Discover, model and combine energy leverages for large scale energy efficient infrastructures

Issam Raïs
Laurent Lefèvre, Anne Benoit, Anne-Cécile Orgerie

Univ. Lyon, Inria, CNRS, ENS de Lyon, UCBL 1, LIP, France

September 28, 2018
Energy, a global concern

An energy driven world\(^1\)

- Computing facilities, big electrical consumers
- In 2017, 7\% of global electricity demand
- 2\% of global carbon emission

Day to day scientific needs of ultra large scale computing

Tackling the unknown at all scales thanks to large scale computing

- Space: Square Kilometer Array (SKA) Project
- Brain: Human Brain Project (HBP)
- Particles: Large Hadron Collider (LHC)

The constant need for computing

- Create or gather huge amount of data
- Computation and data deluge
Large scale computing facilities

Answering computing demands implies high performance facilities

- Datacenters: set of centralized computing and data facilities
- Supercomputers: very large, high performance architecture

Supercomputing: the next milestone

- Exascale: $10^{18}$ floating point operations per second
- Reached by a single running machine
- Defense Advanced Research Projects Agency (DARPA): maximum consumption between 20 to 30 MW
Large scale computing facilities: an eco-system of users

- **Facility Provider**
  - Owns
  - Contract
  - Provisions
  - Controls

- **Electricity Provider**

- **Application User**
  - Uses

- **Administrator**
  - Manages
  - Configures

- **Service(s)**
Energy efficiency: a problem with multiple definitions, for multiple users

<table>
<thead>
<tr>
<th>User</th>
<th>Constraint</th>
<th>Scale</th>
<th>Figure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facility provider</td>
<td>Power envelope</td>
<td>Complete facility</td>
<td>(1) and (4)</td>
</tr>
<tr>
<td>Electrical provider Service</td>
<td>Power capping</td>
<td>Complete or partial</td>
<td>(2)</td>
</tr>
<tr>
<td>Administrator</td>
<td>Energy budget</td>
<td>User</td>
<td>(5) to (6)</td>
</tr>
<tr>
<td>Application user</td>
<td>Relaxed power capping</td>
<td>Complete or partial</td>
<td>(3)</td>
</tr>
<tr>
<td></td>
<td>Energy budget</td>
<td>User</td>
<td>(5) to (6)</td>
</tr>
</tbody>
</table>
OAK RIDGE’s Summit supercomputer

Architecture
- Low power CPUs: 9216 IBM Power9
- Low power GPUs: 27648 Nvidia Volta V100
- Number of nodes: 4608
- Memory: 250 PB
- Connectivity: 100G Infiniband

Characteristics
- 1st in Top 500, 5th in Green 500
- 122 PFLOPS, 1/8 ExaFlop
- First “integer Exascale” machine
- USA, footprint of 13MW → 13M$ per year
(Floating point) Exascale is coming!

Potential architecture

▶ Heterogeneous computing nodes
▶ Hundreds of thousands of computing nodes
▶ Hundreds of cores per node
▶ Dedicated and efficient network

Greatest challenge: energy consumption

▶ Free cooling
▶ Low-power processors
▶ Reuse heat
▶ Use energy-aware middleware
▶ Implement algorithms differently
Energy techniques on large scale computing facilities, the literature

Mono technique studies\(^2,^3,^4\)

- Lots of mono studies evaluation
- No standard definition of leverage

Multi technique studies

- Usually == 2
- No classification
- No automatic extraction of knowledge


\(^3\) Jie Han and Michael Orshansky. “Approximate computing: An emerging paradigm for energy-efficient design”. In: *Test Symposium (ETS), 2013 18th IEEE European*. IEEE. 2013, pp. 1–6.

