Hardware-Validated CPU Performance and Energy Modelling

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

*University of Southampton
Southampton, UK
{mw9g09, gvm, bmah}@ecs.soton.ac.uk

†Arm Ltd.
Cambridge, UK
firstname.surname@arm.com

***Abstract*—Full-system simulation frameworks such as gem5 are used extensively to evaluate research ideas and for design-space exploration. Moreover, energy-efficiency has become the key design constraint in recent years and many works use a separate power modelling framework to evaluate energy consumption. While such tools are convenient and flexible, they are known to contain sources of error which are often not fully understood and potentially impact the conclusions drawn from investigations. This work enables accurate, hardware-validated performance, power, and energy modelling of CPUs by first presenting a methodology to evaluate and identify sources of error in CPU performance models, and secondly developing empirical power models optimised for use with such performance models. Hierarchical clustering, correlation analysis, and regression techniques are used to identify sources of error without requiring detailed CPU specifications and enable existing models to be improved, new models to be developed, validation of simulator changes, and testing of model suitability for specific use-cases. Furthermore, the GemStone open-source software tool is presented, which automates the process of characterising hardware platforms, identifying sources of error in gem5 models, applying power analysis, and quantifying the effect of errors on the performance, power, and energy estimations. In addition, the mean percentage error in execution time was found to swing from -51% to $+10\%$ between two versions of the same gem5 model, underlining the need for an automated tool to validate models against reference hardware, ensuring accuracy and consistency.**

I. Introduction

Architectural and micro-architectural CPU simulators are heavily relied upon in both academia and industry to evaluate new ideas and proposals. Due to its active development community and support for many Instruction-Set Architectures (ISAs), including x86 and ARM, gem5 [1] has become a widely used simulation framework.

In recent years, energy efficiency has become a primary design constraint in modern computer systems, and architectural simulators are often coupled with an energy simulation tool. For example, McPAT [2] is a widely used power, area and timing simulation tool that is often used with gem5 to provide energy analysis.

While such simulation tools are invaluable to research, they inherently contain errors which can impact the conclusions drawn from research, particularly if the sources of error are not well understood. This potentially affects the quality and integrity of research that relies on the tools. Recent works have focussed on these errors and their effects [3], [4], [5], [6].

When evaluating research ideas or conducting design-space exploration, accurate performance and power reference models are key to ensure representative results and correct conclusions. For example, a common use-case of a full-system simulator is evaluating a proposal for a specific part of the system (e.g., the out-of-order scheduling, branch predictor, L1I cache size, etc.). To do this, a baseline model is used, the changes applied, and the differences measured and evaluated. A reference model based on a typical system is therefore an important component of a simulation framework. If there are significant errors in the reference model, it may not respond in a representative way to the change under test.

Performance Monitoring Counters (PMCs), which count architectural and micro-architectural events in the CPU, can be used with empirical data to create accurate power models [7], [8] that are typically used to provide run-time power estimations to an Operating System (OS). Such models are less flexible than simulation tools (e.g. Wattch [9], McPAT) rendering them unsuitable for some applications; however, their accuracy and implicit hardware-validation makes them ideal for reference models.

This work augments the gem5 simulation framework with accurate, hardware-validated empirical PMC-based power models of a real hardware platform. However, through extensive experimental evaluation, significant errors in the workload execution time are found when validating existing gem5 models on a wide selection of workloads. Furthermore, analysis of individual event statistics from the model (e.g. L1D cache misses), which are required as inputs to power modelling tools (including both McPAT and the PMC-based power models), have even larger errors. A key cause is specification error, which explains errors caused by incorrectly setting model parameters due to a lack of information about the device being modelled [10], [5], [11].

To address this problem, a comprehensive methodology for systematically evaluating CPU performance models against hardware platforms and identifying sources of errors is presented (Section IV), along with a corresponding software tool, *GemStone*, which automates the process. GemStone collects

data from hardware platforms; combines the data with gem5 simulation results; identifies errors in the gem5 model using statistical techniques; and evaluates performance, power and energy accuracy (as well as their scaling across frequencies) using the aforementioned power models. Although GemStone is designed to work with the ARM architecture and the gem5 simulation framework, the presented methodology is equally applicable to other architectures and simulators.

