E-Team: Practical Energy Accounting for Multi-Core Systems

Till Smejkal¹, Marcus Hähnel¹, Thomas Ilsche², Michael Roitzsch¹, Wolfgang E. Nagel², and Hermann Härtig¹

¹ Operating Systems Group, TU Dresden
² Center for Information Services and High Performance Computing (ZIH), TU Dresden

July 13th 2017
Existing energy measurement methods are not suited for energy-based billing or accurate energy profiling of applications.
Motivation

Existing energy measurement methods are not suited for energy-based billing or accurate energy profiling of applications.

External Measurement

- 😊 Accurate energy/power values
- 😞 Limited to machine-level granularity
- 😞 Expensive deployment for data centers
Motivation

Existing energy measurement methods are not suited for energy-based billing or accurate energy profiling of applications.

<table>
<thead>
<tr>
<th>External Measurement</th>
<th>Inference-Based Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>☺ Accurate energy/power values</td>
<td>☺ Adjustable granularity and resolution</td>
</tr>
<tr>
<td>☹ Limited to machine-level granularity</td>
<td>☹ Calibration required for every system</td>
</tr>
<tr>
<td>☹ Expensive deployment for data centers</td>
<td>☹ 8% to 10% inaccuracy</td>
</tr>
</tbody>
</table>
Running Average Power Limit
Running Average Power Limit

In 2011 Intel® introduced *Running Average Power Limit* (RAPL) into their processors, a feature to limit the power consumption of the processor.
In 2011 Intel® introduced *Running Average Power Limit* (RAPL) into their processors, a feature to limit the power consumption of the processor.

- Processor internally estimates the consumed energy
- Four energy counters are exported by the processor
- Each CPU socket has separate energy counters
In 2011 Intel® introduced *Running Average Power Limit* (RAPL) into their processors, a feature to limit the power consumption of the processor.

- Processor internally estimates the consumed energy
- Four energy counters are exported by the processor
- Each CPU socket has separate energy counters

**RAPL is very promising for on-line energy measurements.**
RAPL only reports the energy consumption of the whole CPU socket.
RAPL only reports the energy consumption of the whole CPU socket.
RAPL only reports the energy consumption of the whole CPU socket.
RAPL only reports the energy consumption of the whole CPU socket.
Running Average Power Limit

Spatial Granularity

RAPL only reports the energy consumption of the whole CPU socket.

```
App1
  T1  T2

App2
  T1  T2  T3

T1
Core1  Core2  Core3  Core4
```
Running Average Power Limit

Spatial Granularity

RAPL only reports the energy consumption of the whole CPU socket.
Running Average Power Limit

Spatial Granularity

RAPL only reports the energy consumption of the whole CPU socket.

RAPL: 18J

Till Smejkal [TU Dresden]
Running Average Power Limit

Spatial Granularity

App_1

T_1

App_2

T_1

C_2

C_1

Time
Running Average Power Limit

Spatial Granularity

busy-loop

FIRESTARTER

\[ T_1 \]

\[ T_1 \]

C_2

C_1

Time

Till Smejkal [TU Dresden]
Running Average Power Limit

Spatial Granularity

Till Smejkal [TU Dresden]
Running Average Power Limit

Spatial Granularity

Till Smejkal [TU Dresden]
Running Average Power Limit

Spatial Granularity

CPU Time

Instructions

Energy (CPU+Memory) [J]

busy-loop  FIRESTARTER

Till Smejkal [TU Dresden]
Instructions

Energy (CPU+Memory) [J]

Baseline
CPU Time
Instructions

busy-loop  FIRESTARTER

Baseline  404.4  549.3
CPU Time  499.0  451.7
Instructions  180.0  770.7
Running Average Power Limit

Spatial Granularity

Instructions

Energy (CPU+Memory) [J]

E-Team: 408.4 busy-loop, 549.9 FIRESTARTER
Baseline: 404.4 busy-loop, 549.3 FIRESTARTER
CPU Time: 499.0 busy-loop, 451.7 FIRESTARTER
Instructions: 180.0 busy-loop, 770.7 FIRESTARTER
Running Average Power Limit

Temporal Granularity

RAPL energy counters are updated only at discrete intervals.
RAPL energy counters are updated only at discrete intervals.
Running Average Power Limit
Temporal Granularity

RAPL energy counters are updated only at discrete intervals.

