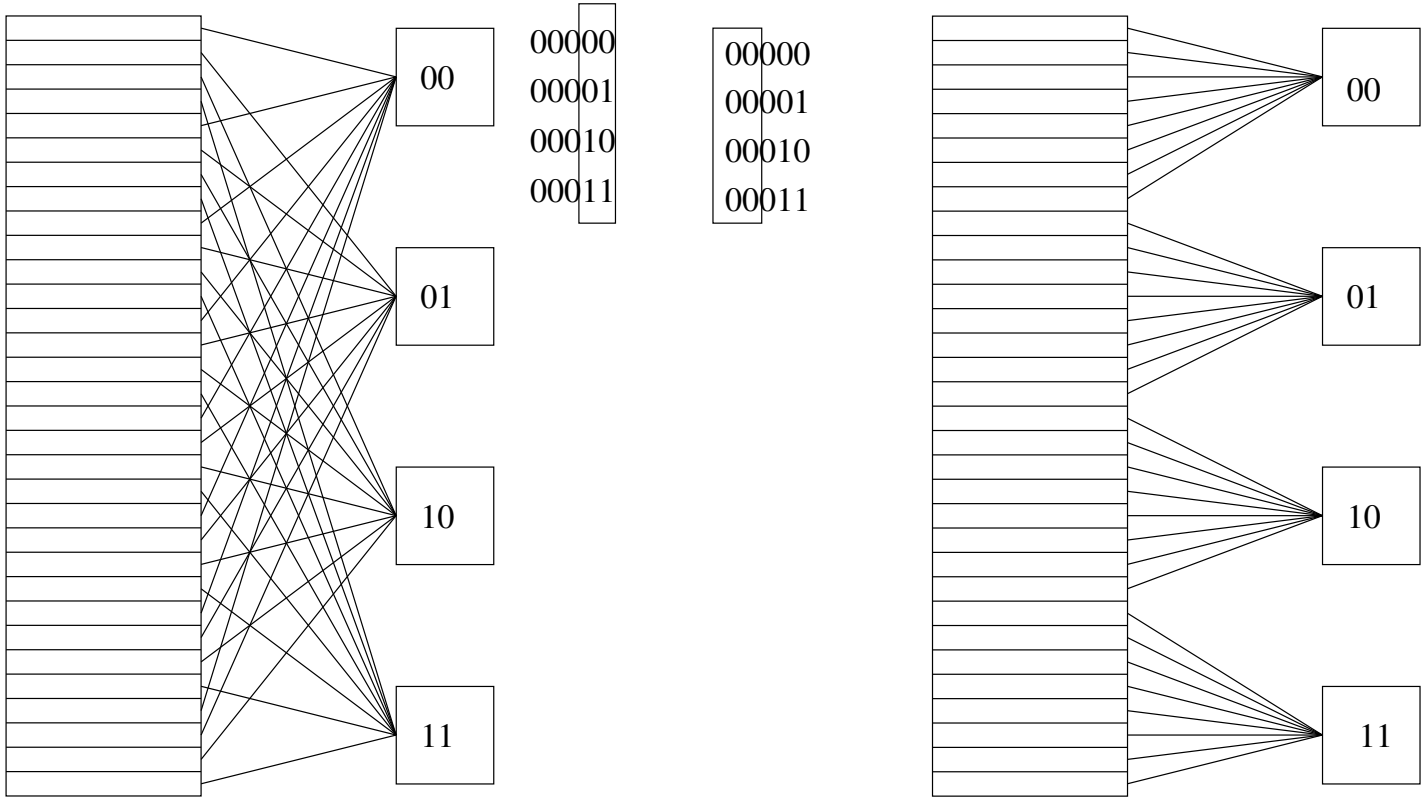
Shared Address Space Machines

- Important aspects are:
  - Memory organization (physically shared)
  - Interconnect
  - Cache coherence mechanism
- These aspects determine:
  - Performance - programming techniques
  - Scalability
  - Cost

Memory Organization

- Single memory module shared among processors causes sequentialization of accesses
- Memory interleaving
  - Splits memory across multiple modules (banks)
  - Non-overlapping regions of address space mapped to banks
  - Banks service read/write requests independently
- Low-order interleaving:
  - Low-order bits of address used to select bank
  - Enables block transfers and reduces bank conflicts
- High-order interleaving
  - High-order bits of address used to select bank
  - Block transfer not possible and high rate of bank conflicts