While some works have evaluated gem5, its accuracy and usefulness depends on the specific model being used and the purpose it is being used for. Furthermore, the active development community constantly improves gem5, meaning that running the same setup with two different versions of gem5 can produce different results; inevitably, bugs can also be introduced occasionally (Section VII). There is therefore a need for an automated tool that compares gem5 models against reference hardware platforms to ensure its accuracy, consistency and applicability for specific use-cases. All graphs in this work, with the exception of Fig. 4, are generated by GemStone.

By building and integrating empirical hardware-validated power models into the gem5 simulator itself, sources of error in the power simulation are reduced and the accuracy for a specific CPU is known (Section V). The result is a framework for building accurate, reliable and hardware-validated gem5 models for performance and energy evaluation. This is important as results from performance and energy simulators underpin the results and conclusions of many works of research and development; limitations in simulation tools can lead to incorrect conclusions and reduce the quality of research if they are not understood.

The key contributions of this paper are as follows:

1) a methodology for evaluating performance models and identifying specific sources of error (Section IV);
2) empirical power models of an Arm Cortex-A7 CPU and Arm Cortex-A15 CPU, optimised for gem5 events (Section V);
3) evaluation of how modelling errors affect performance, power and energy estimations, and DVFS scaling (Section VI);
4) the GemStone open-source software tool is presented, which characterises hardware platforms, evaluates gem5 models, identifies sources of error using statistical methods, and applies power and energy analysis. Software, models, datasets and full results are made available[1].

This paper is organised as follows: related works are discussed in Section II; the methodology overview is described in Section III; Section IV compares gem5 to a hardware platform and presents the method of identifying errors; the power models are developed and validated against a hardware platform in Section V; the effect of errors in the gem5 model on the performance, power and energy, and how they scale with DVFS levels, is presented in VI; and improvements to the evaluated gem5 model are discussed in Section VII.

[1]See http://gemstone.ecs.soton.ac.uk

II. Background and Related Works

Power simulators, such as Wattch and McPAT, are commonly utilised in research that requires power and energy of systems to be evaluated. However, such tools are known to contain large sources of error [3], [6], [4]. While not as flexible, empirical power models utilising hardware PMCs have been shown to provide superior accuracy and a higher level of confidence than simulators [12], [7], [8]. In many research scenarios, an accurate baseline model is required, on which to implement and evaluate research ideas. A recent work presents a methodology and corresponding software tools for creating PMC-based power models [8]. The authors demonstrate their methodology, which aims to improve model stability, with an *Exynos-5422* System-on-Chip (SoC).

Existing works have created and validated performance models of existing hardware in full-system simulation tools (e.g. gem5). Butko *et al.* [13] compare the accuracy of a gem5 model against a hardware device with a dual-core Arm Cortex-A9 CPU, find an average error of between 1.4% and 17.9% and conclude that an overly simple DRAM model is a key source of the error. Endo *et al.* [14] create gem5 models to represent Cortex-A8 and Cortex-A9 CPUs with an average error of 7%. Gutierrez *et al.* [5] analyse sources of error in full system simulation and stress the importance of simulator users understanding the limitations and considering the accuracy of the micro-architectural events as well as the execution time. They model and validate a dual-core Cortex-A15 CPU (with some complex-to-model features, such as prefetching, indirect branch predicting disabled) in gem5 and make improvements to the model to achieve an execution time Mean Absolute Percentage Error (MAPE) of 13% and 17% for SPEC2006 and dual-core PARSEC, respectively. They identify specification error to be the dominant cause of divergence between the model and the hardware platform.

