Electrical Engineering and Computer Science

EECS 358 - INTRODUCTION TO PARALLEL COMPUTING

Lecture 8

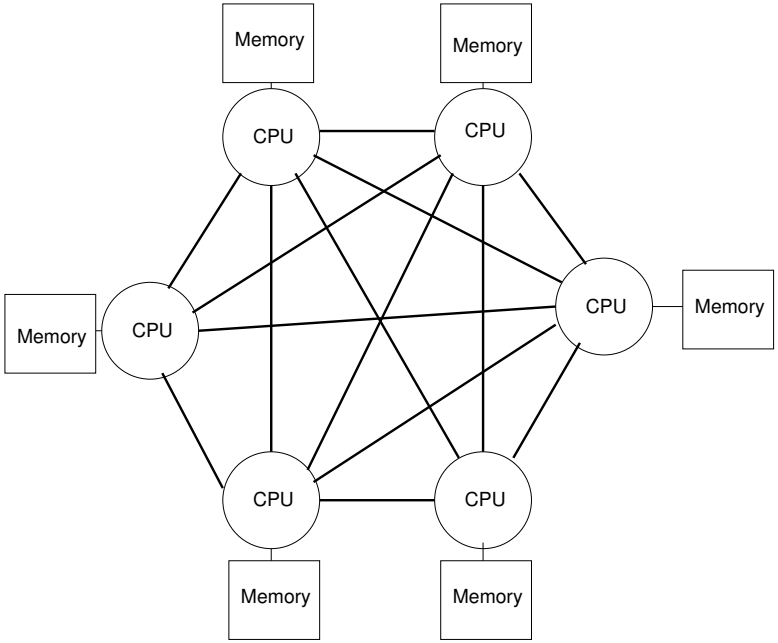
Distributed Memory Parallel Architectures

Outline

- Overview of distributed memory parallel architectures
- Interconnection networks
- Message passing issues
- Case Studies
  - Intel Paragon
  - IBM SP-2
- READING: Chapter 6

Distributed Address Space Machines

- Each processor has its own local address space
- Processors share data via explicit message passing



Distributed Address Space Machines

- Important aspects are:
  - Interconnect
  - Message Routing Mechanism
- These aspects determine:
  - Performance - programming techniques
  - Scalability
  - Cost

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: Evaluation

- Diameter: maximum distance between any two processors. The distance is the least number of links (hops) that need to be traversed between processors

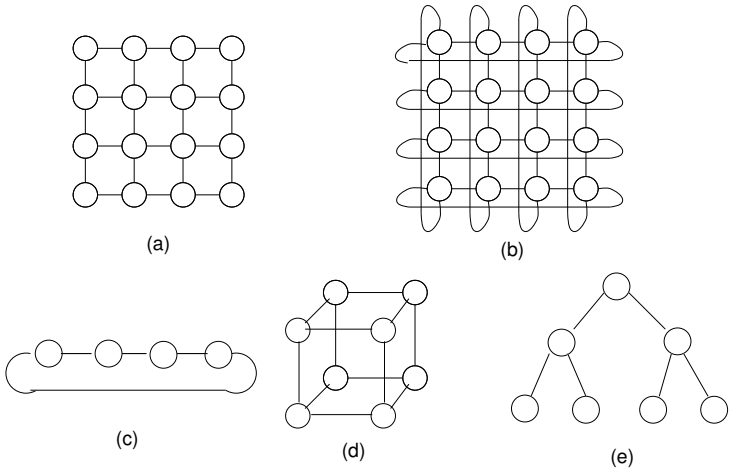
- Arc connectivity: connectivity is a measure of the multiplicity of paths between processors. Arc connectivity is the minimum number of links that need to be removed in order to break the network into two disjoint parts; it is a measure of connectivity. High connectivity (and thus arc connectivity) is desirable for avoiding contention

- Bisection width: the minimum number of links that need to be removed in a network to separate the processors into two halves. Bisection width is a measure of the volume of traffic that can be handled by the network

Interconnects: Choices

- Ring - static
- Multidimensional mesh - static
- Hypercube- static
- Fat tree - static

