ECE574 Cluster Computing

Dichotomy of Parallel Computing Platforms (Continued)

Lecturer: Dr. Yifeng Zhu

Spring, 2008

Outline

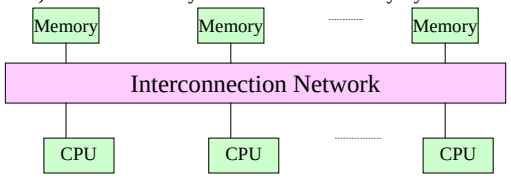
- **Class Review**
- **Network Interconnections**
  - **Crossbar**
    - » **Example: myrinet**
  - **Multistage Network**
    - » **Example: Omega network**
- **Flynn's Classifications (1972)**
  - **SIMD (Single Instruction Stream, Multiple Data Streams):**
    - » a computer which exploits multiple data streams against a single instruction stream to perform operations which may be naturally parallelized. For example, an array processor.
  - **MIMD (Multiple Instruction Streams, Multiple Data Streams)**
    - » multiple autonomous processors simultaneously executing different instructions on different data. Distributed systems are generally recognised to be MIMD architectures; either exploiting a single shared memory space or a distributed memory space.

References:
• *A Survey of Parallel Computer Architectures*, Duncan, Ralph, IEEE Computer. February 1990, pp. 5-16.
• Flynn, M., *Some Computer Organizations and Their Effectiveness*, IEEE Trans. Comput., Vol. C-21, pp. 94, 1972.

Shared Memory

- **One or more memories**
- **Global address space (all system memory visible to all processors)**
- **Transfer of data between processors is usually implicit, just read (write) to (from) a given address (OpenMP)**
- **Cache-coherency protocol to maintain consistency between processors.**

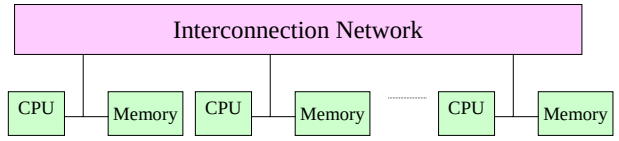
(UMA) Uniform-memory-access Shared-memory System



Distributed Shared Memory

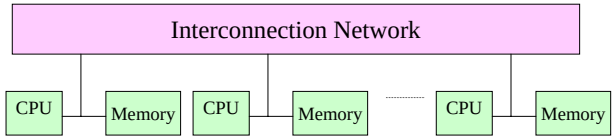
- **Single address space with implicit communication**
- **Hardware support for read/write to non-local memories, cache coherency**
- **Latency for a memory operation is greater when accessing non local data than when accessing date within a CPU's own memory**

(NUMA)Non-Uniform-memory-access Shared-memory System



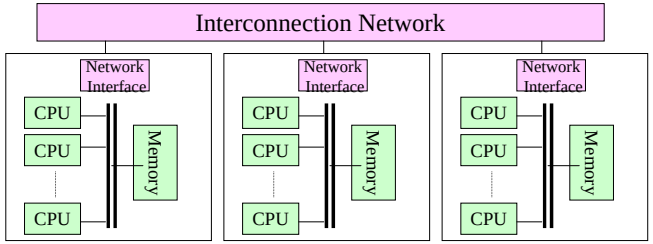
Distributed Memory

- Each processor has access to its own memory only
- Data transfer between processors is explicit, user calls message passing functions
- Common Libraries for message passing
  - MPI, PVM
- User has complete control/responsibility for data placement and management



Hybrid Systems

- Distributed memory system with multiprocessor shared memory nodes.
- Most common architecture for current generation of parallel machines

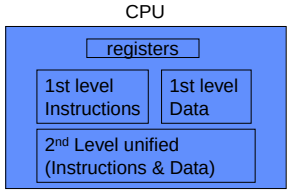


Message Passing vs. Shared Memory Platforms

- Message passing platforms require little hardware support, other than a network.
- Shared memory platforms can easily emulate message passing. The reverse is more difficult to do (in an efficient manner).