Butko *et al.* [11] present an up-to-date gem5 model of an Exynos-5422 SoC. They solve issues with running big.LITTLE systems in gem5, configure model parameters using datasheets and educated guesses, and release their models to the gem5 community. They find an execution time MAPE of 20% using the Rodinia benchmarking suite and validate memory and operation latencies using *lmbench*. They also identify specification error as the key source of error, as well as a simplistic DRAM model. They then use McPAT to conduct energy analysis and find a MAPE of 25% when compared with the hardware platform.

A recent work [15] uses the Cortex-A15 power models from [8] for use in gem5, however, it does not use a detailed Cortex-A15 model, find alternatives to events that are unavailable in gem5, identify exact relationships between the hardware PMCs and gem5 events (e.g. hardware L2 data cache loads are equated to gem5 L2 cache accesses), or consider the energy error, which this work shows to be significant (Section VI). There are currently no power models integrated into the gem5 simulation framework.

This work uses gem5 models, based on the ones presented

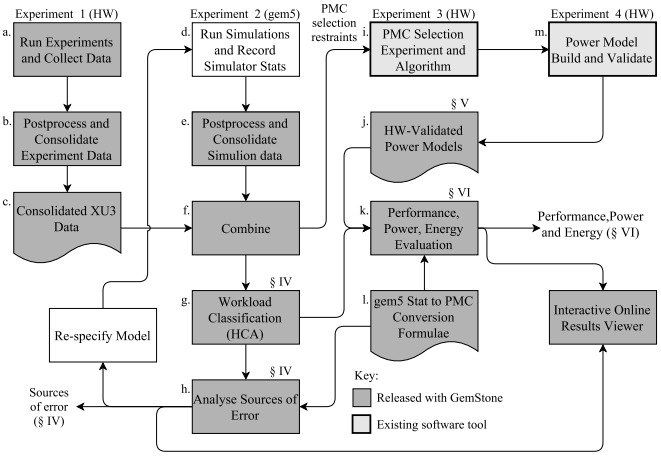


Fig. 1. Experimental setup and methodology overview

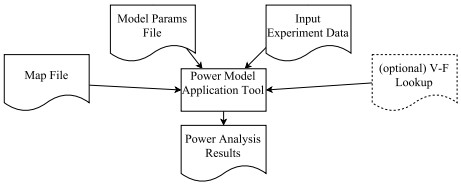


Fig. 2. Software tool for applying power models to either hardware collected data or gem5 data

in [11], upon which to implement power models developed using the method outlined in [8]. The issue of specification error in gem5 models, highlighted by [5] and [11], motivates the development of a methodology to identify sources of error without specific CPU specifications, which is presented herein. This work then develops Cortex-A7 and Cortex-A15 power models (optimised for gem5 events) and a corresponding software tool that enables energy analysis with gem5 through two methods: 1) applying the model to gem5 output files for retrospective power analysis and 2) generating equations that can be inserted directly into gem5 for run-time power analysis within gem5 itself.

III. Experimental Setup and Overview

This work is demonstrated on a Hardkernel ODROID-XU3 development board, which uses a Samsung Exynos-5422 SoC. It is the same device as used in both [8] and [11] and contains a quad-core Arm Cortex-A7 CPU cluster (optimised towards low-energy), and a quad-core Arm Cortex-A15 CPU cluster (optimised towards high-performance). This platform uses the 32-bit ARMv7 architecture. The ODROID-XU3 development board contains power sensors measuring the power consumed by each cluster. The platform was running Ubuntu 14.04 (kernel: 3.10.63).

A set of 65 workloads from several benchmarking suites were used to evaluate the gem5 models and empirical power models, including MiBench [16], ParMiBench [17], LMBench [18], Roy Longbottom's PC Benchmark Collection [19], PARSEC [20], Dhrystone [21] and Whetstone [22]. PARSEC workloads were run both with a single thread and four threads.

