Short Course: Advanced programming with MPI

Edgar Gabriel
Spring 2007

Multi-core processors
Parallel Computer

PC cluster in 2000
- 4U height per node
- Myrinet Network
- Fast Ethernet Network

(1U = 1.75" = 4.45 cm)
Parallel Computer

PC Cluster in 2005
- 1U height
- Infiniband Network
- Gigabit Ethernet Network

IBM BlueGene/L
131,072 Processors

System
(64 racks, 64x32x42)
131,072 procs

Rack
(32 Node boards, 8x8x16)
2048 processors

"Fastest Computer"
- BG/L, 100 MHz, 64K proc
- 32 racks
- Peak: 184 TFlop/s
- Linpack: 135 TFlop/s

Edgar Gabriel
Earth Simulator

- Target: Achievement of high-speed numerical simulations with processing speed of 1000 times higher than that of the most frequently used supercomputers in 1996. (www.es.jamstec.go.jp)
- 640 nodes
- 8 processors/node
- 40 TFLOPS peak
- 10 TByte memory

Top 500 List (www.top500.org)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Name/Machine</th>
<th>Country/Owner</th>
<th>Type</th>
<th>Processors</th>
<th>Memory</th>
<th>Speed</th>
<th>tops</th>
<th>System</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Earth Simulator</td>
<td>Japan</td>
<td>Cray</td>
<td>640 nodes</td>
<td>10 TByte</td>
<td>40 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>2</td>
<td>Titan</td>
<td>United States/DOE</td>
<td>Cray</td>
<td>640 nodes</td>
<td>10 TByte</td>
<td>10 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>3</td>
<td>Super Jain</td>
<td>Japan</td>
<td>Cray</td>
<td>256 nodes</td>
<td>5 TByte</td>
<td>10 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>4</td>
<td>The接着</td>
<td>Japan</td>
<td>Cray</td>
<td>128 nodes</td>
<td>2 TByte</td>
<td>5 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>5</td>
<td>National Supercomputer Facility</td>
<td>United States</td>
<td>Cray</td>
<td>64 nodes</td>
<td>1 TByte</td>
<td>2 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>6</td>
<td>Sequoia</td>
<td>United States/DOE</td>
<td>Cray</td>
<td>64 nodes</td>
<td>1 TByte</td>
<td>2 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>7</td>
<td>Red Storm</td>
<td>United States/DOE</td>
<td>Cray</td>
<td>32 nodes</td>
<td>0.5 TByte</td>
<td>1 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>8</td>
<td>Super JEM</td>
<td>Japan</td>
<td>Cray</td>
<td>16 nodes</td>
<td>0.25 TByte</td>
<td>0.5 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
<tr>
<td>9</td>
<td>Sequoia</td>
<td>United States/DOE</td>
<td>Cray</td>
<td>8 nodes</td>
<td>0.125 TByte</td>
<td>0.25 TFLOPS</td>
<td>1996.09</td>
<td>Research</td>
<td>1996.09</td>
</tr>
</tbody>
</table>
Top 500 List

Accompanying literature (I)


Accompanying literature (II)

- Accompanying literature

Flynn’s Taxonomy

- SISD: Single instruction single data
  - Classical von Neumann architecture
- SIMD: Single instruction multiple data
- MISD: Multiple instructions single data
  - Non existent, just listed for completeness
- MIMD: Multiple instructions multiple data
  - Most common and general parallel machine
Single Instruction Multiple Data (I)

- Also known as Array-processors
- A single instruction stream is broadcasted to multiple processors, each having its own data stream

Single Instruction Multiple Data (II)

- Interesting detail: handling of if-conditions
  - First all processors, for which the if-condition is true execute the according code-section, other processors are on hold
  - Second, all processors for the if-condition is not true execute the according code-section, other processors are on hold
- Some architectures in the early 90s used SIMD (MasPar, Thinking Machines)
- No SIMD machines available today
- SIMD concept used in processors of your graphics card
Multiple Instructions Multiple Data (I)

- Each processor has its own instruction stream and input data
- Most general case – every other scenario can be mapped to MIMD
- Further breakdown of MIMD usually based on the memory organization
  - Shared memory systems
  - Distributed memory systems

Shared memory systems (I)

- All processes have access to the same address space
  - E.g. PC with more than one processor
- Data exchange between processes by writing/reading shared variables
  - Shared memory systems are easy to program
  - Current standard in scientific programming: OpenMP
- Two versions of shared memory systems available today
  - Symmetric multiprocessors (SMP)
  - Non-uniform memory access (NUMA) architectures
Symmetric multi-processors (SMPs)

- All processors share the same physical main memory
- Memory bandwidth per processor is limiting factor for this type of architecture
- Typical size: 2-16 processors

NUMA architectures (I)

- Some memory is closer to a certain processor than other memory
  - The whole memory is still addressable from all processors
  - Depending on what data item a processor retrieves, the access time might vary strongly
NUMA architectures (II)

- Reduces the memory bottleneck compared to SMPs
- More difficult to program efficiently
  - First touch policy: data item will be located in the memory of the processor which touches the data item first
- To reduce effects of non-uniform memory access, caches are often used
  - ccNUMA: cache-coherent non-uniform memory access architectures
- Largest example as of today: SGI Origin with 512 processors

Distributed memory machines (I)

- Each processor has its own address space
- Communication between processes by explicit data exchange
  - Sockets
  - Message passing
  - Remote procedure call / remote method invocation
Distributed memory machines (II)

- Performance of a distributed memory machine strongly depends on the quality of the network interconnect and the topology of the network interconnect
  - Of-the-shelf technology: e.g. fast-Ethernet, gigabit-Ethernet
  - Specialized interconnects: Myrinet, Infiniband, Quadrics, ...

Distributed memory machines (III)

- Two classes of distributed memory machines:
  - Massively parallel processing systems (MPPs)
    - Tightly coupled environment
    - Single system image (specialized OS)
  - Clusters
    - Of-the-shelf hardware and software components such as
      - Intel P4, AMD Opteron etc.
      - Standard operating systems such as LINUX, Windows, BSD UNIX
Hybrid systems

- E.g. clusters of multi-processor nodes