READEX – Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing

EE HPC WG @ SC’17

Andreas Gocht – Technische Universität Dresden
Overview: Project Partners

• Grant agreement No 671657
• Officially started September 1st, 2015

• Technische Universität Dresden/ZIH (Coordinator)
• Norwegian University of Science and Technology
• Technische Universität München
• IT4Innovations, VSB-Technical University of Ostrava
• NUI Galway, Irish Centre for High-End Computing
• Intel France
• Gesellschaft für numerische Simulation mbH
Applications exhibit dynamic behaviour

- Changing resource requirements
- Computational characteristics
- Changing load on processors over time
Overview

READEX creates a **tools-aided methodology for automatic tuning** of parallel applications

- Dynamically adjust system parameters to actual resource requirements

Join technologies from Embedded Systems and HPC

- HPC: PTF, Score-P, and HDEEM
- Embedded Systems: System scenario methodology
Overview

Co-design approach

• Manual tuning for energy efficiency as a baseline
• Automatic tuning for comparison

• Applications
  • PERMON and ESPRESO (FETI tools from IT4Innovations)
  • Indeed (GNS)
  • CORAL benchmark suite
  • ProxyApps
int main(void)
{
    // Initialize application
    int num_iterations = 2;
    for (int iter = 1; iter <= num_iterations; iter++)
    {
        laplace3D();
        reduction();
        fftw_execute();
    }
    // Post processing and finalization
    return 0;
}
```c
int main(void)
{
    // Initialize application
    int num_iterations = 2;
    for (int iter = 1; iter <= num_iterations;
    {
        laplace3D();
        reduction();
        fftw_execute();
    }
    // Post processing and finalization
    return 0;
}
```

Terminology: Tuning Parameter

Tuning Parameter
FREQ=1.5 GHz

Tuning Parameter
FREQ=2 GHz
Overview: Workflow

1. Instrument application
   Score-P provides different kinds of instrumentation

2. Detect dynamism
   Check whether runtime situations could benefit from tuning

3. Detect energy saving potential and configurations (DTA)
   Use tuning plugin and power measurement infrastructure to search for optimal configuration
   Create tuning model

4. Runtime application tuning (RAT)
   Apply tuning model, use optimal configuration
Current Status

- Application instrumentation, dynamism detection, DTA, and RAT are implemented
- Promising results:
Comparison of power consumption and runtime of the BT benchmark
Example: NAS-OMP BT.C Benchmark

Selected tuning parameter: core and uncore frequency of the BT benchmark
Example: NAS-OMP MG.D Benchmark

Comparison of power consumption and runtime of the MG benchmark
Example: NAS-OMP MG.D Benchmark

Selected tuning parameter: core and uncore frequency of the MG benchmark
• Evaluating Application Tuning Parameters (ATPs)
  • Allow the tuning of program internal “decisions”
  • Example preconditioner in ESPRESO a solver for FETI problems:

<table>
<thead>
<tr>
<th>Preconditioner type</th>
<th>Number of iterations</th>
<th>Single iteration cost</th>
<th>Total solution cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Time and energy</td>
<td></td>
</tr>
<tr>
<td>No preconditoner</td>
<td>172</td>
<td>130 + 0 ms</td>
<td>21.4 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.3 + 0.00 J</td>
<td>5.50 kJ</td>
</tr>
<tr>
<td>Weight function</td>
<td>100</td>
<td>130 + 2 ms</td>
<td>12.9 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.3 + 0.53 J</td>
<td>3.28 kJ</td>
</tr>
<tr>
<td>Lumped</td>
<td>45</td>
<td>130 + 10 ms</td>
<td>6.3 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.3 + 3.86 J</td>
<td>1.64 kJ</td>
</tr>
<tr>
<td>Light Dirichlet</td>
<td>39</td>
<td>130 + 10 ms</td>
<td>5.5 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.3 + 3.74 J</td>
<td>1.41 kJ</td>
</tr>
<tr>
<td>Full Dirichlet (default)</td>
<td>30</td>
<td>130 + 80 ms</td>
<td>6.3 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td>32.3 + 20.6 J</td>
<td>1.59 kJ</td>
</tr>
</tbody>
</table>

11.3% energy savings against the default full Dirichlet preconditioners

Note: 130 ms and 32.3 J – is a baseline for single iteration cost without preconditioner
Work in Progress

• Runtime Calibration
  • Automatically finds a good configuration for unseen scenarios
  • Applied at runtime
  • Machine Learning techniques
Thank you for your attention

Questions?