DDDAS Project Update

D. Fuentes

Institute for Computational Engineering and Sciences
The University of Texas at Austin

Austin, Texas
February 1, 2007
Outline

- Hybrid OpenMP/MPI Paradigm

- Code Demonstration
  
  *Data Transfer, Filtering, Visualization*

- Computations
  
  *Optimization window*
OpenMP/MPI Paradigm

- Computer Architecture at TACC is cluster of SMP nodes

  *Lonestar: 4 processors per compute nodes*

- Profiling shows linear solve time $\approx$ constant as increase processors for a given problem size

- Large number of processors communication between MPI tasks dominates solve time

- OpenMP Offers a way to avoid communication overhead

  *no communication between OpenMP threads, but Fork/Join overhead very expensive*
OpenMP/MPI Paradigm

### Sheet

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameter 1</td>
<td>Value 1.2</td>
<td>Value 2.3</td>
<td>Value 3.4</td>
</tr>
<tr>
<td>Parameter 2</td>
<td>Value 1.5</td>
<td>Value 2.6</td>
<td>Value 3.7</td>
</tr>
<tr>
<td>Parameter 3</td>
<td>Value 1.8</td>
<td>Value 2.9</td>
<td>Value 3.0</td>
</tr>
</tbody>
</table>

**MPI VS HYBRID**

<table>
<thead>
<tr>
<th>Time (ms)</th>
<th>Column C</th>
<th>Column D</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.23</td>
<td>2.34</td>
<td>3.45</td>
</tr>
<tr>
<td>2.34</td>
<td>3.45</td>
<td>4.56</td>
</tr>
<tr>
<td>3.45</td>
<td>4.56</td>
<td>5.67</td>
</tr>
</tbody>
</table>

**End of Document**
Code Demonstration

- Data Transfer scripts written in Python

- Bottleneck was writing to disk

  Solution: collect onto one processor, collect into one buffer then write visualization files to disk

- Computations

  Optimization window
- **Optimization window**

  *TWO groups of processors: 1st group is constantly calibrating and sending the calibrated parameters to 2nd group of processors. 2nd group constantly solving optimal control problem given calibrated data.*

- **Time constraint restrictions imposed by real time calculation allow only a handful of objective function/gradient evaluations**

  *Quasi Newton Methods taking too many iterations
  Steepest descent guarantees objective function decrease*