Accurate and Stable Empirical CPU Power Modelling for Multi- and Many-Core Systems

Matthew J. Walker*, Stephan Diestelhorst†, Geoff V. Merrett* and Bashir M. Al-Hashimi*

*University of Southampton
†Arm Ltd.
Motivation: **Run-Time Management (RTM)**

- Run-time control of energy-saving techniques, e.g. **DVFS, DPM**,  
  - Heterogeneous Multi-Processing (HMP) - **Arm big.LITTLE**  
- Trade-off power and performance  
- Improving **energy-efficiency**  
- Maximising peak performance, while respecting **thermal** and **power** limits  
- Lifetime reliability
Motivation: Simple Example

- Power Management + Scheduling must be considered together
  - Energy-Aware Scheduling (EAS) in Linux [1]
  - Uses power model to drive scheduling
- Arm DynamIQ
  - Next generation HMP big.LITTLE
  - A cluster can contain *big and little* simultaneously
  - Supports multiple power domains in the same cluster

More energy-saving opportunities….
…requires more complex RTM to exploit

Multi- and Many-Core Power Modelling

Run workloads

Hardkernel ODROID-XU3

PMCs (Performance Counters)

Power (and voltage)

Linear equations - Ordinary Least Squares estimator

Key Property:
Accurate estimations across a diverse set of workload phases, even if they are not represented in the training set

Originally intended for run-time energy management

- Very accurate
- Only valid for the profiled platform
Performance Monitoring Counters (PMCs)

On many mobile, accessing PMCs is not straightforward

Our method:
• Reads from the PMU (performance monitoring unit) registers directly - no *perf*!
• First, need to enable access to them from *userspace* - LKM to modify USER ENable register.
• Perf not required
• Doesn’t rely on working interrupts
• Doesn’t reset counters - multiple applications can use them simultaneously

Reading PMCs on XU3 + building power models: powmon.ecs.soton.ac.uk
New PMC logging: gemstone.ecs.soton.ac.uk
Model Development Methodology

1. PMC Event Selection:
Identify optimum events using classification techniques

Aim: events that give the most amount of unique information useful for predicting power.

(Make transformations to further reduce multicollinearity)

Stepwise-regression

Hierarchical Cluster Analysis

2. Model Formulation and Validation:
Separates high-level components

\[
P_{\text{cluster}} = \left( \sum_{n=0}^{N-1} \beta_n E_n V^2 f_{\text{clk}} \right) + \beta V^2 f_{\text{clk}} + f(V, T)
\]

BG dynamic
dynamic activity
static

1. Correct Model Specification
2. Consider heteroscedasticity
3. Effects of temperature
4. Non-ideal voltage regulation
Coefficient Stability

• Critical to achieving a stable models:
  1. Diverse observations (e.g. diverse workloads)
  2. Carefully chosen model inputs (e.g. PMC events) - no multicollinearity

• We will show how the “stability” of the model is more important than the reported average error

• We will show how a model can have a good apparent accuracy but perform poorly when faced with diverse workloads, and how a stable model is able to remain accurate across a diverse range of scenarios.
‘Unstable’ vs. ‘Stable’ Selection

Models trained on X workloads and tested on Y workloads (X | Y)

F = Full workload set (60)
S.T = Small typical (e.g. MiBench) workload set (20)
S.R = Small random (diverse) workload set (20)

Feature Selection

- Hierarchical Cluster Analysis (HCA) + Correlation with power
- p-values and Variance Inflation Factor (VIF)
- Forward stepwise selection
- Using VIF to apply linear transformations

\[ VIF = \frac{1}{1 - R^2} \]
What is the model formulation?

$$P = const + \beta_0 Frequency + \beta_1 Voltage + \beta_2 E_0 + \beta_3 E_1 + \beta_4 E_2 + \ldots$$

Typical regression-based power model formulation [1-4]

Not like this!
Relationships have not been captured
CPU Idle.. etc. give same information as PMCs!

Wikipedia says:

$$P_{cpu} = P_{dyn} + P_{sc} + P_{leak}$$
$$P_{dyn} = CV^2 f$$

Chosen Equation

• Breaks down dynamic and idle power

• Time to run experiment:
  • frequencies * different core utilisations * workloads * average workload time

• Therefore, run all workloads at a **single frequency** and just one workload (i.e. sleep) at all of the frequencies

• Effects of temperature “absorbed”

\[
P_{\text{cluster}} = \left( \sum_{n=0}^{N-1} \beta_n E_n V_{DD}^2 f_{clk} \right) + f(V_{DD}, f_{clk})
\]

dynamic activity

static and BG dynamic

<table>
<thead>
<tr>
<th></th>
<th>Avg. Error (%)</th>
<th>Experiment Time (hours)</th>
<th>Workloads</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow</td>
<td>2.8</td>
<td>40</td>
<td>60</td>
</tr>
<tr>
<td>Fast</td>
<td>3.4</td>
<td>0.42 (25 min.)</td>
<td>30</td>
</tr>
</tbody>
</table>

