Lecture 19

Large Scale Computing


Announcements

• Assignment #6 due on Thursday
• Final exam review next time
• Valkyrie is up and all 16 processors running

Today’s lecture

• Large scale computing
• A brief detour into bin sorting on NERSC’s IBM SP “Seaborg” system
• Leading edge systems
  – Multi-tier systems and programming
  – Earth simulator
  – Blue Gene/L

NERSC Seaborg system

• Located at Lawrence Berkeley Lab
  – “Multi-tier” SMP cluster
• SMP nodes
  – 380 16-way Power3+ “wide” compute nodes [6,080 processors]
  – 375 MHz CPU
  – 16 to 64 GB memory per node
  – 64 KB L1, 8MB L2 per processor
  – Caches have a 128 Byte line size
• Differential MPI communication rates
  (peak Ring)
  – 400 MB/sec off-node
  – 500 MB/sec on node
Multi-tier Computers

- Hierarchical organization
  - Each node contains several processors sharing memory
  - Trades off switch cost against peak floating point performance
- Trends
  - $r_\infty$: DGEMM floating point rate per node, MFLOP/s
  - $B_\infty$: peak MPI message BW, MBYTE/s
- NPACI Blue Horizon: $r_\infty = 14,000$, $B_\infty = 400$
- Previous generation SP system
  IBM SP2/Power2SC: $r_\infty = 640$, $B_\infty = 100$

What programming models are available for multi-tier computers?

- Single Tier
  - Flatten the hierarchical communication structure of the machine: one level or “tier” of parallelism
  - Simplest approach; MPI codes are reusable
  - But may not use shared memory effectively
- Multi-tier
  - Cognizant of the hierarchical communication structure of the machine
  - More complicated hybrid programming model

Bucket sorting on NERSC Seaborg system

- Scale input size with the number of processors

Multi-tier programming

- For an n-level machine, we identify n levels of parallelism + one collective level of control
- We have 3 levels of control:
  - Collective level: operations performed on all nodes
  - Node level: operations performed on one node
  - Processor level: operations performed on one processor
- A simple approach to realizing this model is hybrid programming
  - MPI + OpenMP (or threads)
Motivating application

- Iterative finite difference solver for Poisson’s equation in 3 dimensions
- Red black ordering

\[
\text{for } (i,j,k) \text{ in } 1:N \times 1:N \times 1:N \\
\text{where } (i+j+k) \text{ is odd/even} \\
u[i][j][k] = (u[i-1][j][k] + u[i+1][j][k] + \\
u[i][j-1][k] + u[i][j+1][k] + \\
u[i][j][k+1] + u[i][j][k-1])/6;
\]

Multi-tier or hybrid formulation

- Hierarchical control flow
  - One process per node, \( p \) threads per process
  - One task on each node executes communication
  - A different task carries out computation
  - Employ multithreading within a task

Overlap strategy

- Isolate the inner region from the halo
- Defer computation on the annulus
- Execute communication concurrently with computation on the inner region
- Compute on the annulus when the halo finishes
Multi-tier overlapped variant

- Hierarchical control flow
  - One process per node, \( p \) threads per process
- As with single-tier overlapped variant
  - Overlap communication with computation
  - Defer computation on the inner annulus

More about the model

- Communication reflects the hierarchical organization of the hardware
- Two kinds of communication: slow messages, fast shared memory
- One task per node communicates on behalf of the node’s processors

A few implementation details

- Some versions of MPI can realize overlap with MPI_IRecv and MPI_Isend
- If not, then we can use multithreading to handle the overlap
- We let one or more processors handle communication

A performance model of overlap

- Assumptions
  - \( p \) = number of processors per node
  - Running time = 1.0
  - \( f < 1 \) = communication time with MT(k) (i.e. not overlapped)

\[
T = 1.0 - f
\]
Performance

- When we displace computation to make way for the proxy, computation time increases.
- Wait on communication drops to zero ideally.
- When $f < p/(2p-1)$: improvement is $(1-f)x(p/(p-1))^f$.
- Communication bound: improvement is $1/(1-f)$.

\[ T = \frac{1}{1-f} \]  
\[ T = 1.0 \]  
\[ T = (1-f)x(p/(p-1)) \]

NPACI Blue Horizon

- Multiple SMP nodes:
  - 144 8-way Power3+ “high” nodes
  - 375 MHz CPU
  - 4 GB memory per node
  - 64 KB L1S, 4MB L2S per processor
  - Caches have a 128 Byte line size
- Differential MPI communication rates (peak Ring):
  - 400 MB/sec off-node
  - 500 MB/sec on node

SMP Node performance

- Shared memory contention limits performance.
- 57% parallel efficiency on 1 node w/ 8 threads.