Energy techniques on large scale computing facilities, the literature

Mono technique studies\(^2,^3,^4\)

- Lots of mono studies evaluation
- No standard definition of leverage

Multi technique studies\(^5,^6\)

- Usually == 2
- No classification
- No automatic extraction of knowledge

---


Energy techniques on large scale computing facilities, the literature

Mono technique studies$^2,^3,^4$

- Lots of mono studies evaluation
- No standard definition of leverage

Multi technique studies$^5,^6$

- Usually \( \approx 2 \)
- No classification
- No automatic extraction of knowledge

- No generic solution
- No automated solution
Energy capabilities: families

Infrastructure level
- Energy harvester
- Cooling system

Middleware level
- Scheduler policies
- OpenMP and MPI configuration

Hardware level
- Sleep states and shutdown techniques
- Dynamic voltage and frequency scaling

Application level
- Vectorization
- Computation precision
Energy capabilities: families

**Infrastructure level**
- Energy harvester
- Cooling system

**Middleware level**
- Scheduler policies
- OpenMP and MPI configuration

**Hardware level**
- Sleep states and shutdown techniques
- Dynamic voltage and frequency scaling

**Application level**
- Vectorization
- Computation precision
Leverage, Energy and Power leverage

Leverage
We define a leverage $L$ as a triplet $(S, s_c, f)$

- $S = \{s_1, s_2, ..., s_n\}$ is the set of possible states for $L$
- $s_c$ is the current state of $L$, $s_c \in S$
- $f$ is the function that permits the modification of $s_c$

Energy and power leverage
if and only if using it impacts directly or indirectly power or energy consumption of a machine or an IT facility
In this thesis: challenges and problems

Be energy efficient?

- Efficient at all levels: hard to implement
- Lot of expertise at various levels
- Using leverages ≠ being energy efficient
- Need automated techniques

Tackled problems

- How to evaluate and model a single energy and power leverage?
- How to automatically discover and benchmark chosen leverages?
- How to combine and orchestrate leverages in order to be energy efficient?
- How to extract knowledge from the combination of available leverages?
In this thesis: challenges and problems

Be energy efficient?

- Efficient at all levels: hard to implement
- Lot of expertise at various levels
- Using leverages ≠ being energy efficient
- Need automated techniques

Tackled problems

- How to evaluate and model a single energy and power leverage?
- How to automatically discover and benchmark chosen leverages?
- How to combine and orchestrate leverages in order to be energy efficient?
- How to extract knowledge from the combination of available leverages?
Contributions

Chapters:

- A definition of a leverage, and a first classification of usually available leverages in a computing facility: Chapter 2
- A definition of a methodology to evaluate and model a leverage: Chapter 3
- Application of this methodology on a leverage from the literature: the shutdown leverage: Chapter 4
- A solution to combine and use multiple leverages at the same time to answer chosen constraints while being energy efficient: Chapter 5
- Generic software framework formalizing the combination of leverages and extraction of knowledge from the table of leverages: Chapter 6
Outline

Discover and model a leverage: a methodology

Methodology applied to the shutdown leverage

Combine multiple energy and power leverages

Conclusion and perspectives
Discover and model a leverage: a methodology

Methodology applied to the shutdown leverage

Combine multiple energy and power leverages

Conclusion and perspectives
A methodology to study a leverage

A methodology to study a leverage

- A step by step methodology
- How it works and operates
- Estimate usage as an energy and power leverage
Stage 1: How a leverage operates

- Understand how it works
- Detecting all states
- Detecting how to go from one state to the other
- Done through an exploration of the studied infrastructure
Stage 2: The influence on a studied metric and monitoring

- Influence of operating on a given state
- Influence of changing the current state on a given metric
- Evaluates the real cost of states and transitions for the given metric in a given context
A methodology to study a leverage: Stage 3

Stage 3: Providing actors

- Actor: entity that makes a choice concerning $s_c$ of leverage $L$
- Answers if a state is beneficial to the studied metric
- Answers if a state helps answer a constraint
- Takes into account transition and state costs
Actor usage

Actor aim
At given time $T$, an actor aims at
- Answering whether the leverage can switch state
- While respecting imposed constraints
- While improving studied metric

Actor scope
Could be used at different scale
- On one device
- On a sub-set of devices
- On all devices
The methodology, lessons learned