Using stability to reduce workloads
Splitting idle and dynamic activity

Error for ‘fast’ calculated by testing on 40 hour data
## Chosen Equation

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Weight</th>
<th>95% Confidence Interval</th>
<th>P-Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>7.526e+2</td>
<td>-8.858e+2 to -6.193e+2</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x11:Frequency_A15:Voltage_A15_Squared</td>
<td>5.721e-10</td>
<td>5.548e-10 to 5.895e-10</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x1b_minus_EPH_0x73:Frequency_A15:Voltage_A15_Squared</td>
<td>7.297e-10</td>
<td>6.935e-10 to 7.659e-10</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x50:Frequency_A15:Voltage_A15_Squared</td>
<td>8.115e-9</td>
<td>7.395e-9 to 8.835e-9</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x6a:Frequency_A15:Voltage_A15_Squared</td>
<td>1.606e-8</td>
<td>1.462e-8 to 1.749e-8</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x73:Frequency_A15:Voltage_A15_Squared</td>
<td>8.574e-11</td>
<td>6.271e-11 to 1.088e-10</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x14:Frequency_A15:Voltage_A15_Squared</td>
<td>1.083e-9</td>
<td>9.974e-10 to 1.168e-9</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>EPH_0x19:Frequency_A15:Voltage_A15_Squared</td>
<td>2.505e-9</td>
<td>2.220e-9 to 2.790e-9</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Frequency_A15</td>
<td>1.516e-1</td>
<td>1.161e-1 to 1.870e-1</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Voltage_A15</td>
<td>2.506e+3</td>
<td>2.068e+3 to 2.944e+3</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Frequency_A15:Voltage_A15</td>
<td>-6.025e-1</td>
<td>-7.273e-1 to -4.778e-1</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Voltage_A15_Squared</td>
<td>-2.774e+3</td>
<td>-3.253e+3 to -2.295e+3</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Frequency_A15:Voltage_A15_Squared</td>
<td>7.650e-1</td>
<td>6.182e-1 to 9.118e-1</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Voltage_A15:Voltage_A15_Squared</td>
<td>1.021e+3</td>
<td>8.468e+2 to 1.195e+3</td>
<td>p &lt; .0001</td>
</tr>
<tr>
<td>Frequency_A15:Voltage_A15:Voltage_A15_Squared</td>
<td>-3.140e-1</td>
<td>-3.713e-1 to -2.567e-1</td>
<td>p &lt; .0001</td>
</tr>
</tbody>
</table>

Cortex-A15 MAPE: 2.8%
Deduce *how* power is consumed

Predicted power and modelled power for 30 different workloads
Deduce how power is consumed – dynamic activity

Breakdown of estimated dynamic power for six different workloads

- 0x11: Cycle Count
- 0x50 – L2D Cache Load
- 0x6A – Unaligned Load/Store Spec. Exec.
- 0x73 – Integer Instr. Sepc. Exec.
- 0x14 – L1 Instruction Cache Access
- 0x19 – Bus Cycle
Comparison with Existing Work

Example of how a model built with our stable approach achieves a low average error and narrow error distribution compared to existing techniques.

Models trained with 20 workloads, validated with 60.
Heteroscedasticity

Assumptions of linear regression must be respected, including:

• No multicollinearity
• Correct model specification
• No Heteroscedasticity

Inherent to CPU power power modelling
E.g. food expenditure, annual income with wage

Affects standard error estimates

We use robust standard error estimates
(HC3)
1. Take a reference system model
2. Apply the idea
3. Compare the performance and energy between the before and after case

Questions:
- Are the models representative?
- Does the model respond to my change in a representative way?
- How much do the errors influence the conclusion?
Hardware-Validated gem5 Models + Empirical Power Models

1. Compare HW and gem5 Models

2. Use ML techniques to identify and understand sources of error

3. Apply empirical power models

4. Evaluate Scaling between HMP cores and DVFS levels
GemStone

Five Open-Source Software Tools:

1. GemStone Profiler-Logger Records PMCs with low overhead from any Arm dev board (ARMv7 and ARMv8)

2. GemStone Profiler-Automate Automates the running of experiments on a hardware platform and conducts post-processing (workloads, frequencies, core masks, PMC events, multiple iterations)

3. GemStone Gem5 Auto Automates the running of identical experiments on gem5, batch

4. GemStone Gem5-Validate Combines gem5 and HW data, uses statistical + ML techniques to evaluate errors

5. GemStone ApplyPower Applies power models to both HW and gem5 stats. Also creates equations for gem5 power framework. + performance, power and energy scaling

Online Results Visualiser + Tutorials

gemstone.ecs.soton.ac.uk
Video demo...

• (see http://gemstone.ecs.soton.ac.uk/gemstone-website/gemstone/results-viewer-gs-results.html)
Hardware-Validation Conclusion

Enables gem5 models to be:

- Improved;
- Extended to other CPUs;
- Validated after changes;
- Applicability tested for specific use-cases.

Implemented and evaluated power models with gem5 models

Before | After
--- | ---
MAPE | 59 % | 18 %
MPE | -51 % | +10 %
Conclusion

• Newer systems have larger numbers of HMP cores - need RTM and power models to exploit efficiently

• Accurate and stable run-time power models [1]
  • Feature selection for stable coefficients
  • Appropriate model specification
  • Heteroscedasticity
  • Temperature compensation [2]
  • Non-Ideal Voltage Regulation

• Performance and Energy modelling in gem5 [3]
  • Identifying sources of error in performance simulator
  • Integrating and evaluating power models


Powmon: http://www.powmon.ecs.soton.ac.uk
Gemstone: http://gemstone.ecs.soton.ac.uk
Questions?