Processors and the Memory Hierarchy

- Registers (1 clock cycle, 100s of bytes)
- 1st level cache (3-5 clock cycles, 100s KBytes)
- 2nd level cache (~10 clock cycles, MBytes)
- Main memory (~100 clock cycles, GBytes)
- Disk (milliseconds, 100GB to gianormous)



IBM Dual Core

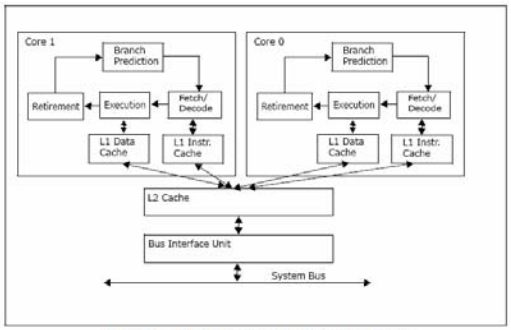
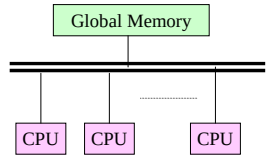


Figure 2-3. Intel Advanced Smart Cache Architecture

From Intel® 64 and IA-32 Architectures Optimization Reference Manual
http://www.intel.com/design/processor/manuals/248966.pdf

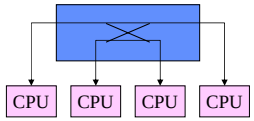
Interconnection Network Topologies - Bus

- **Bus**
  - **A single shared data path**
  - **Pros**
    - » **Simplicity**
      - **cache coherence**
      - **synchronization**
  - **Cons**
    - » **fixed bandwidth**
      - **Does not scale well**



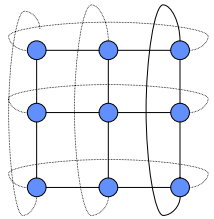
Interconnection Network Topologies – Switch based

- **Switch Based**
  - **mxn switches**
  - **Many possible topologies**
- **Characterized by**
  - **Diameter**
    - » **Worst case number of links between two processors (The maximum shortest path between two processors)**
    - » **Impacts latency**
  - **Bisection width**
    - » **Minimum number of connections that must be removed to split the network into two parts with equal size**
    - » **Communication bandwidth limitation**
  - **Edges per switch**
    - » **Best if this is independent of the size of the network**



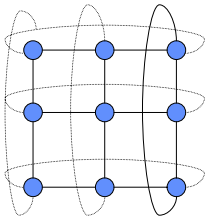
Interconnection Network Topologies - Mesh

- **2-D Mesh**
  - **2-D array of processors**
- **Torus/Wraparound Mesh**
  - **Processors on edge of mesh are connected**
- **Characteristics (n nodes)**
  - **Diameter = or**
  - **Bisection width = or**
  - **Switch size =**
  - **Number of switches =**



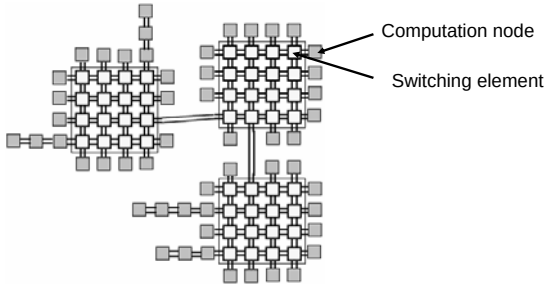
Interconnection Network Topologies - Mesh

- **2-D Mesh**
  - 2-D array of processors
- **Torus/Wraparound Mesh**
  - Processors on edge of mesh are connected
- **Characteristics (n nodes)**
  - Diameter = $\sqrt{n}$ (for wraparound) or $2(\sqrt{n}-1)$ (for mesh)
  - Bisection width = $2\sqrt{n}$ (for wraparound) or $\sqrt{n}$ (for mesh)
  - Switch size = 4
  - Number of switches = n