Interconnect Examples



Interconnects: Ring

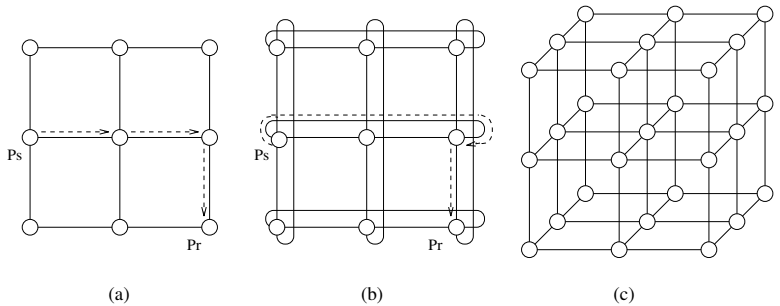
- Each processor connected to 2 other processors
- Diameter of a ring is $\lfloor \frac{p}{2} \rfloor$
- Arc connectivity of a ring is 2
- Bisection width of a ring is 2

(a) (b)

Interconnects: Multidimensional Meshes

- Each processor in an d dimensional mesh is connected to $2d$ other processors except for the corner processors
- In practice, only 2 or 3 dimensional meshes are constructed
- Mesh with wrap around - torus

Multi-dimensional Meshes



Interconnects: Multidimensional Meshes

- Diameter of a 2 dimensional mesh is $2(\sqrt{p}-1)$; diameter of a 2 dimensional torus is $2\lfloor\frac{\sqrt{p}}{2}\rfloor$
- Arc connectivity of a 2 dimensional mesh is 2 and for a torus it is 4
- Bisection width of a 2 dimensional mesh is $\sqrt{p}$; for a torus it is $2\sqrt{p}$

