Partitioning in Particle Transport Problem

Ping (Ann) An
Particle Transport Computation

Problems:
- Regular Grid problem
  KBA, Hyb, Ext_Hyb, Vol
- Unstructured Arbitrary Grid problem
  currentlymetis and kba-like partitioning
Analytical Study of Regular Grids

- KBA partitioning
- Contention on P4 added extra terms into $T_{\text{total}}$, the parallel running time

$T_{\text{total}} = (y + x - 1)C_p + (x-1)C_m + yC_p + yC_p + yC_p$

*Cp*: computation cost per cellset

*Cm*: communication cost per cellset
Extended Hybrid: add more processors in the central column(s), based on Hybrid partitioning

\[ \text{Cp} = \text{Cm} = 1; \; x = y = 9; \]

<table>
<thead>
<tr>
<th></th>
<th>T_total</th>
<th>KBA</th>
<th>Vol</th>
<th>Hyb</th>
<th>Ext_ Hyb</th>
</tr>
</thead>
<tbody>
<tr>
<td>A only</td>
<td>25</td>
<td>29</td>
<td>26</td>
<td>26</td>
<td>29</td>
</tr>
<tr>
<td>A&amp;B</td>
<td>34</td>
<td>38</td>
<td>31</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>A&amp;D</td>
<td>34</td>
<td>38</td>
<td>26</td>
<td>26</td>
<td>31</td>
</tr>
<tr>
<td>All 4</td>
<td>52</td>
<td>65</td>
<td>36</td>
<td>36</td>
<td>34</td>
</tr>
</tbody>
</table>
Outline

- Evaluation of regular grids partitioning methods
- Preliminary performance modeling of regular grids
- Arbitrary grids partitioning methods
  - Evaluated through simulation
Sweep Times for Strong Scaling of A&M Code on Frost

- Grid Size: 24x48x24
- $K_z = 1$ (1 plane blocking)
- S2, 1 iteration
- Hybrid better than KBA and Volumetric
- Extended Hybrid surpasses Hybrid

*Frost*: IBM RS/6000 SP system, 68-node, 16-way symmetric multiprocessor (SMP) (POWER3 375 MHz NH2 SMP), 16 GB memory per node, 2 GB/sec. node-to-node bandwidth (LLNL)
Speedups for Strong Scaling of A&M Code on MCR

- Grid Size: 64 cube
- Kz = 1 (1 plane blocking)
- S6, 45 iterations mostly
- Hybrid is better than KBA and Volumetric
- Extended Hybrid catches up with Hybrid

MCR: Linux cluster, 1,152 nodes, two 2.4-GHz Pentium 4 Xeon processors with 4 GB memory per node (LLNL)
Weak Scaling Results of A&M Code on MCR

- Problem/proc:
  - KBA: 8x8x256
  - Hyb: 16x8x256
  - Kz = 1 (1 plane blocking)
  - S6, 45 iterations mostly
Outline

- Evaluation of regular grids partitioning methods
- Preliminary performance modeling of regular grids
  - Code in performance tuning stage
  - Not enough data to calibrate the model accurately
- Arbitrary grids partitioning methods
  - Evaluated through simulation
Performance Modeling (Preliminary Study)

- Calibrate the model
  - $tc$: Computation cost per cell
  - $tw$: Communication cost per cell
  - $ts$: Communication startup cost

- Validate with experimental results

- Predict the behavior of Vol, KBA, Hyb, and Ext_Hyb
  - Evaluate current implementation by comparing with observed behavior

- Study the cellset effects
  - What’s the optimal cellset size? The size and shape determine the computation to communication ratio
  - How the cellset should affect the behavior of different partitioning methods?
Regular Grid Performance Model

Grid (nx, ny, nz)
procs (px, py, pz)
cellset (kx,ky,kz)

Grid: (nx=32, ny=1, nz=16)
Procs: (px=4, py=1, pz=2)
Cellset: (kx=8, ky=1, kz=2)

\[ T_{\text{total}} = T_{\text{comp}} + T_{\text{comm}} \]  \hspace{1cm} (1)

\[ T_{\text{comp}} = Cp \left( \frac{nx}{kx} + \frac{ny}{ky} + \frac{nz}{kz} - 2 \right) + ctf \cdot Cp \cdot \frac{nz}{pz \cdot kz} \]  \hspace{1cm} (2)

\[ Cp = tc \cdot kx \cdot ky \cdot kz; \]  \hspace{1cm} (3)

\[ tc : \text{computing cost per cell} \]

\[ ctf = \begin{cases} 
7 & pz = 1, KBA \\
3 & pz > 1, Other 
\end{cases} \]  \hspace{1cm} (4)

\[ T_{\text{comm}} = C_{mx} \cdot (px - 1) + C_{my} \cdot (py - 1) + C_{mz} \cdot (pz - 1) \]  \hspace{1cm} (5)

\[ C_{mx} = ts + tw \cdot (ky \cdot kz) \]  \hspace{1cm} (6)

\[ C_{my} = ts + tw \cdot (kx \cdot kz) \]  \hspace{1cm} (7)

\[ C_{mz} = ts + tw \cdot (kx \cdot ky) \]  \hspace{1cm} (8)

\[ ts : \text{communication startup cost} \]
\[ tw : \text{communication cost per cell} \]

\[ tc, ts, tw \text{ are 3 machine specific unkowns we can estimate based on experiments.} \]
Model Calibration