The methodology

- Understand and evaluate a leverage and its underlying costs
- Clear answer to changing the state of a leverage
- A "à la carte" usage of a leverage
- Applied to leverages in our publications (TEG, Shutdown, OpenMP, Version of code, MPI, Computation precision, Scheduling policies)
Outline

Discover and model a leverage: a methodology

Methodology applied to the shutdown leverage

Combine multiple energy and power leverages

Conclusion and perspectives
A methodology to study a leverage

The shutdown leverage

- One of the most promising leverage
- Non-proportional computing units
- Over provisioning of infrastructures
- Non negligible energy consumption when idle
Real experiments and calibrations

Grid’5000

- Large-scale and versatile testbed
- Experiment-driven research in all areas of computer science
- High heterogeneity in 9 different sites
- Fine grain trace (every Watt consumed every second)
- Three different nodes used: Taurus, Orion, Paravance (Rennes)
Real experiments and calibrations

<table>
<thead>
<tr>
<th>Features</th>
<th>Orion</th>
<th>Taurus</th>
<th>Paravance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server model</td>
<td>Dell PowerEdge R720</td>
<td>Dell PowerEdge R720</td>
<td>Dell PowerEdge R630</td>
</tr>
<tr>
<td>CPU model</td>
<td>Intel Xeon E5-2630</td>
<td>Intel Xeon E5-2630</td>
<td>Intel Xeon E5-2630v3</td>
</tr>
<tr>
<td># of CPU</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Cores per CPU</td>
<td>6</td>
<td>6</td>
<td>8</td>
</tr>
<tr>
<td>Memory (GB)</td>
<td>32</td>
<td>32</td>
<td>128</td>
</tr>
<tr>
<td>Storage (GB)</td>
<td>2 x 300 (HDD)</td>
<td>2 x 300 (HDD)</td>
<td>2 x 600 (HDD)</td>
</tr>
<tr>
<td>GPU</td>
<td>Nvidia Tesla M2075</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Stage 1: How a leverage operates, the shutdown leverage

How the shutdown leverage operates: the states and transitions

- Available sleep states on a computing node
- Pass by the Idle state to go to a sleep state
- Every transition has a cost
- S5 or Off state
Stage 2: Influence on metrics, the shutdown leverage

How the shutdown leverage operates: the state and transition costs

- **Energy**: non negligible budget
- **Time**: delay caused by transitions
- **Power**: multiple picks and high disturbance
The monitoring of a leverage, the shutdown leverage

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Orion</th>
<th>Taurus</th>
<th>Paravance</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E_{\text{OffIdle}}$ (J)</td>
<td>23386</td>
<td>19000</td>
<td>19893</td>
</tr>
<tr>
<td>$E_{\text{IdleOff}}$ (J)</td>
<td>775.79</td>
<td>616.08</td>
<td>1115</td>
</tr>
<tr>
<td>$T_{\text{OffIdle}}$ (s)</td>
<td>150</td>
<td>150</td>
<td>167.5</td>
</tr>
<tr>
<td>$T_{\text{IdleOff}}$ (s)</td>
<td>6.1</td>
<td>6.1</td>
<td>13</td>
</tr>
<tr>
<td>$P_{\text{idle}}$ (W)</td>
<td>135</td>
<td>95</td>
<td>150</td>
</tr>
<tr>
<td>$P_{\text{off}}$ (W)</td>
<td>18.5</td>
<td>8.5</td>
<td>4.5</td>
</tr>
</tbody>
</table>

How the shutdown leverage operates: the state and transition costs

- Focus on the S5 (Off) state
- Monitoring of three different servers
- Low standard deviation (7% in worst case)
Stage 3: Providing actors, the shutdown leverage

**Basic actors**

Used by most papers in the literature
- No-OnOff: the nodes are never shut down
- LB-ZeroCost-OnOff: no cost to shut down or wake up nodes

**Sequence-aware actors**

Make sure that the transitions costs:
- SAT: Time constrained, fits in time
- SAE: Energy constrained, beneficial in energy