Till Smejkal [TU Dresden]
Running Average Power Limit
Temporal Granularity

RAPL energy counters are updated only at discrete intervals.
Running Average Power Limit
Temporal Granularity

RAPL energy counters are updated only at discrete intervals.

App₁

T₁ T₂

C₁ C₂

Time
RAPL energy counters are updated only at discrete intervals.
Running Average Power Limit

Temporal Granularity

RAPL energy counters are updated only at discrete intervals.
Running Average Power Limit
Temporal Granularity

RAPL energy counters are updated only at discrete intervals.
**Running Average Power Limit**

**Temporal Granularity**

RAPL energy counters are updated only at discrete intervals.
Running Average Power Limit

Temporal Granularity

[Diagram showing time axis and power limits for different applications]

C_1

C_2

App_1

T_1

Till Smejkal [TU Dresden]
Running Average Power Limit

Temporal Granularity

C1  C2

App1

T1

Time

Till Smejkal [TU Dresden]
Running Average Power Limit

Temporal Granularity

Till Smejkal  [TU Dresden]
Running Average Power Limit

Temporal Granularity
Running Average Power Limit

Temporal Granularity

Naïve

Energy (CPU+Memory) [J]
Running Average Power Limit

Temporal Granularity

Energy (CPU+Memory) [J]

Baseline: 6.3
Naïve: 5.7

Till Smejkal [TU Dresden]
Running Average Power Limit

Temporal Granularity

- E-Team: 6.3 J
- Baseline: 6.3 J
- Naïve: 5.7 J

Energy (CPU+Memory) [J]

Till Smejkal [TU Dresden]
E-Team
E-Team allows per-application energy accounting based on socket-local measurements by leveraging a specialized scheduling scheme.
E-Team allows per-application energy accounting based on socket-local measurements by leveraging a specialized scheduling scheme.
E-Team allows per-application energy accounting based on socket-local measurements by leveraging a specialized scheduling scheme.
E-Team allows per-application energy accounting based on socket-local measurements by leveraging a specialized scheduling scheme.
E-Team allows per-application energy accounting based on socket-local measurements by leveraging a specialized scheduling scheme.
E-Team
Solving Spatial Granularity

Team 1

Team 2

Till Smejkal [TU Dresden]
E-Team
Solving Spatial Granularity
E-Team
Solving Spatial Granularity

Team 1

Team 2

Core 1
Core 2
Core 3
Core 4

T1 T2
T1 T2 T3
T1

T1

T1

T1

T1

T2

T2

T3

T1

RAPL: 20J

RAPL: 10J

Till Smejkal [TU Dresden]
E-Team
Solving Spatial Granularity

Team 1

Team 2

Core 1
Core 2
Core 3
Core 4

RAPL: 20J 😊
E-Team
Solving Spatial Granularity

Team 1

\[ \text{T1} \quad \text{T2} \]

Team 2

\[ \text{T1} \quad \text{T2} \quad \text{T3} \]

Core 1  Core 2  Core 3  Core 4

RAPL: 20J

RAPL: 10J

Till Smejkal [TU Dresden]
E-Team
Solving Spatial Granularity

Team 1
T1  T2

Team 2
T1  T2  T3

Core 1  Core 2  Core 3  Core 4

Till Smejkal [TU Dresden]
E-Team
Solving Spatial Granularity

Team 1
T1 T2

Team 2
T1 T2 T3 T1

Core1 Core2 Core3 Core4

RAPL: 10J 😊

Till Smejkal [TU Dresden]
E-Team
Solving Temporal Granularity

<table>
<thead>
<tr>
<th>Team1</th>
<th>Team2</th>
</tr>
</thead>
<tbody>
<tr>
<td>T₁</td>
<td>T₁</td>
</tr>
<tr>
<td>T₂</td>
<td>T₂</td>
</tr>
<tr>
<td></td>
<td>T₃</td>
</tr>
<tr>
<td></td>
<td>T₁</td>
</tr>
</tbody>
</table>
E-Team
Solving Temporal Granularity
E-Team
Solving Temporal Granularity

Team 1
- T1
- T2

Team 2
- T1
- T2
- T3

Time

Till Smejkal [TU Dresden]
E-Team
Solving Temporal Granularity

Team 1

Team 2

C2

C1

Time

Till Smejkal [TU Dresden]
E-Team
Solving Temporal Granularity

Team1
T₁ T₂

Team2
T₁ T₂ T₃ T₁

C₂
C₁

Time

Till Smejkal [TU Dresden]
E-Team
Solving Temporal Granularity

Team 1

T1  T2

Team 2

T1  T2  T3

C2

C1

Time

Till Smejkal [TU Dresden]
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.