The overall experimental setup comprises of four key experiments for data collection (Fig. 1): 1) collection of performance and PMC data from the hardware (HW) platform for a baseline to compare the gem5 models against; 2) running of the gem5 model simulations; 3) collecting power and PMC data for every workload and PMC event (for power model PMC event selection); and 4) collecting power and PMC data (across the events selected in 3) for every frequency to build the empirical power models. Experiments 3 and 4 and the corresponding post-processing and analysis (boxes *i* and *m*) are automated by the *Powmon* software tools presented by Walker *et al.* [8]. The power sensors on the ODROID-XU3 provide readings at 3.8 Hz (the sensors internally sample at a higher frequency and provide an average). While this rate can be increased, it comes at a cost of overhead. The workloads were therefore repeated so that they exercised the CPU for at least 30 seconds to obtain accurate and repeatable power measurements.

Experiment 1 (box *a*) collects data for evaluating the gem5 models and has different requirements to Experiments 3 and 4. Single runs of MiBench, ParMiBench, PARSEC (both single threaded and multi-threaded), Dhrystone and Whetstone (45 workloads in total) were run on the ODROID-XU3 board. Each workload was run five times and the observation with the median execution time used. The experiment was repeated to capture 68 PMC events (only a limited set of PMC events can be measured simultaneously). When running at 2 GHz on the Cortex-A15 (the maximum Cortex-A15 clock frequency), throttling occurred due to high CPU temperatures. A frequency of 1.8 GHz was therefore the highest used and a 5 second delay was inserted between workloads to allow the CPU to cool down. This experiment was run at 200 MHz, 600 MHz, 1 GHz and 1.4 GHz on the Cortex-A7 and 600 MHz, 1 GHz, 1.4 GHz and 1.8 GHz on the Cortex-A15. Throughout this work the ODROID-XU3 hardware platform will be referred to as HW.

The *ex5_LITTLE.py* and *ex5_big.py* CPU models built into the gem5 simulator are used in this work for Experiment 2. They are designed to represent the Exynos-5422 SoC found in the ODROID-XU3 board and are based on the work presented by Butko *et al.* [11] (Section II). The simulator was running Ubuntu 11.04 (kernel: 4.4). The simulations were run with the same workloads and DVFS levels as Experiment 1.

The results from the gem5 experiments and the gem5 hardware validation experiments are collated and combined (box *f*), and the workloads clustered to identify patterns and errors between workload types (box *g*). Knowledge of PMC events that are not available or reliable are fed back to the PMC event selection algorithm (PMC selection restaints) so power models can be formulated with events that work well in gem5. Equivalent gem5 events are found to the PMC events chosen in the model (box *l*) and a software tool is presented to apply the power model to both the hardware collected data and the gem5

simulated data for power and energy analysis (part of box *k*). This software tool (Fig. 2) is compatible with the *Powmon* model building software [8], allowing the models created by that software to be applied to gem5 simulations or HW data. The advantage of this tool is that power models can be applied to gem5 results after the simulation, meaning that the selected power model or the voltage for a selected frequency can be changed without re-running the gem5 simulation. This tool also outputs power equations in a format that allows run-time power analysis in gem5 itself. The methodology for deriving the sources of error in gem5 (box *h*) is described in the next section and allows the gem5 models to be iteratively improved to match the hardware platform.

IV. Identifying Sources of Error in gem5

This section evaluates the existing gem5 models against the hardware (HW) platform and describes a methodology for identifying sources of error. A key problem in CPU simulation is specification error (Section II). Moreover, there are difficulties in precisely matching many HW PMCs to gem5 events as the hardware documentation is not detailed and the specifics of many events are implementation defined [23]. The methodology presented in this section applies several statistical methods to analyse relationships between modelling errors and workload types, HW PMC events, and modelled events to identify the sources of error, without requiring detailed CPU specifications or matched events (Sections IV-B, IV-C and IV-D). GemStone applies these techniques and automatically produces tables and graphs that enable error correlations to be observed and points of interest to be extracted (e.g. Figs. 3 and 5). By carefully cross-comparing these results, a user can identify causality and the key sources of error using the techniques presented in this section. Adjustments can then be made to the problem component of the gem5 model by the user, and the effects of this change evaluated by re-running the gem5 simulation and the analysis (GemStone automates this). Microbenchmarks can also be employed to target the identified component if necessary. GemStone also allows a user to write equations relating gem5 events to HW PMC events (if known) and directly compares them for a more detailed picture of deviations between the gem5 model and HW (Section IV-E). The remainder of this section uses the existing *ex5_big.py* gem5 model as an example to demonstrate the proposed methodology of evaluating the model and identifying sources of error.