**Power-capping-aware actors**

Aims at maintaining an average power budget
- $PC_{\text{Min}}$: lower limit for power usage
- $PC_{\text{Max}}$: upper limit for power usage
Simulation setup

Simulation input

- Extracted traces (Jobs, energy consumption)
- Real calibration

Simulation hypothesis

- Homogeneous datacenter
- Node reservation

Extracted metrics

- On servers lifetime: Number of On/Off cycles per policy
- On energy consumption: percentage of gained energy per actor
Simulation: LB-ZeroCost-OnOff, Seq-Aw-T and Seq-Aw-E actors

<table>
<thead>
<tr>
<th>Actor</th>
<th>Energy (Giga J)</th>
<th># cycles</th>
<th>% e. Saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-OnOff</td>
<td>6.0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>LB-ZeroCost-OnOff</td>
<td>3.9</td>
<td>1794</td>
<td>34.52</td>
</tr>
<tr>
<td>Seq-Aw-T</td>
<td>4.0</td>
<td>964</td>
<td>33.99</td>
</tr>
<tr>
<td>Seq-Aw-E</td>
<td>4.0</td>
<td>844</td>
<td>34.00</td>
</tr>
</tbody>
</table>

*Grid’5000 trace, 1 week*
Simulation: Power-Cap actors

<table>
<thead>
<tr>
<th>Actor</th>
<th>Energy (Giga J)</th>
<th># cycles</th>
<th>% e.Saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>No-OnOff</td>
<td>6.0</td>
<td>0</td>
<td>0.0</td>
</tr>
<tr>
<td>Seq-Aw-T</td>
<td>4.0</td>
<td>964</td>
<td>33.99</td>
</tr>
<tr>
<td>Power-Cap 2000 min</td>
<td>4.4</td>
<td>855</td>
<td>27.65</td>
</tr>
<tr>
<td>Power-Cap 4000 min</td>
<td>4.5</td>
<td>761</td>
<td>24.49</td>
</tr>
<tr>
<td>Power-Cap 6000 min</td>
<td>5.0</td>
<td>617</td>
<td>16.82</td>
</tr>
</tbody>
</table>
The shutdown, lessons learned

Larger scale experiments

- Traces from E-Biothon supercomputer (1.5 years)
- Traces from Grid’5000 (6 years)
- Up to 43% of energy saved

Larger set of actors

- Electricity aware
- Cooling system aware
- Renewable energy aware
- Analysis of combination of actors

The methodology applied to the shutdown leverage

- Shutdown is an energy and power leverage
- Large possibility of usage, one simulated
- Proposed actors can help to be energy efficiency
- Generic actors that can be adapted to every device that can be shut down and waked up
Outline

Discover and model a leverage: a methodology

Methodology applied to the shutdown leverage

Combine multiple energy and power leverages

Conclusion and perspectives
Large variability:

- Lot of leverage families, lot of leverages per family
- Literature usually explores one leverage at a time
- Making it complicated to reach energy efficiency at large scale

A generic solution is needed!

Our proposition: the table of leverages

- A score table
- Various users
- Extraction of energy efficient hints
Large variability:

- Lot of leverage families, lot of leverages per family
- Literature usually explores one leverage at a time
- Making it complicated to reach energy efficiency at large scale

A generic solution is needed!

Our proposition: the table of leverages

- A score table
- Various users
- Extraction of energy efficient hints
Problems and contributions

Problems

▶ How to discover, benchmark and orchestrate leverages?
▶ How to combine and evaluate leverages?