Used 3 values in KBA results:

\[ t_c = 0.0129 \text{ ms} \]
\[ t_s = 1.3486 \text{ ms} \]
\[ t_w = 0.1974 \text{ ms (per cell)} \]

<table>
<thead>
<tr>
<th>Machine</th>
<th>( t_s )</th>
<th>( t_w )</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM SP2</td>
<td>40</td>
<td>0.11</td>
</tr>
<tr>
<td>Intel DELTA</td>
<td>77</td>
<td>0.54</td>
</tr>
<tr>
<td>Intel Paragon</td>
<td>121</td>
<td>0.07</td>
</tr>
<tr>
<td>Meiko CS-2</td>
<td>87</td>
<td>0.08</td>
</tr>
<tr>
<td>nCUBE-2</td>
<td>154</td>
<td>2.4</td>
</tr>
<tr>
<td>Thinking Machines CM-5</td>
<td>82</td>
<td>0.44</td>
</tr>
<tr>
<td>Workstations on Ethernet</td>
<td>1500</td>
<td>5.0</td>
</tr>
<tr>
<td>Workstations on FDDI</td>
<td>1150</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Table 3.1: Approximate machine parameters for some parallel computers, in microseconds sec.

*Designing and Building Parallel Programs*, by *Ian Foster (1995)*
Model Evaluation with 64cube Hybrid

Observed and predicted sweep time per iteration for HYB

Sweep Time per Iteration (ms)

Number of processors

Hyb Observed
Hyb Predicted
Predicted and Observed (64cube)
Cellset Effect

A simplified model assuming:

\[
\begin{align*}
\text{nx} &= \text{ny} = n \\
\text{kx} &= \text{ky} = k \\
\text{kz} &= 1 \\
\text{px} &= \text{py} = p = n / k
\end{align*}
\]

\[
T_{\text{comp}} = t_c \cdot k^2 \cdot \left( \frac{2n}{k} + nz - 2 \right) + ctf \cdot \frac{nz}{pz}
\]

\[
T_{\text{comm}} = ts \cdot (2p + pz - 3) + 2 \cdot tw \cdot k(p - 1) + tw \cdot k^2 \cdot (pz - 1)
\]

For strong scaling with fixed grid size (nx, ny, nz):

\[k \uparrow , T_{\text{comp}} \uparrow\]

\[k \uparrow , p \downarrow , T_{\text{comm}}?\]
Predicted and Observed Cellset Effects

Strong scaling 64cube
Outline

● Evaluation of regular grids partitioning methods

● Preliminary performance modeling of regular grids

● Arbitrary grids partitioning methods
  – Evaluated through simulation
Arbitrary Grids Partitioning
Analysis of Cylinder Grids

Why analyze cylinder grids?
- Convex geometry shapes of arbitrary grids
  - Physical constraint that no re-entrant from geometric boundary
  - Good simulation of arbitrary grids
- A common grid type in transport computation
  - E.g. RZ problem with reflection boundaries.

How would different partitioning schemes affect the makespans?
- Would more processors in the contention center help?
- How should they be added?
Perfect Cylinder Grids Partitioning

Add more processor in contention center
Cylinder Grids Analysis

• Radius partitioning is always better than circular

• Processors added to the radius direction break the crucial path, produces worse partition.

  Crucial Path: the path from a source to a sink cell, and passes through the contention center. (not necessarily the critical path)

• Radius partitioning makes the sweeps interfere the least with each other’s crucial path
Strategies for Arbitrary Grids partitioning

- Divide contention region
- Preserve crucial paths
- Let as many processes to start from the beginning as possible
Contention Analysis

- Adjacent sweeps may create contention fronts
  Where 2 adjacent sweeps collide

- Multiple sweeps may create contention regions
  where multiple sweeps reach within a time window
Simple Partitioning Algorithm

- For $M$ sweeps, considers $R^M$ in the problem only
  - Most demanding contention region

- Often reflects some basic problems
  - Brick, Cylinder or sphere-like grids with uniform or close to uniform meshes

- Basic shapes are building blocks for complex problems

- Mimic radius partitioning
RTT Mesh

Note: RTT Mesh generator is from LANL
Simple Algorithm Applied to RTT Mesh

Simulated Makespans of RTT Mesh Partitioning
Hex(10x10x1), 6 Tetrahedral per Hex
600 cells, 4 anglesets

- **KBA-mimic**
- **Metis**
- **SA**

<table>
<thead>
<tr>
<th>Number of Processors</th>
<th>Makespan</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>32</td>
</tr>
</tbody>
</table>
Simulation Results with Arbitrary Mesh

Simulated Makespans of Arbitrary Grid Partitioning
1183 elements, 4 angles (45, 135, 225, 315 degrees)

- Multiple contention regions
- 3D structure
  - 3 cylinders in an ellipse
Conclusions and Future Work

• Simple algorithm considering contention region is better than Metis and KBA like for arbitrary grids partitioning

• Preserving crucial path and balancing work load are both important

• Partitioning algorithms for multiple contention region problems

• Evaluate the algorithms with A&M code

• Apply the algorithms in time dependent problems