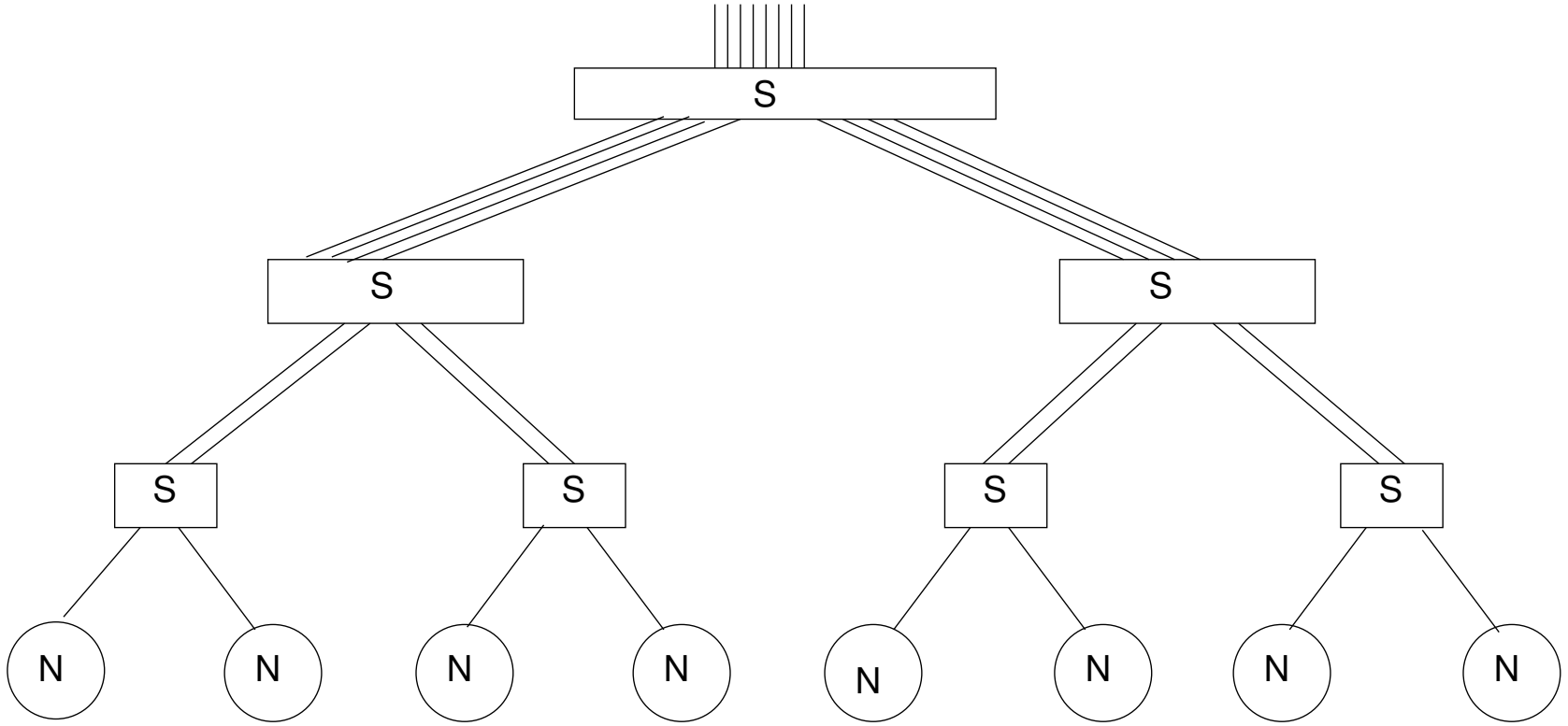
Interconnects: Fat Tree

- A tree network is one where there is exactly one path between a pair of processors
- In a simple tree, the bandwith on higher level links is shared among all processors below creating bottlenecks
- A fat tree solves the problem by using wider links at higher levels in the tree

Interconnects: Fat Tree

- Diameter of a fat tree is $2\log(p)$
- Arc connectivity is 1
- Bisection width is 1

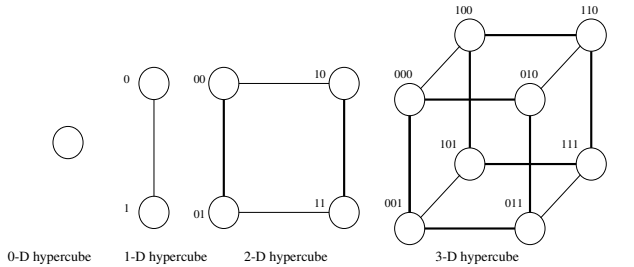
Interconnects: Fat Tree

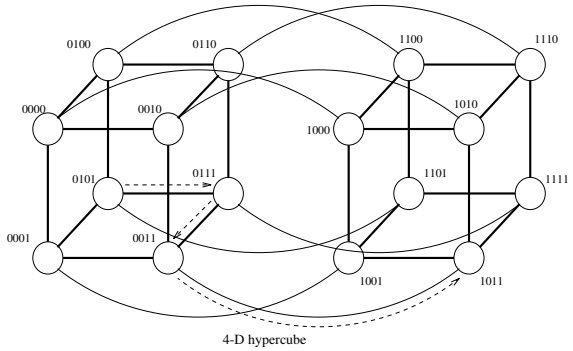


Interconnects: Hypercube

- A hypercube is a multidimensional mesh with exactly two processors in each dimension
- In a d dimensional hypercube, each processor is connected with d other processors
- Hypercubes can be constructed recursively; when two d dimensional hypercubes are used to construct a $d+1$ dimensional hypercube, the labels on the nodes of the original hypercubes are prefixed with a 0 or a 1 for the new hypercube

Interconnects: Hypercube





Interconnects: Hypercube

- Two processors are connected if their labels differ in exactly one bit position
- In a d dimensional hypercube, each processor is connected to exactly d other processors
- Consider two processors with labels s and t, the Hamming distance for the two processors is defined to be the number of positions in which their labels differ
- For example if $s = 011$ and $t = 101$, Hamming distance is 2
- The Hamming distance represents the shortest path between two processors

Interconnects: Hypercube

- Diameter of a hypercube is $\log(p)$
- Arc connectivity of a hypercube is $\log(p)$
- Bisection width of a hypercube is $\frac{p}{2}$

Interconnects: Comparison

Network	Diameter	Bisection Width	Arc Connectivity
Completely connected	1	$\frac{p^2}{4}$	$p-1$
Ring	$\lfloor \frac{p}{2} \rfloor$	2	2
Mesh	$2(\sqrt{p}-1)$	$\sqrt{p}$	2
Torus	$2\lfloor \frac{\sqrt{p}}{2} \rfloor$	$2\sqrt{p}$	4
Binary Tree	$2\log(p)$	1	1
Hypercube	$\log(p)$	$\frac{p}{2}$	$\log(p)$