Memory Organization

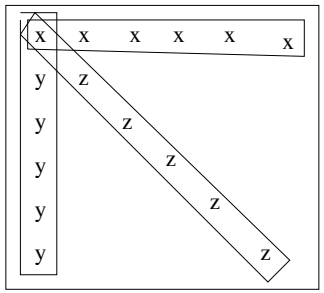


Low Order Interleaving

High Order Interleaving

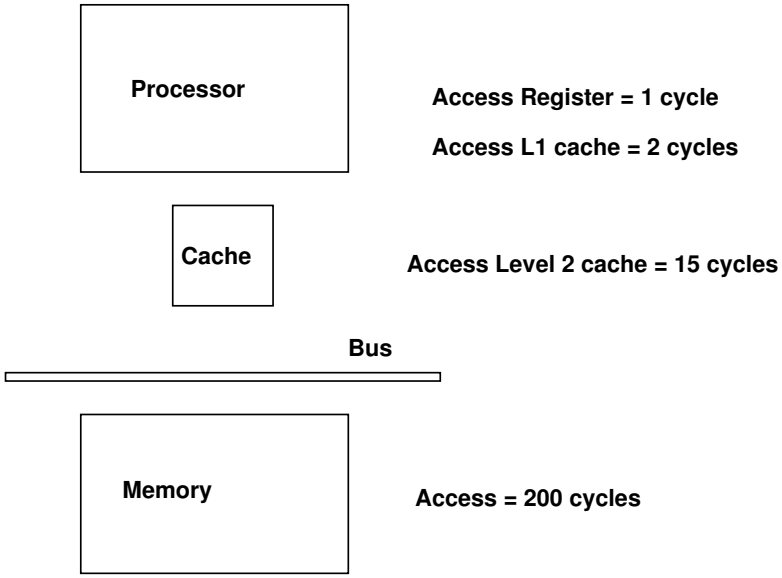
Memory Organization

- Typically, low-order interleaving is used because it reduces bank conflicts
- Programming issues:
  - Must spread accesses across banks to avoid bank conflicts
  - Involves data placement as well as code restructuring



Caches

- Used to exploit temporal locality in programs

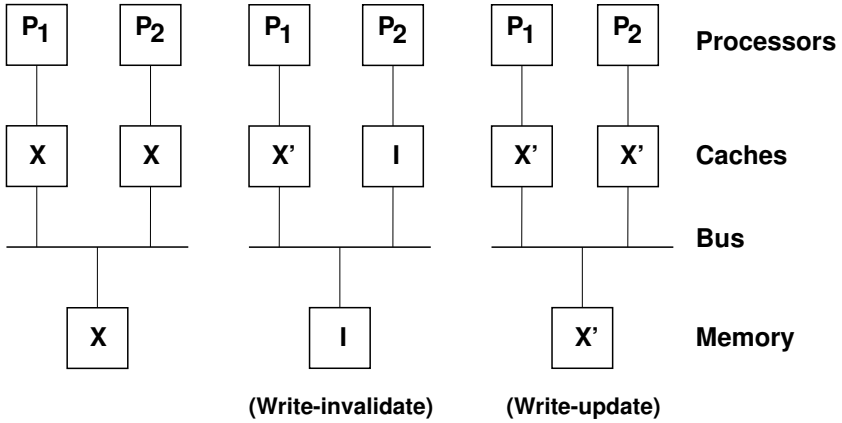


Cache Coherence Protocols

- Needed to avoid incorrect results due to sharing of cache blocks
- On write to a block at a processor:
  - Invalidate copies of the block for other processors - write-invalidate
  - Update copies of the block for other processors - write-update
- On write to a block at a processor:
  - Update block in main memory - write-through
  - Don't update block in main memory - write-back
- Types of cache coherence mechanisms:
  - Snoopy cache coherence protocols
  - Directory-based cache coherence protocols

