Gedae Runtime Kernel
Performance Characterization

Kerry Barnes, William Lundgren,
James Steed
Gedae, Inc.

Gedae, Inc.
www.gedae.com
Efficiency of Gedae Applications

- How efficient are Gedae applications compared to hand coding?
- What is the overhead of the Gedae Runtime Kernel?
- What is the cost of executing a primitive?
- What is the cost of dynamic vs. static scheduling?
Instrumenting Kernel

- Gedae Kernel instrumented with timer functions
- 44 sections of code were timed
- Benchmarks
  - rt_stay_easy: static scheduling
  - e_comm: dynamic scheduling
  - noise_removal: dynamic scheduling with segmentation
- Timings gathered on AltiVec targets from 3 vendors
- 1-3 usec overhead for all benchmarks

<table>
<thead>
<tr>
<th>Graph:</th>
<th>rt_stap_easy</th>
<th>e_comm</th>
<th>noise_removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proc</td>
<td>Efficiency</td>
<td>Overhead (usec/fire)</td>
<td>Efficiency</td>
</tr>
<tr>
<td>AltiVec1</td>
<td>97</td>
<td>1.7</td>
<td>90</td>
</tr>
</tbody>
</table>

Gedae, Inc.
www.gedae.com
Categorize Overhead

- Four categories accounted for most of the overhead time
- Suggested enhancements:
  - Code generation of static schedule
  - Eliminate copies by directly adjusting pointers
  - Improve schedule state change functions

<table>
<thead>
<tr>
<th>Function</th>
<th>rt_stap_easy</th>
<th>e_comm</th>
<th>noise_removal</th>
</tr>
</thead>
<tbody>
<tr>
<td>run-sched-pass</td>
<td>90%</td>
<td>20%</td>
<td>18%</td>
</tr>
<tr>
<td>copy functions</td>
<td>18%</td>
<td>9%</td>
<td></td>
</tr>
<tr>
<td>schedule state</td>
<td>36%</td>
<td>26%</td>
<td></td>
</tr>
<tr>
<td>queue management</td>
<td>10%</td>
<td>15%</td>
<td></td>
</tr>
<tr>
<td>Total % of Overhead</td>
<td>90%</td>
<td>84%</td>
<td>68%</td>
</tr>
</tbody>
</table>

We can comprehensively measure the overhead and systematically optimize the most costly components