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

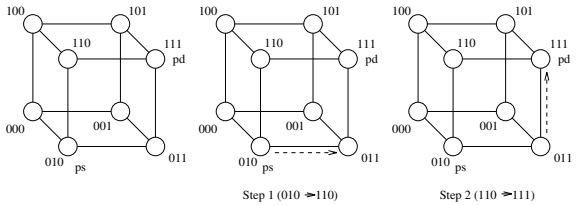
Message Routing Mechanisms

- Routing mechanism determines the path a message takes through the network when going from a source to a destination processor
- Minimal or non-minimal routing:
  - Minimal always takes the shortest path
  - Non-minimal may sometimes take a longer path to avoid congestion
- Deterministic or adaptive:
  - Deterministic routine always takes the same path between processors
  - Adaptive routing can take different paths depending on congestion

Example: E-cube Routing in Hypercubes

- If the message is at processor P_i and needs to go to P_d, compute $s = P_i \oplus P_d$. Send message along dimension k from P_i, where k is the least significant non-zero bit in s

- Continue this process at each successive processor until P_d is reached



Example: XY Routing in 2d Mesh

- Message sent along X dimension till destination's X coordinate is reached
- Now message is sent along Y dimension till it reaches destination processor
- Scheme is minimal and deterministic

Message Transfer Mechanisms

- Messages typically consist of:
  - A header which contains information about the destination
  - The data that needs to be transmitted
  - A trailer which signals the end of a message
- Message transfer mechanism determines how message data is actually transferred across network links in the chosen message route
- Three components to message transfer cost:
  - Startup time (t_s) - cost of handling message at sending processor
  - Per-hop time (t_h) - it is the time taken by the header to traverse a link
  - Per-word transfer time (t_w) - time taken for a word to traverse a link

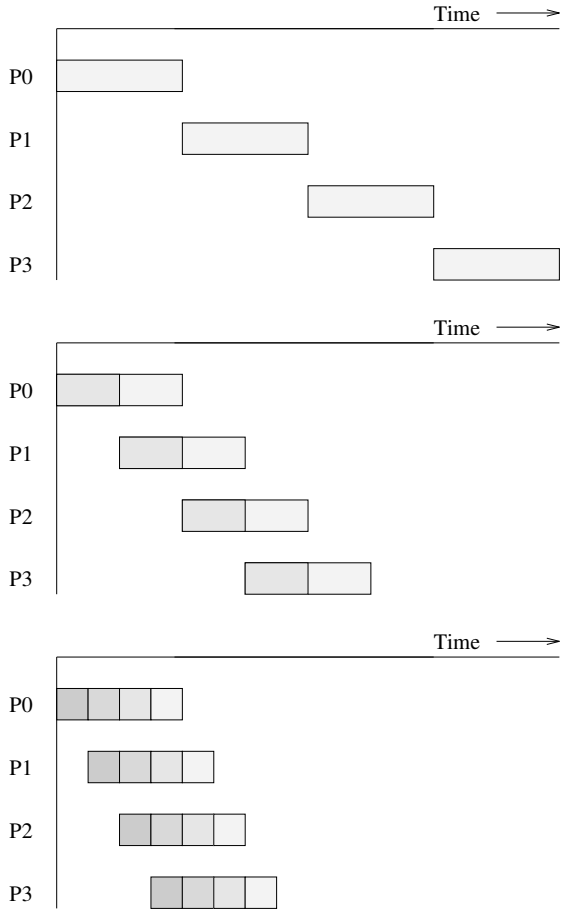
Message Transfer Mechanisms

- Store-and-forward routing: the message is transferred completely from link to link along the route between processors; it is buffered at intermediate processors

- Cut-through routing: the message is transmitted on the outgoing link as soon as it is received on the incoming link at intermediate processors, thus removing the need for buffering

- Wormhole routing: This is a popular type of cut-through routing. The message is chopped into small units called flow-control digits or flits. Flits are transmitted in a pipelined fashion between processors along the message route

Message Transfer Mechanisms



(a) A single message sent over a store-and-forward network

(b) The same message broken into two parts and sent over the network.