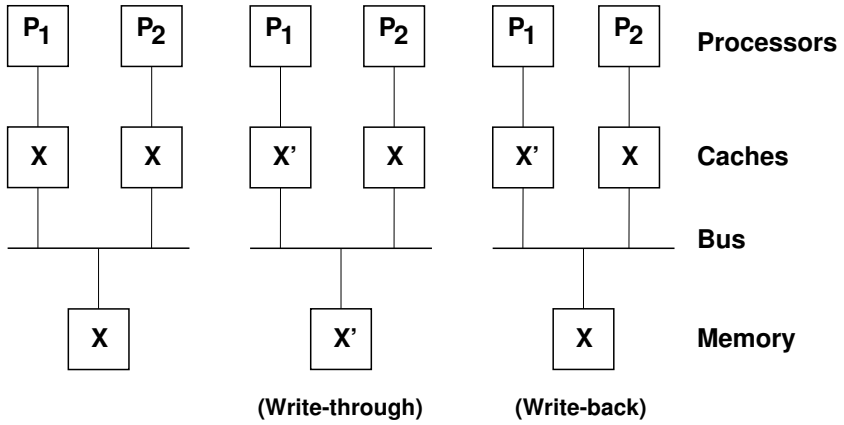
Cache Coherence Protocols

- Write-invalidate versus write-update



Cache Coherence Protocols

- Write-back versus write-through



Snoopy Cache Coherence Protocols

- All processors monitor bus traffic for information affecting their current cache blocks
- Can use state-transition graphs to represent protocols
- Write-invalidate protocol for a write-through cache

Directory-based Cache Coherence Protocols

- Snoopy protocols involve repeated broadcasts using precious bandwidth - bus systems are an exception
- Directory based schemes track processors using each cache block
- Accesses to memory through the directory - helps locate latest version of a cache block
- Directory organization:
  - Centralized - information for all cache blocks together
  - Distributed - information for cache blocks in each memory module

Effect of Caches on Performance

- False sharing:
  - Processors write to non-overlapping parts of the same block
  - Block ping-pongs between processors
- Spatial locality
  - When a block is fetched from memory, all elements must be used
- Both effects depend on the underlying architectural parameters

Interconnects: Design Choices

- Operational Characteristics:
  - Topology - dynamic or static
  - Timing protocol - sychronous or asynchronous
  - Switching method - circuit or packet
  - Control strategy - centralized or distributed
- Performance criteria:
  - Functionality - type of support for various services
  - Network latency - worst case time for unit message
  - Bandwidth - maximum data transfer rate
  - Hardware complexity - cost of implementation
  - Scalability - ease of expansion

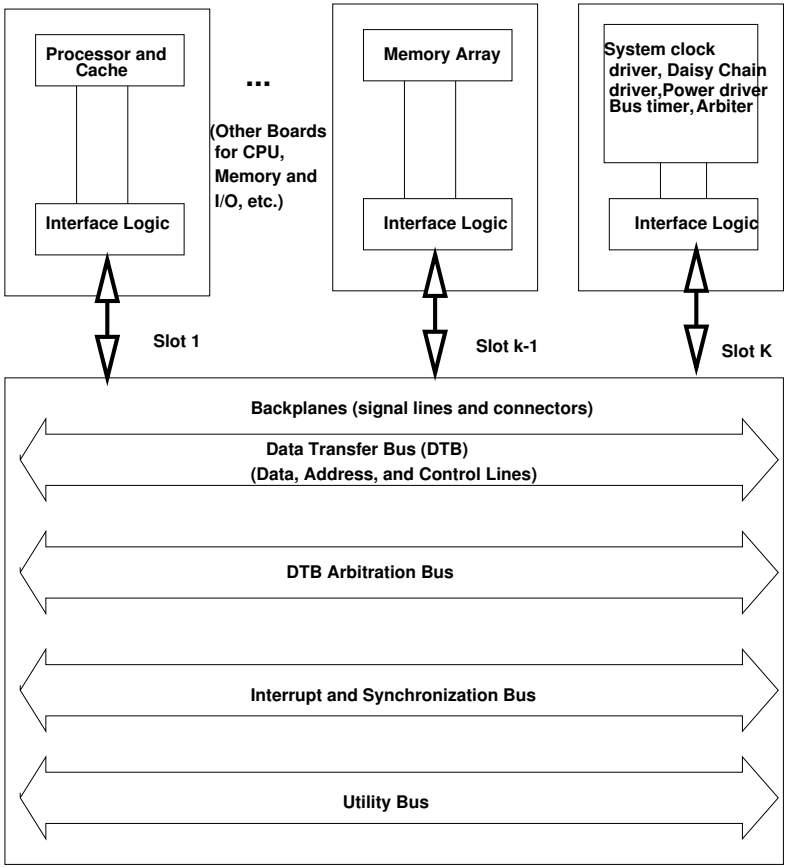
Interconnects: Choices

- Bus systems - dynamic
- Crossbars - dynamic
- Multistage networks - dynamic
- Multidimensional meshes - static

Interconnects: Bus Systems

- Data transfer bus - data, address and control lines
- Arbitration bus - who uses the data transfer bus in case of conflicts
- Interrupts and sychronization bus
- Utility bus - system clock, power up/down control, etc.
- Functional Boards - specialized for various tasks such as processing, memory, i/o, bus control, etc.