One-Dimensional Array

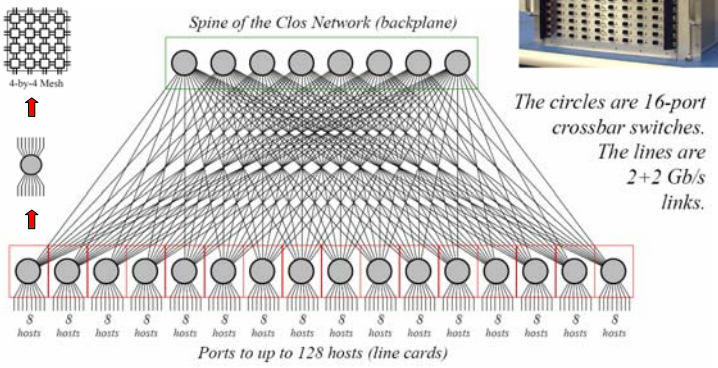
An Example to improve cost scalability



Example: One-dimensional Array

Myrinet: Flat Tree Topology

Another Example to improve cost scalability

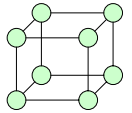


Myrinet-2000, 128 Host "Network in a Box"

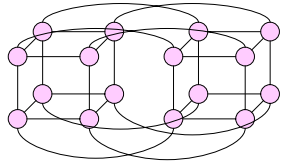
Interconnection Network Topologies - Hypercube

- **Hypercube**
  - A d-dimensional hypercube has $n=2^d$ processors.
  - Each processor directly connected to d other processors
  - Shortest path between a pair of processors is at most d
- **Characteristics ($n=2^d$ nodes)**
  - Diameter =
  - Bisection width =
  - Switch size =
  - Number of switches =

3-D Hypercube



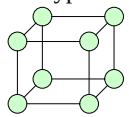
4-D Hypercube



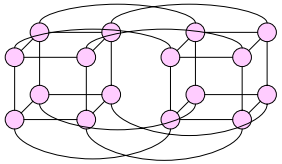
Interconnection Network Topologies - Hypercube

- **Hypercube**
  - A d-dimensional hypercube has $n=2^d$ processors.
  - Each processor directly connected to d other processors
  - Shortest path between a pair of processors is at most d
- **Characteristics ($n=2^d$ nodes)**
  - Diameter = d
  - Bisection width = $n/2$
  - Switch size = d
  - Number of switches = n

3-D Hypercube

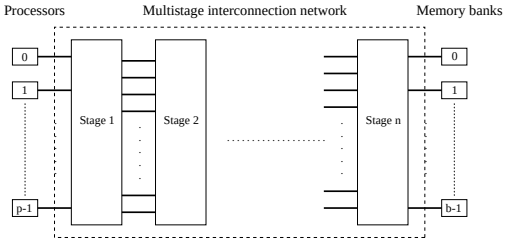


4-D Hypercube



Multistage Interconnection Network

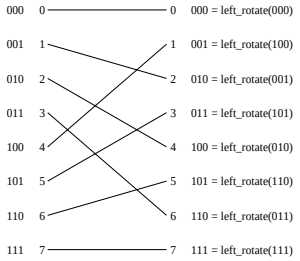
- **Can reduce switch requirements**
  - at cost of additional series switch latency
- **Example: Omega Network**
  - This network consists of *log p* stages, where *p* is the number of inputs/outputs.
  - O(p *log*(p)) switches, O(*log*(p)) delay



Schematic of a typical multistage interconnection network.

Omega Network: Perfect shuffle

Each stage of the Omega network implements a perfect shuffle by "rotate left" as follows:



A perfect shuffle interconnection for eight inputs and outputs.

8 x 8 OMEGA NETWORK