(c) The same message broken into four parts and sent over the network.

Message Transfer Mechanisms

- Cost of message transfer for a message length of m words along a route of l links:
  - Store-and-forward routing:
    $t_{comm} = t_s + (mt_w + t_h)l \approx t_s + mt_w l$
  - Wormhole routing:
    $t_{comm} = t_s + lt_h + mt_w$

Example Distributed Memory Parallel Machines

- Intel iPSC/860 (Hypercube)
- NCUBE/2 (Hypercube)
- Meiko (Mesh)
- Intel Paragon (2-dim Mesh)
- IBM SP-2 (Multistage Network, 8X8 switches)
- Thinking Machines CM-5 (Fat Tree)
- INMOS Transputers (Any topology, degree 4)

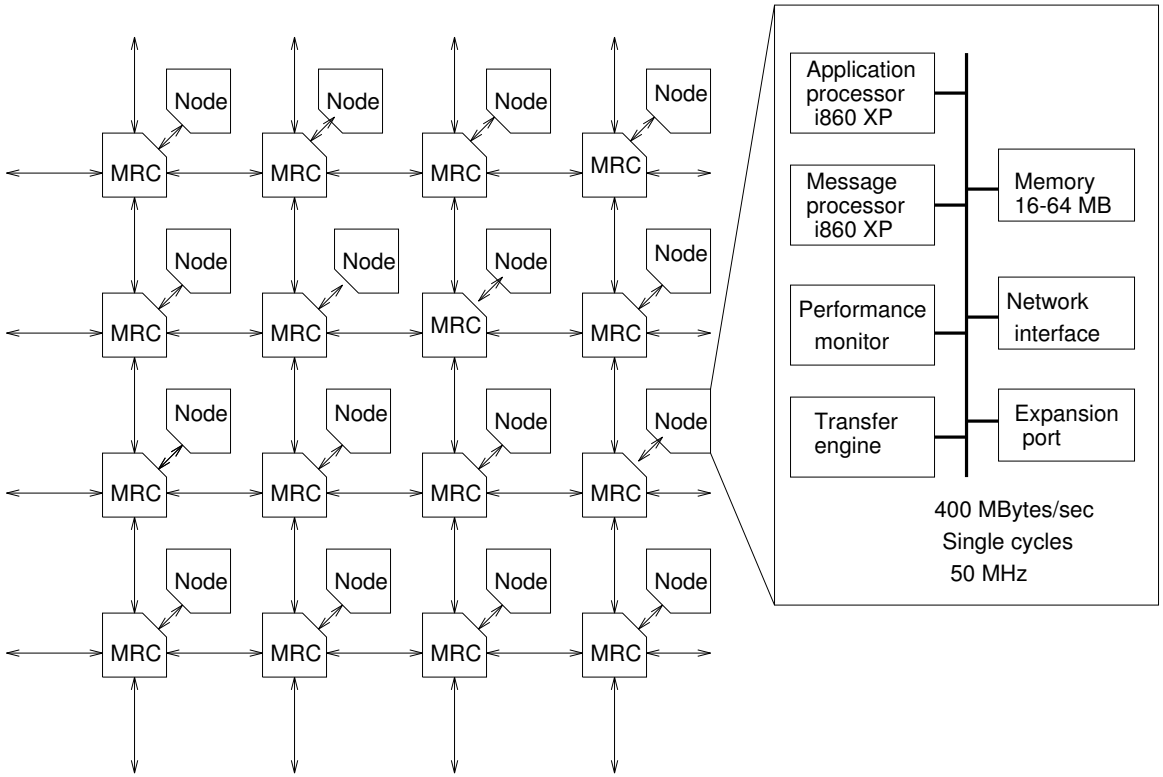
Intel Paragon

- The Intel Paragon is a distributed memory message-passing MIMD multicomputer, which consists of 1024 processors connected as a two-dimensional toroidal mesh
- Each node in the mesh has up to four application processors and one message processor.
- Each processor is a 50 MHz Intel i860 XP^{TM} processor, rated at 75 MFLOPS.
- The total system performance is rated at 300 GFLOPS of 64-bit floating point computations.
- Each processor has between 16 and 128 Mbytes of memory, but sees a separate virtual address space and runs the OSF/1 operating system.