Contributions

▶ Definition of the table of leverages
▶ Generic framework formalizing the combination of leverages
▶ Experimental method based on benchmarks and monitoring to build the table of leverages
▶ Tools to extract knowledge from the table
Formalism of the construction of table of leverages: 3 basic blocks

Metrics

- Focus of the user
- Multiple occurrences
- Example: energy and power related metric

Benchmarks

- Self-contained application or portion of code
- Representative of a real application
- Example: CPU intensive, gemm kernels

Leverages

- A description of the set of states
- An iterator to go from one state to the other
- Example: three leverages, different families
The architecture of the framework

Highly expandable

- Either benchmarks, metrics and leverages
- Interfaces act as contract
A user workflow of the framework: the *construct()* function

- **Discover**
  - Leverages Input
  - Benchmark Monitoring Phase
    - Init all current states (Sc)
    - Start monitoring
    - Execute benchmark
    - End monitoring
    - Next unseen state
    - GOTO Start monitoring
  - TableOfLeverages
- **Normalize**
  - (b)
- **Analyze**
  - (c)

- Blue: user input
- Green: output of the framework
- Black: internal framework transitions
- (a) does not launch if an error is detected
- (b) checks the metrics validity
Illustration leverages: application and middleware level

Computation precision leverage

- Exploit various computation precision
- Denoted $Prec.$, set of states is \{int, float, double\}
- For each of these states, a compilation flag is modified

Vectorization leverage

- Exploit inter-core parallelism
- Denoted $Vect.$, set of states is \{none, SSE3, AVX2\}
- For each of these states, a compilation flag is modified

Multi-thread leverage

- Used to exploit intra node parallelism (OpenMP)
- Denoted $\#Threads$, the set of states is \{1, \ldots, n\}
- For each of these states, we modify a global variable
## Gemm energy and power table of leverages, Nova nodes

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>#Threads</th>
<th>Prec.</th>
<th>Vect.</th>
<th>avrgWatt</th>
<th>Joules</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>int</td>
<td>none</td>
<td>1.43</td>
<td>13.48</td>
<td>9.44</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>int</td>
<td>SSE3</td>
<td>1.4</td>
<td>4.68</td>
<td>3.33</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>int</td>
<td>AVX2</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>float</td>
<td>none</td>
<td>1.45</td>
<td>7.4</td>
<td>5.11</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>float</td>
<td>SSE3</td>
<td>1.41</td>
<td>3.76</td>
<td>2.67</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>float</td>
<td>AVX2</td>
<td>1.56</td>
<td>3.11</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>double</td>
<td>none</td>
<td>1.53</td>
<td>8.34</td>
<td>5.44</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>double</td>
<td>SSE3</td>
<td>1.53</td>
<td>8.52</td>
<td>5.56</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>double</td>
<td>AVX2</td>
<td>1.54</td>
<td>7.0</td>
<td>4.56</td>
</tr>
</tbody>
</table>
# Gemm energy and power table of leverages, Nova nodes

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>avrgWatt</th>
<th>Joules</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>#Threads</strong></td>
<td><strong>Prec.</strong></td>
<td><strong>Vect.</strong></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>int</td>
<td>none</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
<td>SSE3</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
<td>AVX2</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>none</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>SSE3</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>AVX2</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>none</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>SSE3</td>
<td>1.07</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>AVX2</td>
<td>1.06</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
<td>none</td>
<td>1.43</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
<td>SSE3</td>
<td>1.4</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
<td>AVX2</td>
<td>1.0</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
<td>none</td>
<td>1.45</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
<td>SSE3</td>
<td>1.41</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
<td>AVX2</td>
<td>1.56</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
<td>none</td>
<td>1.53</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
<td>SSE3</td>
<td>1.53</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
<td>AVX2</td>
<td>1.54</td>
</tr>
</tbody>
</table>
### Gemm energy and power table of leverages, Nova nodes

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>avrgWatt</th>
<th>Joules</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Threads</td>
<td>Prec.</td>
<td>Vect.</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>int</td>
<td>none</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
<td>SSE3</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
<td>AVX2</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>none</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>SSE3</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>AVX2</td>
<td>1.05</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>none</td>
<td>1.06</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>SSE3</td>
<td>1.07</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>AVX2</td>
<td>1.06</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
<td>none</td>
<td>1.43</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
<td>SSE3</td>
<td>1.4</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
<td>AVX2</td>
<td>1.0</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
<td>none</td>
<td>1.45</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
<td>SSE3</td>
<td>1.41</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
<td>AVX2</td>
<td>1.56</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
<td>none</td>
<td>1.53</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
<td>SSE3</td>
<td>1.53</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
<td>AVX2</td>
<td>1.54</td>
</tr>
</tbody>
</table>
## Gemm energy and power table of leverages, Nova nodes