Interconnects: Bus Systems

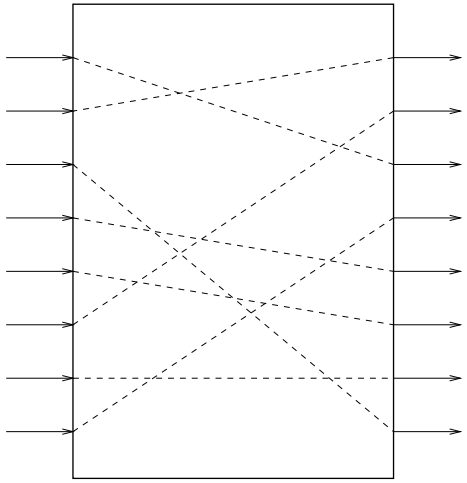


Interconnects: Bus Systems

- Only one pair of processors or processor and memory can exchange data at a time
- Cache coherence using snoopy protocols - modules watch bus transactions and take appropriate action
- Bus systems are usually low cost:
  - Bus standards such as VME Bus, GGBA, PCI, SOME, Multi Bus and Future Bus
  - Commodity parts for standard bus systems are low cost
- Bus systems are not scalable:
  - Bus bandwidths is shared among member boards
  - Physical limitations make packaging a large number of boards difficult

Interconnects: Crossbars

- Designed to increase the traffic between processors and memory
- Given N processors and an N way interleaved memory, an $N \times N$ crossbar provides N simultaneous connections
- Can control crossbar settings to achieve any desired permutation



An 8 x 8 crossbar switch
(one out of 8! = 40320 permutations shown)

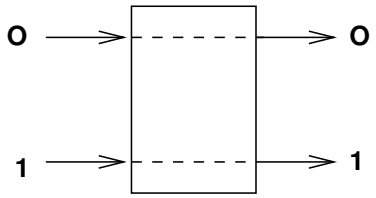
Interconnects: Crossbars

- All processors cannot access same module simultaneously
- Cache coherence using directory based schemes
- Crossbars are very expensive to build:
  - $N \times N$ requires N^2 components
- Crossbars are not scalable:
  - Infeasible for large N

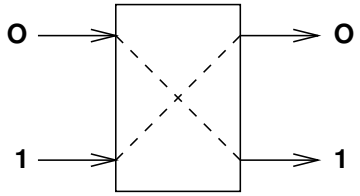
Interconnects: Multistage Network Systems

- Crossbar switches use N^2 switches for N processors
- Multistage networks use $Nlog(N)$ switches
- Use multiple stages of switches to build interconnects
- Design choices in multistage networks:
  - Basic switch type and size
  - Number of stages
  - Connections between stages
- The switch shown is used in Omega networks