E-Team

Implementation

Till Smejkal [TU Dresden]
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team
Implementation

E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
E-Team is implemented as a new scheduler in the Linux kernel.
Evaluation
Evaluation

- Intel® Haswell Core™ i7-4770 with 3.4 GHz nominal frequency
- Disabled Hyper-Threading and Turbo Boost
- NAS Parallel Benchmarks\(^1\) (OpenMP version) with 1 to 4 threads
- Results are relative differences to execution in stripped-down environment

\(^1\)Bailey et al., “The NAS parallel benchmarks”.
Evaluation

Solo

Benchmark

T1  T2

Till Smejkal [TU Dresden]
Evaluation

Solo

Team_{BM}
Evaluation

Solo

Team\textsubscript{BM}

C\textsubscript{1} \quad C\textsubscript{2}

Time
Evaluation

Solo

CPU Time Overhead [%]

1 Thread  | 2 Threads  | 3 Threads  | 4 Threads
---------|------------|------------|------------

bt.A      |            |            |            |
ge.B      |            |            |            |
ep.B      |            |            |            |
ff.B      |            |            |            |
is.C      |            |            |            |
lu.A      |            |            |            |
mg.C      |            |            |            |
sp.A      |            |            |            |
uA.A      |            |            |            |
Evaluation

Solo

Till Smejkal [TU Dresden]
Evaluation

Background

Team_{BM}

T_1   T_2

Till Smejkal [TU Dresden]
Evaluation

Background
Evaluation

Background

- TeamBM
  - T\textsubscript{1}
  - T\textsubscript{2}

- Loop
  - T\textsubscript{1}

- Time
  - C\textsubscript{2}
  - C\textsubscript{1}
Evaluation

Background

CPU Time Overhead [%]

- 1 Thread
- 2 Threads
- 3 Threads
- 4 Threads

Till Smejkal [TU Dresden]
Energy Deviation [%]

1 Thread  2 Threads  3 Threads  4 Threads

Till Smejkal [TU Dresden]
Evaluation

Comparison with External

Team_{BM}

T_1 \quad T_2
Evaluation

Comparison with External

Team \(_{BM}\) vs Benchmark'
Evaluation

Comparison with External

Team_{BM}  
\begin{array}{c|c}
T_1 & T_2 \\
\end{array}

Team_{BM'}  
\begin{array}{c|c}
T_1 & T_2 \\
\end{array}

Till Smejkal [TU Dresden]
Evaluation
Comparison with External

Team_{BM}  
\begin{array}{c}
T_1 \\
T_2 
\end{array}

Team_{BM'}  
\begin{array}{c}
T_1 \\
T_2 
\end{array}

C_2

C_1

Time
Evaluation

Comparison with External

<table>
<thead>
<tr>
<th></th>
<th>E-Team</th>
<th>External</th>
</tr>
</thead>
<tbody>
<tr>
<td>ft.B</td>
<td>783.4</td>
<td>792.3</td>
</tr>
<tr>
<td>is.C</td>
<td>312.5</td>
<td>321.9</td>
</tr>
</tbody>
</table>

Energy [J]

Till Smejkal [TU Dresden]
Limitations & Conclusion
Limitations

- Exclusive access to whole CPU socket potentially wastes resources
- I/O-intensive applications require frequent use of *Short-Time RAPL*
Limitations

- Exclusive access to whole CPU socket potentially wastes resources
- I/O-intensive applications require frequent use of Short-Time RAPL

Both limitations can be mitigated using random sampling.
Conclusion

- Accurate energy accounting for arbitrary groups of threads and applications
- No additional equipment, calibration, adaptation required
- Easy-to-use library and command line tool
- Extendable to other energy measurement methods

https://github.com/TUD-OS/{eteam,eteam-rt}