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>#Threads</th>
<th>Prec.</th>
<th>Vect.</th>
<th>avrgWatt</th>
<th>Joules</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>int</td>
<td>none</td>
<td>1.05</td>
<td>65.09</td>
<td>61.89</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>int</td>
<td>SSE3</td>
<td>1.06</td>
<td>28.26</td>
<td>26.56</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>int</td>
<td>AVX2</td>
<td>1.06</td>
<td>29.32</td>
<td>27.67</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>float</td>
<td>none</td>
<td>1.05</td>
<td>72.97</td>
<td>69.67</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>float</td>
<td>SSE3</td>
<td>1.06</td>
<td>33.8</td>
<td>31.89</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>float</td>
<td>AVX2</td>
<td>1.05</td>
<td>36.8</td>
<td>34.89</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>double</td>
<td>none</td>
<td>1.06</td>
<td>81.59</td>
<td>76.89</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>double</td>
<td>SSE3</td>
<td>1.07</td>
<td>58.52</td>
<td>54.89</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>double</td>
<td>AVX2</td>
<td>1.06</td>
<td>57.72</td>
<td>54.22</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>int</td>
<td>none</td>
<td>1.43</td>
<td>13.48</td>
<td>9.44</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>int</td>
<td>SSE3</td>
<td>1.40</td>
<td>4.68</td>
<td>3.33</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>int</td>
<td>AVX2</td>
<td>1.00</td>
<td>1.00</td>
<td>1.0</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>float</td>
<td>none</td>
<td>1.45</td>
<td>7.4</td>
<td>5.11</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>float</td>
<td>SSE3</td>
<td>1.41</td>
<td>3.76</td>
<td>2.67</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>float</td>
<td>AVX2</td>
<td>1.56</td>
<td>3.11</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>double</td>
<td>none</td>
<td>1.53</td>
<td>8.34</td>
<td>5.44</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>double</td>
<td>SSE3</td>
<td>1.53</td>
<td>8.52</td>
<td>5.56</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>double</td>
<td>AVX2</td>
<td>1.54</td>
<td>7.0</td>
<td>4.56</td>
</tr>
</tbody>
</table>
The table of leverage, layer by layer: no vectorisation focus

<table>
<thead>
<tr>
<th>#Threads</th>
<th>Prec.</th>
<th>Vect.</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>int</td>
<td>none</td>
<td>65.09</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
<td>none</td>
<td>72.97</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
<td>none</td>
<td>81.59</td>
</tr>
</tbody>
</table>

Focus

- Joules metric
- *None* state for Vectorization
- 1 as state for #Threads leverage
- Score for the *Precision* leverage: int, float, double
The table of leverage, layer by layer: no vectorisation focus

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>#Threads</td>
<td>Prec.</td>
</tr>
<tr>
<td>32</td>
<td>int</td>
</tr>
<tr>
<td>32</td>
<td>float</td>
</tr>
<tr>
<td>32</td>
<td>double</td>
</tr>
</tbody>
</table>

Focus

- Joules metric
- *None* state for Vectorization
- 32 as state for #Threads leverage
- Score for *Precision* states: float, double, int
- Noticeable change in the scoring!
The table of leverage, layer by layer: mono core focus

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Threads</td>
<td>Prec.</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
</tr>
<tr>
<td>1</td>
<td>int</td>
</tr>
</tbody>
</table>

Focus

- Joules metric (again)
- `int` state for Precision
- 1 as state for #Threads leverage
- Score for *vectorization* leverage: SSE3, AVX2, none
The table of leverage, layer by layer: mono core focus

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Threads</td>
<td>Prec.</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
</tr>
<tr>
<td>1</td>
<td>float</td>
</tr>
</tbody>
</table>