Redblack3D on 1 node

Performance improves with overlap

- Computation time increases with fewer CPUs.
- But loss is mitigated by memory system saturation at high levels of parallelism.

Redblack3D, NPACI Blue Horizon, 6 Nodes
Performance Summary

- 14% improvement on 8 nodes, max theoretical 9%
  - Displacing one processor to make room for the proxy slows down computation only 5%
  - We predicted \( \frac{1}{7} = 14\% \)
  - The proxy does not slow down computation when we shut off communication
- 10.5% improvement on 27 nodes, max theoretical 17%
  - Requires tuning in the messaging layer

A closer look at performance

<table>
<thead>
<tr>
<th>NThrs</th>
<th>Total</th>
<th>Wait</th>
<th>Compute</th>
<th>Proxy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>293</td>
<td>16.9</td>
<td>279</td>
<td>16.8</td>
</tr>
<tr>
<td>6</td>
<td>55.1</td>
<td>15.4</td>
<td>41.2</td>
<td>15.4</td>
</tr>
<tr>
<td>7</td>
<td>50.1</td>
<td>15.6</td>
<td>36.3</td>
<td>15.6</td>
</tr>
<tr>
<td>8</td>
<td>45.5</td>
<td>15.6</td>
<td>31</td>
<td>15.6</td>
</tr>
<tr>
<td>6</td>
<td>49.9</td>
<td>0.398</td>
<td>49.8</td>
<td>32.6</td>
</tr>
<tr>
<td>7</td>
<td>44.2</td>
<td>1.48</td>
<td>44.2</td>
<td>31.9</td>
</tr>
<tr>
<td>8</td>
<td>48.4</td>
<td>6.62</td>
<td>42.5</td>
<td>34.1</td>
</tr>
</tbody>
</table>

Further performance tuning

- Communication wait times are still long: 20 ms per iteration
- On closer examination, processors don’t get an even share of memory bandwidth

Another level of the memory hierarchy

- There are two ports to memory
- Each group of 4 processors shares one port
- With overlap, the proxy is using one CPU
  - But the proxy is utilized only about 30% of the time
  - Therefore, the remaining 5 processors obtain more memory bandwidth per processor than the other 4
- Solution: non-uniformly partition the work over the threads
- Improves performance by 7%
- User, rather than the thread scheduler, mitigates the imbalance
Scalability issues

- On 27 processors, we'd like to use a 3 x 3 x 3 decomposition
- But we had to resort to a 2D decomposition
- Recall that we sweep over a thin inner annulus after communication completes
- With 3D decompositions, two faces of the annulus exhibit poor cache locality
  - A cache miss on each element
  - A TLB miss as well
  - Little reuse since we are sweeping only one plane
- We spend comparable time sweeping over the annular faces as we do on the rest of the mesh!

Other large scale systems

- TCS system at Pittsburgh Supercomputing Center
  - 750 4-way Alpha SMPs (1GHz)
- Earth simulator
  - 640 8-way nodes
    - 40 TF peak performance
    - 26.6 TFlops best performance for a real application (65% of peak)
    - 3 Earth simulator entries won the Gordon Bell prize at SC 2002
  - 500 MHz NEC vector CPUs
    - Heavy pipelining, extra wide vector registers
    - A single instruction carries out an implied loop of up to 256 computations
    - 8 GF peak per CPU
  - Massive shared memory bandwidth on-node
    - 32 memory ports
    - 2048 way interleaving: rapid access to a region of contiguous memory
    - Processors can stream data at a high rate from memory
  - High performance interconnect
    - Crossbar with a direct connection between all pairs of processors
    - 12.3 GF point-to-point bi-direction communication, all pairs
    - Special network for doing barrier synchronization in 3.3 µs
### Programming

- Uses hybrid MPI-OpenMP model
- Also uses vectorization technology (compiler) to exploit vector units
  - This is like an additional level of parallelism
- Uses MPI-2 extensions
  - MPI_Put: single sided communication combines benefits of shared memory and message passing
  - 11.6 GB/sec, \( a = 6.6 \mu s \)

### Blue Gene/L

- IBM-US Dept of Energy collaboration
- 64K dual processor nodes: 180 (360) TF peak
  - One CPU dedicated to communication on each node
    - But may be used for computation
  - Includes L3 cache
  - L1 caches not coherent
- Low power
  - Relatively slow processors; power PC 440
  - Small memory (256 MB)
- High performance interconnect
  - 3D toroidal mesh (end around), 175 MB peak, bidirectional
  - Rapid combining-broadcast network, 300/MB + 1.5 \( \mu s \) (1-way)
  - Fast barriers (1.5 \( \mu s \))
- Lightweight kernel runs on each node
  - Single user, single 2-thread process
  - No context switching, no demand paging