For workloads from the PARSEC suite the gem5 model predicts program execution time with a MAPE of 25.5% and a Mean Percentage Error (MPE) of -7.5% across both CPU clusters and at all tested DVFS levels. A negative MPE indicates that the gem5 model underestimates performance (overestimates the execution time). However, when testing on a larger set of workloads (45 in total) from different benchmarking suites, the MAPE is 40% and a MPE is -21%, highlighting the importance of considering many diverse workloads. The Cortex-A7 model achieves a higher accuracy and tends to underestimate execution time (MAPE and MPE at 1 GHz of 20% and 8.5%, respectively) while the Cortex-A15 model significantly overestimates execution time (MAPE and MPE at 1 GHz of 59% and -51%, respectively).

Hierarchical Cluster Analysis (HCA) applied to the measured HW PMC events was used to group workloads of similar behaviour together. When simultaneously comparing the MPE and the clusterings (Fig. 3), it is observed that: 1) the MPE varies significantly between different workloads; 2) workloads of the same cluster exhibit similar MPEs (e.g. cluster 4: +47%, cluster 8: -66%, cluster 10: -3%); 3) workloads with significantly large MPEs tend to be in a cluster of their own (as they exhibit specific and repeated micro-architectural behaviour). The workload *par-basicmath-rad2deg* (Cluster 16) has the highest MAPE of 285% at 600 MHz.

The workload errors have a similar pattern across all frequencies tested, and the MPE on both the Cortex-A7 and Cortex-A15 becomes gradually more positive with frequency.

A. Micro-benchmarks

Micro-benchmarks are often used to identify specific system metrics. The *lmbench* micro-benchmarking suite was used to measure the latency of accessing specific parts of the memory hierarchy on both the hardware platform and the model (Fig. 4) and found that the DRAM memory latency was too low in the model and that the Cortex-A7 L2 cache latency was too high, with the other measurements being very close between the gem5 model and HW platform. Memory latency, operation latency and memory bandwidth tests corroborate the tests conducted in [11]. They show several aspects of the model that can be improved (such as DRAM memory latency, also highlighted in [11]) but these results do not explain the significant negative execution time MPEs.

While micro-benchmarks are well suited for measuring specific micro-architectural metrics, they do not give an idea of where the large sources of error are for different workloads or which sources are most significant. The remainder of this section employs statistical methods to accomplish this, and focusses on the Cortex-A15 to demonstrate the approach.

B. Cluster and Correlation Analysis (HW PMC Events)

Clustering of the workloads demonstrated how the execution time MPE was closely related to the *type* of workload, as indicated by HW PMC events. In this section, HCA is used to identify clusters of PMC events that correlate with each other across the workloads. This enables groups of PMCs with similar behaviour to be identified and shows the relationships between the events. The correlation between each PMC event and the execution time MPE was calculated and then combined with the HCA to establish how the workload execution time is affected by the PMC event cluster rates (Fig. 5). A positive correlation means that the execution time of a workload with a high rate of the event in question tends to be underestimated.