Focus

- Joules metric (again)
- *float* state for Precision
- 1 as state for #Threads leverage
- Same score for *vectorization* leverage: SSE3, AVX2, none
The table of leverage, layer by layer: mono core focus

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Threads</td>
<td>Prec.</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
</tr>
</tbody>
</table>

Focus

- Joules metric (again)
- *double* state for Precision
- 1 as state for #Threads leverage
- Score for *vectorization* leverage: AVX2, SSE3, none
- Noticeable change in the scoring! (again)

Observations

- A lot of insights about energy and power leverages
- Still complicated to extract knowledge from it
- How to extract knowledge from it?
The table of leverage, layer by layer: mono core focus

<table>
<thead>
<tr>
<th>Leverage states</th>
<th>Joules</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Threads</td>
<td>Prec.</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
</tr>
<tr>
<td>1</td>
<td>double</td>
</tr>
</tbody>
</table>

Focus

- Joules metric (again)
- double state for Precision
- 1 as state for #Threads leverage
- Score for vectorization leverage: AVX2, SSE3, none
- Noticeable change in the scoring! (again)

Observations

- A lot of insights about energy and power leverages
- Still complicated to extract knowledge from it
- How to extract knowledge from it?
Exploiting the table of leverage

Question
When I fix a state, do I always improve a given metric?

Formalism
Consider state $x_a$ of leverage $\chi$. We want to check whether for all $i \in [0, \ldots, n_x]\backslash\{a\}$, for all $l, j \in [0, \ldots, n_y]$, and for all $m, k \in [0, \ldots, n_z]$, we have:

$$ToL_m(x_a, y_l, z_m) \leq ToL_m(x_i, y_j, z_k).$$

For the Joules metric:

- Only \#Threads $== 32$ answers this predicate
- Thus, using this state will always be beneficial
- No specific results with other metrics
Exploiting the table of leverage

**Question**
If some states are fixed for a subset of leverages, is a given state for the remaining leverages the best choice to optimize a given metric?

**Formalism**
Consider that the state of leverages $\psi, \omega$ is fixed to $y_b, z_c$. We are asking whether state $x_a$ of leverage $\chi$ is the best choice for metric $ToL_m$. Therefore, we need to check whether for all $i \in [0, \ldots, n_x]\{a\}$, we have:

$$ToL_m(x_a, y_b, z_c) \leq ToL_m(x_i, y_b, z_c),$$

For the fixed combination $\{32, \text{SSE3}\}$:

- Joules or Time: the best state for the *Precision* leverage is *float*
- AvrgWatt: the best state for the *Precision* metric is *int*
Large scale usage of Table of Leverages

Realistic set-up

- Modular constraints
- Production application (FullSWOF2D)
- Two infrastructures: Grid’5000 and Curie

Proposed actor

- Builds the table
- Chooses the state for all leverages
- Respect constraints, reduce consumed energy

- Three Leverages (#Processes, #Threads, CodeVersion)
- Proposition of a new leverage
- Evaluation of proposed leverage and actor
Large scale usage of Table of Leverages

Results

- Grid’5000 4 nodes, Curie 128 nodes
- Up to 39.81% of energy savings
Leverage combination: Conclusions

A framework:

- Implements the combination of leverages
- Ease the discovery and understanding of leverages
- Generic and highly expendable
- Ease the study and combination of leverages through the construction of the table of leverages
- Ease the hints extraction from the table of leverages
- 30k lines of Python code

Perspectives:

- Explore other phases
- Automatic re-usability validation exploration
- Include user acceptance
Outline

Discover and model a leverage: a methodology

Methodology applied to the shutdown leverage

Combine multiple energy and power leverages

Conclusion and perspectives
Contributions

- Definition of a leverage, an energy and power leverage
- First classification of usually available leverages in a computing facility
- A methodology to evaluate and model a leverage
- Methodology applied on leverages from the literature
- A methodology to combine and use multiple leverages at the same time to answer chosen constraints
- GreenFactory: Generic software framework formalizing the combination of leverages and extraction of knowledge from the table of leverages
Perspectives

