Parallel Processing

2009 가을학기

Prof. Heon Y. Yeom
Seoul National University
Parallel Architecture
Flynn’s Taxonomy

- Instruction stream
- Data stream
- Single vs. multiple
- Four combinations
  - SISD
  - SIMD
  - MISD
  - MIMD
SISD

- Single Instruction, Single Data
- Single-CPU systems
- Note: co-processors don’t count
  - Functional
  - I/O
- Example: Older computers, PCs
SIMD

- Single Instruction, Multiple Data
- Two architectures fit this category
  - Pipelined vector processor (Cray-1)
  - Processor array
    (Connection Machine)
MISED

- A single data stream is fed into multiple processing units.
- Each processing unit operates on the data independently via independent instruction streams.
- Some conceivable uses might be:
  - A multiple frequency filters operating on a single signal stream
  - multiple cryptography algorithms attempting to crack a single coded message.
MIMD

- Currently, the most common type of parallel computer. Most modern computers fall into this category.
- Multiple Instruction: every processor may be executing a different instruction stream
- Multiple Data: every processor may be working with a different data stream
- Execution can be synchronous or asynchronous, deterministic or non-deterministic
- Examples: most current supercomputers, networked parallel computer clusters and "grids", multi-processor SMP computers, multi-core PCs.
- Note: many MIMD architectures also include SIMD execution sub-components
Parallel Computer Memory Architecture

- **Shared Memory**
  - Uniform memory access
  - Non-uniform memory access
- **Distributed memory**
- **Hybrid Distributed-Shared memory**
Shared Memory (Uniform Memory Access)

- Most commonly represented today by Symmetric Multiprocessor (SMP) machines
- Identical processors
- Equal access and access times to memory
- Sometimes called CC-UMA. Cache coherent means if one processor updates a location in shared memory, all the other processors know about the update. (accomplished at the hardware level)
(Non-Uniform Memory Access)

- Often made by physically linking 2+ SMPs
- One SMP can directly access memory of another SMP
- Not all processors have equal access time to all memories
- Memory access across link is slower
- If cache coherency is maintained, then may also be called CC-NUMA - Cache Coherent NUMA
Advantages:

- Memory is scalable with number of processors.
- Each processor can rapidly access its own memory without cache coherency.
- Cost effectiveness: can use commodity, off-the-shelf processors and networking.

Disadvantages:

- The programmer is responsible for many of the details associated with data communication between processors.
- It may be difficult to map existing data structures.
- Non-uniform memory access (NUMA) times.
Hybrid Memory

- The largest and fastest computers in the world today employ both shared and distributed memory architectures.
- The shared memory component is usually a cache coherent SMP machine.
- The distributed memory component is the networking of multiple SMPs. Network communications are required to move data from one SMP to another.
- Current trends seem to indicate that this type of memory architecture will continue to prevail and increase at the high end of computing for the foreseeable future.
**Interconnection Networks**

- **Uses of interconnection networks**
  - Connect processors to shared memory
  - Connect processors to each other

- **Interconnection media types**
  - Shared medium
  - Switched medium
Shared versus Switched Media

(a) Shared Medium

(b) Switched Medium

Processors
Shared Medium

- Allows only 1 message at a time
- Messages are broadcast
- Each processor “listens” to every message
- Arbitration is decentralized
- Collisions require resending of messages
- Ethernet is an example
Switched Medium

- Supports point-to-point messages between pairs of processors
- Each processor has its own path to switch
- Advantages over shared media
  - Allows multiple messages to be sent simultaneously
  - Allows scaling of network to accommodate increase in processors
Switch Network Topologies

- **View switched network as a graph**
  - Vertices = processors or switches
  - Edges = communication paths

- **Two kinds of topologies**
  - Direct
  - Indirect
Direct Topology

- Ratio of switch nodes to processor nodes is 1:1
- Every switch node is connected to
  - 1 processor node
  - At least 1 other switch node
Indirect Topology

- Ratio of switch nodes to processor nodes is greater than 1:1
- Some switches simply connect other switches
Evaluating Switch Topologies