Intel Paragon

- The total system memory is 128 Gbytes.
- Different processors communicate via message passing using the same iPSC send and receive calls as the Intel iPSC hypercubes.
- The network is a two-dimensional toroidal mesh with a total bandwidth of 500 Gbytes/s, and uses a fast mesh routing chip (MRC) at each node of the mesh to accomplish the routing.

Organization of the Intel Paragon



IBM SP-2

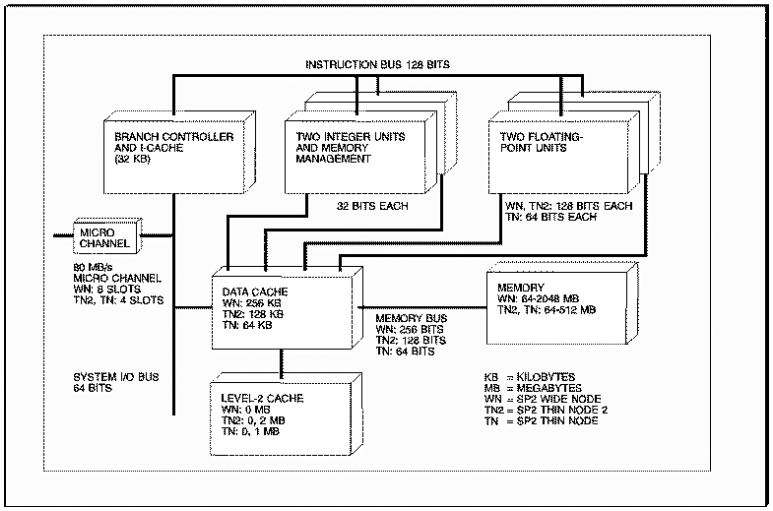
- IBM SP-2 is a general-purpose scalable parallel system based on distributed memory message passing architecture.
- SP2 systems range from 2 to 512 nodes (or processing elements)
- Each node is a POWER2 technology RISC/6000 processor
- Interconnected by high-performance, multi-stage, packet-switched network for interprocessor communication.
- Each node contains copy of standard AIX operating system
- Supports both message-passing and data-parallel programming models

Organization of the SP-2 Nodes

- Consists of wide nodes and thin nodes
- Both processors have two fixed-point units, two floating-point units (each capable of a multiply and add each cycle), and a instruction and branch control unit.
- Peak processing 267 MFLOPS
- Wide node memory is 2 GBytes, and bandwidth of 2.1 GB/sec
- Thin nodes have less memory and less bandwidth

Organization of SP-2 node

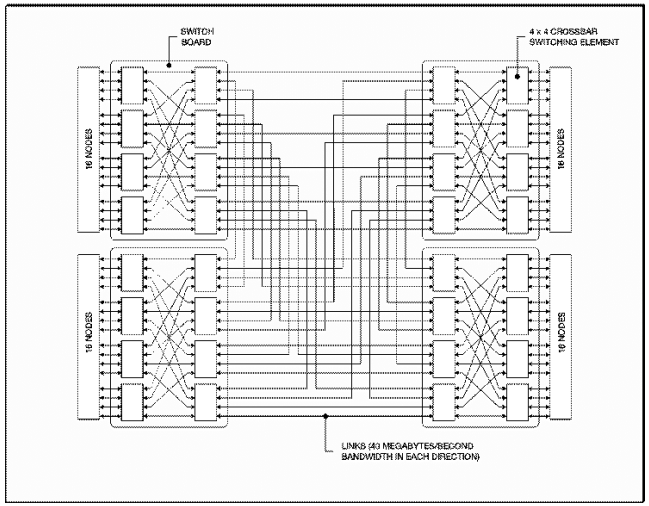
Figure 9 The SP2 processor node structure for thin and wide nodes



Organization of IBM SP-2 Switch

- Each switch is a 4 X 4 bidirectional cross-bar switch.

Figure 10 SP2 Switch structure for a 64-node system



Summary

- Overview of distributed memory parallel architectures
- Interconnection networks
- Message passing issues
- Case Studies
- NEXT LECTURE: Distributed Memory Message Communication Issues
- READING: Chapter 6