Short term

- Explore other leverages
- Reducing the search space for table
- Support sub-application leverages

Long term

- Categorize uncommon leverages
- Table of leverages for every phase
- Generic actors
- GreenFactory out of the computing facility (Fog, IoT)
Thank you

International Journals

- **IJHPCA**, João Vicente Ferreira Lima, Issam Raïs, Laurent Lefèvre, Thierry Gautier, 2018
- **CCPE**, Issam Raïs, Anne-Cécile Orgerie, Martin Quison and Laurent Lefèvre, 2018
- **IJHPCA**, Anne Benoit, Laurent Lefèvre, Anne-Cécile Orgerie, and Issam Rais, 2017

International Conferences

- **ICA3PP**, Issam Raïs, Laurent Lefèvre, Anne-Cécile Orgerie, Anne Benoit, 2018
- **HPCS**, Issam Raïs, Mathilde Boutigny, Laurent Lefèvre, Anne-Cécile Orgerie, Anne Benoît, 2018
- **CCGRID**, Pierre-François Dutot, Yiannis Georgiou, David Glesser, Laurent Lefèvre, Millian Poquet, and Issam Rais, 2017
- **Euro-Par**, Anne Benoit, Laurent Lefèvre, Anne-Cécile Orgerie and Raïs, Issam, 2017
- **ICA3PP**, Issam Raïs, Anne-Cécile Orgerie, and Martin Quinson, 2016

International Workshops

- **SBAC-PAD**, João Lima, Issam Rais, Laurent Lefèvre, Thierry Gautier, 2017
- **HPCS**, Issam Rais, Laurent Lefèvre, Anne Benoit, and Anne-Cécile Orgerie 2016

Han, Jie and Michael Orshansky. “Approximate computing: An emerging paradigm for energy-efficient design”. In: *Test Symposium (ETS), 2013 18th IEEE European*. IEEE. 2013, pp. 1–6.


Re-usability of studied metrics for one node

(a) CPU stress
(b) IO stress
(c) HDD stress
(d) RAM stress

Figure: Nova-1, 30 runs of various stresses for Time (seconds) and Power (Watts)
Re-usability of studied metrics for multiple nodes

<table>
<thead>
<tr>
<th>Hardware family</th>
<th>Joules (J)</th>
<th>AvrgWatt(W)</th>
<th>Time(t)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Av. - StD.</td>
<td>Av. - StD.</td>
<td>Av. - StD.</td>
</tr>
<tr>
<td>CPU</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taurus</td>
<td>6807.0 - 68.8</td>
<td>205.84 - 1.37</td>
<td>32.81 - 0.39</td>
</tr>
<tr>
<td>Nova</td>
<td>4998.86 - 49.3</td>
<td>154.91 - 1.09</td>
<td>32.06 - 0.43</td>
</tr>
<tr>
<td>HDD</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taurus</td>
<td>5055.98 - 365.33</td>
<td>140.58 - 2.98</td>
<td>35.85 - 2.4</td>
</tr>
<tr>
<td>Nova</td>
<td>9381.94 - 251.5</td>
<td>107.8 - 0.57</td>
<td>87.01 - 2.47</td>
</tr>
<tr>
<td>IO</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taurus</td>
<td>3957.52 - 34.98</td>
<td>123.46 - 0.21</td>
<td>32.0 - 0.3</td>
</tr>
<tr>
<td>Nova</td>
<td>4194.53 - 68.06</td>
<td>130.3 - 0.67</td>
<td>32.04 - 0.66</td>
</tr>
<tr>
<td>RAM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Taurus</td>
<td>5097.83 - 55.81</td>
<td>222.14 - 2.2</td>
<td>32.5 - 0.52</td>
</tr>
<tr>
<td>Nova</td>
<td>7282.26 - 115.89</td>
<td>158.53 - 0.8</td>
<td>31.93 - 0.44</td>
</tr>
</tbody>
</table>

The context

- Average and standard deviation
- 10 Taurus, 5 Nova nodes
- 10 runs