The events with the largest positive correlation all appear in Cluster 1, which contains events related to memory barriers and exclusive instructions (0x6C, 0x6D, 0x7E, see Fig. 5), which occur frequently in concurrent applications, suggesting

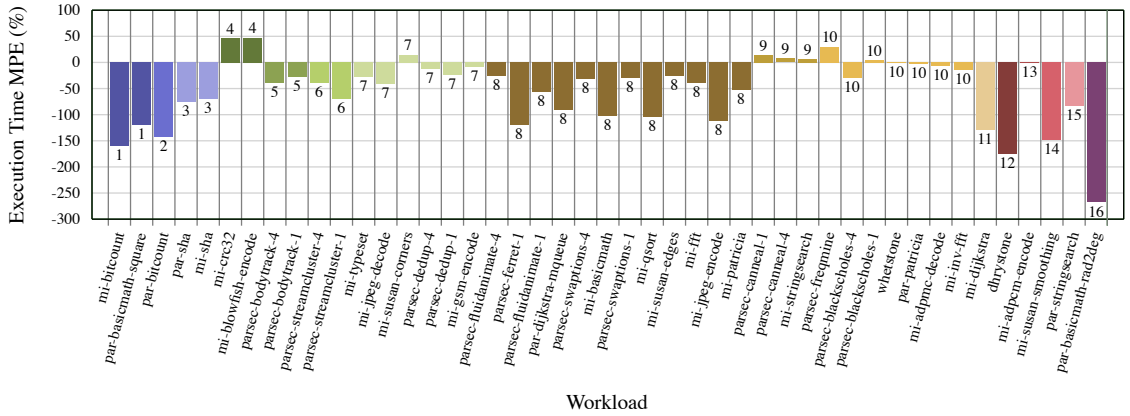


Fig. 3. Execution time Mean Percentage Error (MPE) for each workload running at 1 GHz on the Cortex-A15 cluster. Workloads are ordered, coloured and labelled (above bars) by cluster designation from Hierarchical Cluster Analysis (HCA) of the hardware PMC data. A positive MPE indicates an overestimation of the performance (underestimation of the execution time). MiBench prefix: *mi*, ParMiBench prefix: *par*, PARSEC prefix: *parsec*.

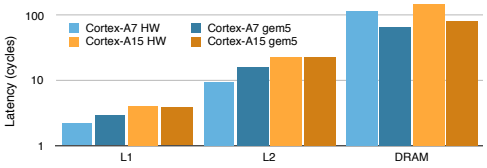


Fig. 4. Measured memory latency with a stride of 256

the cost of inter-process communication could be too low. Cluster 28, which contains events counting unaligned memory accesses, also has a large positive correlation.

As discovered earlier in this section, many of the larger errors are associated with workloads whose execution time is overestimated. Cluster 5 has the largest negative correlation and contains events related to the rate of branches and control flow operations (0x12, 0x76, 0x78). However, the rate of branch *mispredictions* (0x10) has a negative but notably smaller (in magnitude) correlation. Cluster 7 (instructions retired and instructions speculatively executed) and Cluster 8 (integer data processing speculatively executed) also have notable negative correlations, showing that CPU and integer intensive workloads tend to have large negative MPEs. This does not necessarily mean that the integer operation latencies are incorrect; workloads with many integer operations may also have many branch instructions which could be a source of error (only the correlation can be seen, not the causation).

C. Cluster and Correlation Analysis (gem5 Events)

The previous section analyses how the error is correlated with HW PMC events. Conducting the same analysis using the estimated gem5 events gives a different insight and is contrasted with the analysis of the previous section to identify differences between the modelled system and the hardware platform. The gem5 simulation outputs thousands of statistics compared to the tens of hardware PMC counters. The events with an absolute correlation of over 0.3 were extracted, resulting in a total number of 94 events. The largest cluster in these selected events was *Cluster A*, which was made up of 31 events and had the largest negative correlation with every event in the cluster having a correlation lower than -0.51. The vast majority of the events were related to the ITLB (Instruction Translate Lookaside Buffer). Most of the events concerned accesses to the *itb_walker_cache* specifically, which is designed to approximate the L2 ITLB component in the real hardware. Included in these were both hits and misses in the L2 ITLB. There were, however, also events related to the *itb* component (modelling the L1 ITLB) but these events were only related to misses, showing that that large negative execution time errors tend to occur when there are many L1 ITLB misses in gem5, and the L2 ITLB is accessed (resulting in a hit or a miss). This could suggest that the latency in the L2 ITLB is too high in the model, or that the source of error is highly correlated to this event. There are several events in *Cluster A* that are not directly related to the ITLB: *iew.exec_nop*, *fetch.TlbCycles*, *iew.predictedTakenIncorrect*, *fetch.PendingTrapStallCycles*, and *branchPred.RASInCorrect* (showing strong correlation between L2 ITLB accesses and branch mispredictions).