- Diameter
- Bisection width
- Number of edges / node
- Constant edge length? (yes/no)
2-D Mesh Network

- Direct topology
- Switches arranged into a 2-D lattice
- Communication allowed only between neighboring switches
- Variants allow wraparound connections between switches on edge of mesh
2-D Meshes

(a)

(b)
Evaluating 2-D Meshes

- Diameter: $\Theta(n^{1/2})$
- Bisection width: $\Theta(n^{1/2})$
- Number of edges per switch: 4
- Constant edge length? Yes
Binary Tree Network

- Indirect topology
- $n = 2^d$ processor nodes, $n-1$ switches
Evaluating Binary Tree Network

- Diameter: $2 \log n$
- Bisection width: 1
- Edges / node: 3
- Constant edge length? No
Hypertree Network

- Indirect topology
- Shares low diameter of binary tree
- Greatly improves bisection width
- From "front" looks like $k$-ary tree of height $d$
- From "side" looks like upside down binary tree of height $d$
Hypertree Network

(a)

(b)

(c)
Evaluating 4-ary Hypertree

- Diameter: $\log n$
- Bisection width: $n / 2$
- Edges / node: 6
- Constant edge length? No
Butterfly Network

- Indirect topology
- $n = 2^d$ processor nodes connected by $n(\log n + 1)$ switching nodes
Butterfly Network Routing
Evaluating Butterfly Network

- Diameter: \( \log n \)
- Bisection width: \( n / 2 \)
- Edges per node: 4
- Constant edge length? No
Hypercube

- Directory topology
- $2 \times 2 \times \ldots \times 2$ mesh
- Number of nodes a power of 2
- Node addresses $0, 1, \ldots, 2^k-1$
- Node $i$ connected to $k$ nodes whose addresses differ from $i$ in exactly one bit position
Hypercube Addressing

The diagram illustrates the hypercube addressing scheme with binary addresses ranging from 0000 to 1111. Each node represents a binary address, and the connections between nodes demonstrate the hypercube topology, where each dimension splits the address space into two parts, forming a cube-like structure.
Hypercubes Illustrated
Evaluating Hypercube Network

- Diameter: \( \log n \)
- Bisection width: \( n / 2 \)
- Edges per node: \( \log n \)
- Constant edge length? No
Shuffle-exchange

- Direct topology
- Number of nodes a power of 2
- Nodes have addresses 0, 1, ..., $2^k-1$
- Two outgoing links from node $i$
  - Shuffle link to node $LeftCycle(i)$
  - Exchange link to node $[\text{xor} (i, 1)]$
Shuffle-exchange Illustrated
Shuffle-exchange Addressing
Evaluating Shuffle-exchange

- Diameter: \(2\log n - 1\)
- Bisection width: \(\approx n / \log n\)
- Edges per node: 2
- Constant edge length? No
Comparing Networks

- All have logarithmic diameter except 2-D mesh
- Hypertree, butterfly, and hypercube have bisection width $n/2$
- All have constant edges per node except hypercube
- Only 2-D mesh keeps edge lengths constant as network size increases
Vector Computers

- Vector computer: instruction set includes operations on vectors as well as scalars
- Two ways to implement vector computers
  - Pipelined vector processor: streams data through pipelined arithmetic units
  - Processor array: many identical, synchronized arithmetic processing elements
Why Processor Arrays?

- Historically, high cost of a control unit
- Scientific applications have data parallelism
Processor Array

- **Front end computer**
  - Program
  - Data manipulated sequentially

- **Processor array**
  - Data manipulated in parallel
Multiprocessors

- Multiprocessor: multiple-CPU computer with a shared memory
- Same address on two different CPUs refers to the same memory location
- Avoid three problems of processor arrays
  - Can be built from commodity CPUs
  - Naturally support multiple users
  - Maintain efficiency in conditional code
Centralized Multiprocessor

- Straightforward extension of uniprocessor
- Add CPUs to bus
- All processors share same primary memory
- Memory access time same for all CPUs
  - Uniform memory access (UMA) multiprocessor
  - Symmetrical multiprocessor (SMP)
Centralized Multiprocessor
Private and Shared Data

- **Private data**: items used only by a single processor
- **Shared data**: values used by multiple processors
- In a multiprocessor, processors communicate via shared data values
Problems Associated with Shared Data

- **Cache coherence**
  - Replicating data across multiple caches reduces contention
  - How to ensure different processors have same value for same address?