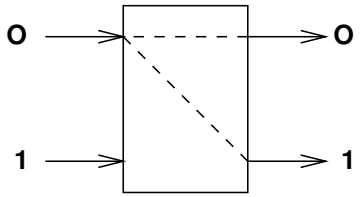
Omega Network Switch



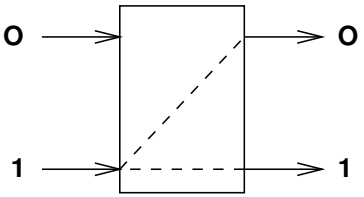
(a) Straight



(b) Crossover

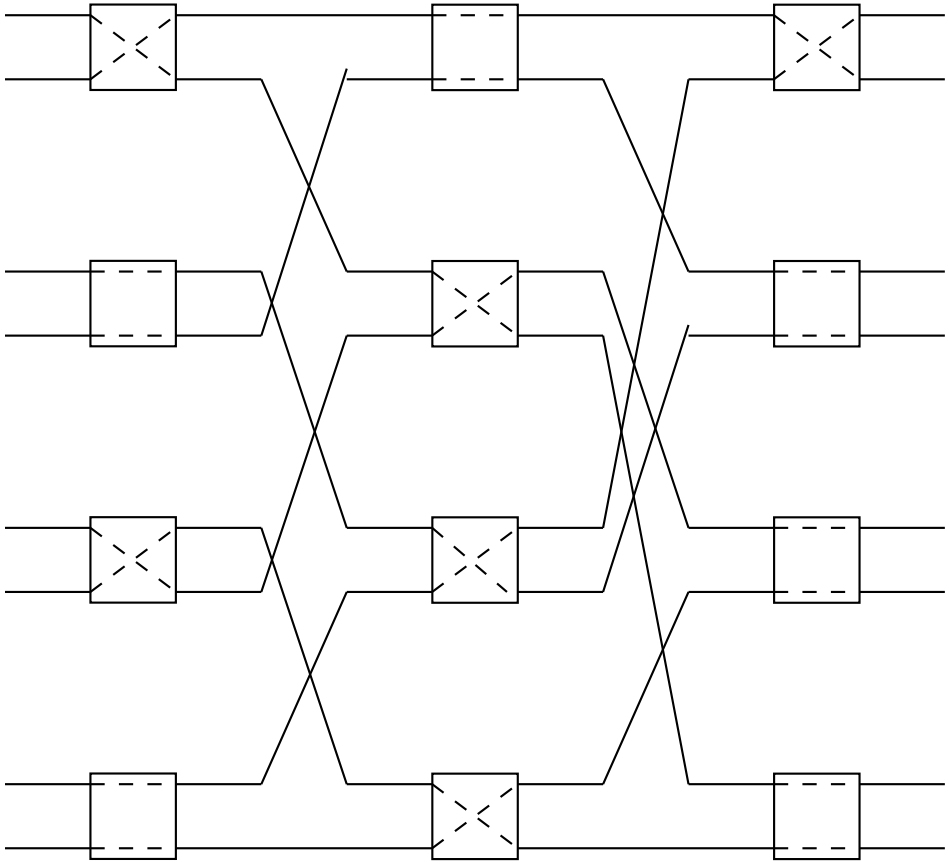


(c) Upper Broadcast



(d) Lower Broadcast

Interconnects: Omega Network



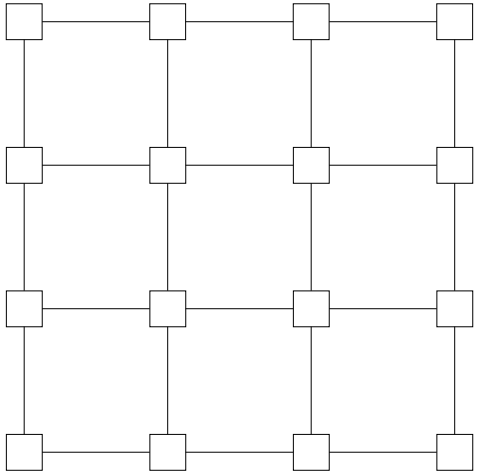
Interconnects: Multistage Network Systems

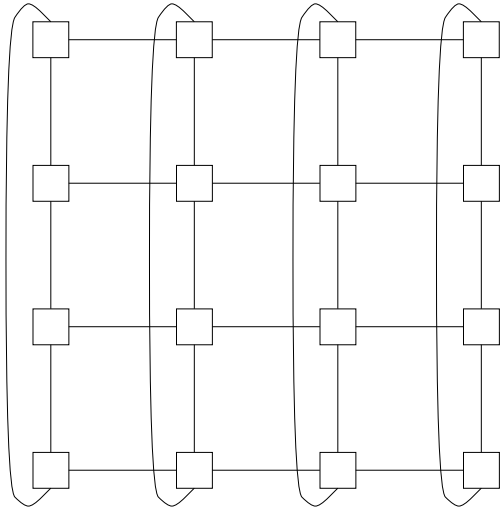
- Processors can connect to multiple memory modules simultaneously
- Cache coherence using directory based schemes
- Multistage networks are medium cost systems:
  - Require low complexity custom components
- Multistage networks are not very scalable:
  - Can add more switches easily, but need to completely rewire
  - May need to change switch control logic also