Fourteen events with large negative correlations (between -0.46 and -0.31) appear in *Cluster B*. Most of the events are related to predicted and mispredicted branches, e.g.: *commit.branchMispredicts*, *fetch.predictedBranches* and *branchPred.usedRAS* (Return Address Stack).

The next largest cluster is *Cluster C*, the events in which all have a smaller negative correlation of -0.35 or -0.36. All events in this cluster are related to L1I cache misses. Other gem5 events with negative correlation are related to both the L2 MSHR (Miss Status Holding Registers), uncacheable latency due to CPU data and the L2 overall miss rate.

Forty of the gem5 events have a positive correlation and the largest cluster has three gem5 events, which are related to the fetch rate and the number of instructions committed per cycle. Other gem5 events with a positive correlation relate to

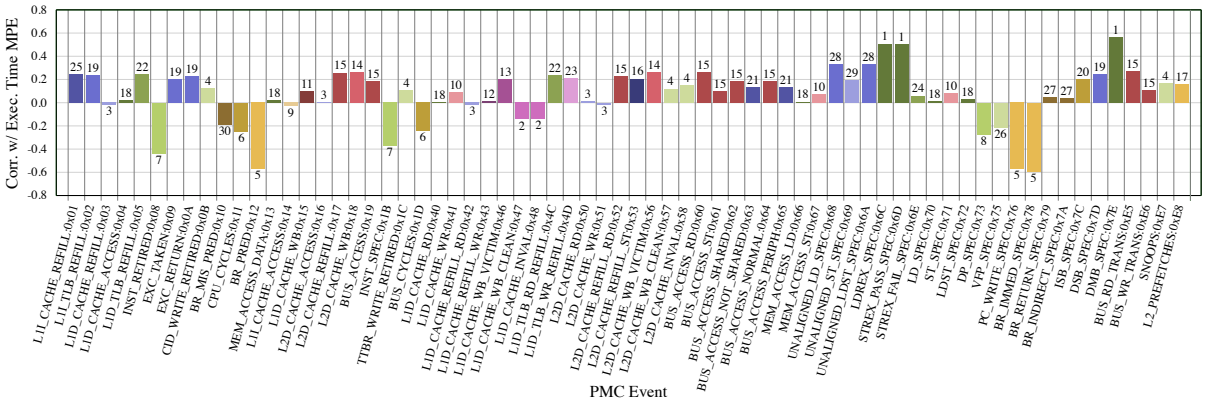


Fig. 5. Correlation of each HW PMC (rate) with the execution time MPE, labelled with clusters derived from HCA of the HW PMC data.

the L2 cache writebacks and the L2 cache miss latency, again suggesting the DRAM memory latency is too low.

This disparity between this analysis (analysing the modelled events with execution time error) and the analysis of profiled HW PMCs with execution time error (Section IV-B) identifies differences between HW and the model, and also allows an understanding of which events are the source, and which ones are simply correlated with the source. For example, the differences in branches and mispredicted branches between the two analyses suggest a significantly larger misprediction rate in the model; the BP is a potential source. While gem5 L2 ITLB access is highly correlated with error, it is also highly correlated with branch mispredictions. Furthermore, ITLB misses in HW events has a small positive correlation, showing a discrepancy between the number of ILTB misses in HW and gem5; the errors are not simply due to the ITLB miss penalty being too high, but another component causing a larger number of ITLB misses. Therefore, analysing the disparities between the HW event behaviour and gem5 event behaviour has shown that a large number of branch mispredictions are causing a large number of ITLB misses, significantly reducing the performance of the gem5 model.