- **Synchronization**
  - Mutual exclusion
  - Barrier
Cache-coherence Problem

![Diagram of cache-coherence problem](image)
Distributed Multiprocessor

- Distribute primary memory among processors
- Increase aggregate memory bandwidth and lower average memory access time
- Allow greater number of processors
- Also called non-uniform memory access (NUMA) multiprocessor
Distributed Multiprocessor
Cache Coherence

- Some NUMA multiprocessors do not support it in hardware
  - Only instructions, private data in cache
  - Large memory access time variance
- Implementation more difficult
  - No shared memory bus to “snoop”
  - Directory-based protocol needed
Directory-based Protocol

- Distributed directory contains information about cacheable memory blocks
- One directory entry for each cache block
- Each entry has
  - Sharing status
  - Which processors have copies
Sharing Status

- **Uncached**
  - Block not in any processor’s cache

- **Shared**
  - Cached by one or more processors
  - Read only

- **Exclusive**
  - Cached by exactly one processor
  - Processor has written block
  - Copy in memory is obsolete
Directory-based Protocol

Interconnection Network

- Directory
  - Local Memory
    - Cache
      - CPU 0
  - Directory
    - Local Memory
      - Cache
        - CPU 1
  - Directory
    - Local Memory
      - Cache
        - CPU 2
Multicomputer

- Distributed memory multiple-CPU computer
- Same address on different processors refers to different physical memory locations
- Processors interact through message passing
- Commercial multicomputers
- Commodity clusters
Commodity Cluster

- Co-located computers
- Dedicated to running parallel jobs
- No keyboards or displays
- Identical operating system
- Identical local disk images
- Administered as an entity
Beowulf Concept

- NASA (Sterling and Becker)
- Commodity processors
- Commodity interconnect
- Linux operating system
- Message Passing Interface (MPI) library
- High performance/$ for certain applications
Network of Workstations

- Dispersed computers
- First priority: person at keyboard
- Parallel jobs run in background
- Different operating systems
- Different local images
- Checkpointing and restarting important
<table>
<thead>
<tr>
<th>MANUFACTURER/COMPUTER</th>
<th>LOCATION</th>
<th>COUNTRY</th>
<th>CORES</th>
<th>R_{\text{Max}}</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM BladeCenter QS22/LS21, PowerXCell 3.2 GHz / Opteron 1.8 GHz, Voltaire Iband</td>
<td>DOE/NNSA/LANL</td>
<td>USA</td>
<td>129600</td>
<td>110500</td>
</tr>
<tr>
<td>Cray XT5 QC 2.3 GHz</td>
<td>DOE/OS/ORNL</td>
<td>USA</td>
<td>150152</td>
<td>105900</td>
</tr>
<tr>
<td>IBM Blue Gene/P Solution</td>
<td>Forschungszentrum Juelich</td>
<td>Germany</td>
<td>294912</td>
<td>825500</td>
</tr>
<tr>
<td>SGI Altix ICE 8200EX, Xeon QC 3.0/2.66 GHz</td>
<td>NASA/Ames Research Center/NAS</td>
<td>USA</td>
<td>51200</td>
<td>487005</td>
</tr>
<tr>
<td>IBM BlueGene/L - eServer Blue Gene Solution</td>
<td>DOE/NNSA/LLNL</td>
<td>USA</td>
<td>212992</td>
<td>478200</td>
</tr>
</tbody>
</table>

**PERFORMANCE DEVELOPMENT**

**PROJECTED PERFORMANCE DEVELOPMENT**
Thank You!

Q & A