Interconnects: Multidimensional Meshes

- Each processor in an d dimensional mesh is connected to $2d$ other processors
- Data between processors routed via intermediate processors (hops)
- In practice, only 2 or 3 dimensional meshes are constructed
- Mesh with wrap around - torus

Interconnects: Multidimensional Meshes





Interconnects: Multidimensional Meshes

- If a d dimensional mesh has size N in each dimension:
  - Worst case hops for data exchange is dN
  - For a torus the worst case is $\frac{dN}{2}$
- Cache coherence using directory based schemes
- Meshes are medium cost systems:
  - Require low complexity custom components
- Meshes are very scalable:
  - Can add more processors easily without any rewiring

Effects of Interconnects on Programming

- Congestion
  - Traffic patterns on the network can create hot spots
- Latency hiding
  - Do some other work while a data transfer is in progress
- Latency amortization
  - Try to fetch large amounts of data at the same time

Case Studies

- SGI Altix 3700 - network
- SGI Origin 2000 - hypercube
- Sun Enterprise 450 Server - bus
- Sun Enterprise 10000 Server - crossbar
- HP K580 Server - bus
- HP V2500 Server - cross-bar
- Dell Poweredge 6800 enterprise server - bus based
- NEC SX6+ - crossbar
- IBM Blue Gene - networks, torus

SGI Origin 2000

- Distributed shared memory multiprocessor, scalable from 1 to 128 processors, upto 256 GB of memory
- Consists of a scalable interconnection (hypercube) of node boards that each have:
- Two MIPS R10000 processors, One network hub, Some main memory, and a directory
- Directory based hardware cache coherence

SUN Enterprise 450 Server

- Consists of four SUN ULtraSPARC-II 400 MHz processors
- Upto 4 GB of main memory
- Interconnect: bus-based 1.6 GB/s UPA interconnect

SUN Enterprise 10000 Server (Starfire)

- Between 4 to 64 processors, each processor is a 400 MHz Superscalar Ultra-SPARC processor (64 bit Ultra port architecture)
- Each processor has a primary cache of 16 KB instruction cache, 16 KB data cache, Secondary cache of 4MB.
- Main memory can be 2 GB to 64 GB
- Maximum of 16 boards in system, can be memory boards or CPU-memory boards. A CPU-memory board contains two CPUs and 16 memory SIMMs.
- System Interconnect is a Gigaplane-XB crossbar at 12.8 GB/s bandwidth with 500 Nano-second latency.
- Runs Solaris 2.6 operating systemwith symmetric multiprocessing, can be used for number crunching and database applications
- Performance with 64 processors, 11908 SPECfprate95, 9181 SPECintrate95

HP 9000 K580 Server

- Between 1 to 6 processors
- Each processor is a 240 MHz HP PA-8200 CPU with 2 MB instruction cache 2 MB data cache
- Main memory 256 MB to 8.192 GB
- Interconnection network is bus based

HP 9000 V2500 Server

- Consists of 2 to 32 processors of symmetric multiprocessing
- Each processsor consists of a 440 MHz PA8500 4-way superscalar 64-bit CPUs
- Between 1 to 32 GB of main memory, with 32 way interleaving
- Consists of Hyperplane 8 X 8 cross-bar interconnect provides bandwidth of 15.4 GB/sec, with bidirectional 960 MB/sec per port
- Runs HPUX version 11.0 symmetric multiprocessing
- These clusters can be scaled up to 128 processor using HP's Scalable Computing Architecture with 128 GB memory with 61.6 GB/sec bandwidth (needs four cabinets and SCA hyperlink)

Dell Poweredge 6800 Enterprise Server

- Consists of upto 4 Intel Pentium III Xeon 3.6 GHz processors with 8 MB L3 cache
- Consists of upto 32 GB main memory, and 3.6 TB of disk
- Interconnect: bus-based
- Runs Windows NT and Linux operating systems

IBM Blue Gene/L

- Consists of 65,536 PowerPC 440 processor with three levels of caches
- 2 processors on a card (1 compute and 1 communication processor), 16 cards per node, 16 boards per "plane"
- 64x32x32 torus
- Directory-based cache coherence
- 360 TFLOPS

Summary

- Overview of shared memory architectures
- Memory organizations
- Interconnect organizations
- Case studies
- NEXT LECTURE: Shared Memory Parallel Programming - I
- READING: B. Bauer, "Practical Parallel Programming", Academic Press, Chapter 3