D. Regression Analysis

Regression analysis was used to approximate the relationship between the hardware PMC events and the gem5 model error. The regression analysis is an important step in the methodology as events with a large correlation are not necessarily the most useful in identifying the sources of error. A forward-selection stepwise approach using the R^2 (a measure between 0 and 1 indicating *goodness-of-fit*) was used to identify which hardware PMC events to use as inputs to the model. Both the total event counts and the rates were made available as candidates to the selection process, which aimed to maximising the R^2 value. The dependent variable was set as the difference between the measured hardware execution time and the estimated gem5 execution time and a frequency of 1 GHz was considered in this analysis. The process adds events to the model until the *p-value* of any of the terms rises above 0.05 (a common rule of thumb is that terms with p-values above 0.05 are not statistically significant [24]).

The model selected seven events and achieved an R^2 and *Adjusted* R^2 (compensating for the number of predictors) of 0.97, showing that a model just using the hardware PMCs can accurately predict the gem5 model execution time error. The single best PMC event to predict the error was PC_WRITE_SPEC (total). The regression analysis finds SNOOPS and L1D_CACHE_REFILL_WR to be important in predicting error, despite not being found to be significant in the PMC correlation and cluster analysis. Other events in this selection (which include LDREX_SPEC and BR_RETURN_SPEC) corroborate the previous analysis.

The same analysis is conducted using the gem5 event statistics, and eight events were automatically selected, resulting in a model that achieved an R^2 and Adjusted R^2 of 0.99. The eight selected events included *commit.commitNonSpecStalls*, *branchPred.indirectMisses*, *dtb.prefetch faults* and *l2.ReadExReq hits (data)*.

The regression analysis shows how the gem5 error can be accurately predicted simply from the hardware PMC events. It selects (in order of importance) a handful of independent events related to different aspects of the system affecting the error, which can be cross-compared with the event clustering. In this example, it has largely reinforced the conclusions from the previous analysis and also identified some parts of the system to look at in more detail (e.g. snoops, DTLB prefetch faults, L1D cache writebacks, L2 data hits).

E. Event Comparison

The previous analysis has shown that the key sources are related to the Branch Predictor (BP) and the ITLB, specifically when accessing the L2 ITLB. Key gem5 events were matched and normalised to their HW PMC equivalents (Fig. 6). As well as comparing gem5 and PMC events for the mean of all of the workloads, the mean of selected clusters was also observed. As expected, there was a negligible difference in the total number of instructions committed (0x08) between HW and the gem5 model. Significantly fewer ($0.06\times$) ILTB misses (0x02) occurred in the model and they are very workload

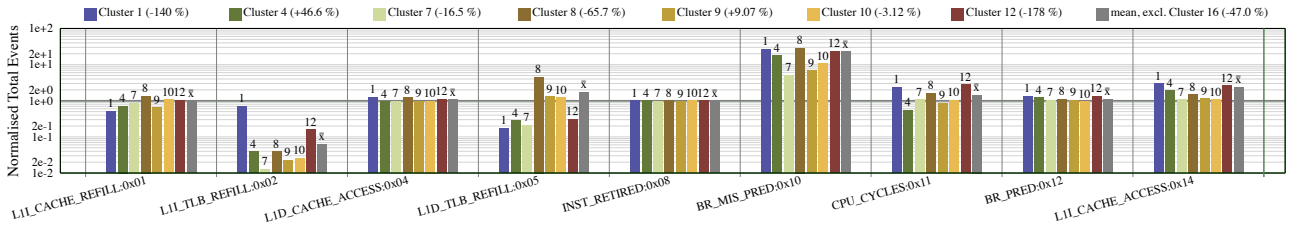


Fig. 6. Total gem5 events normalised with their HW PMC equivalent (bars over 1 indicate that gem5 overestimates the number of events). Results shown for selected clusters (cluster designations correspond to those in Fig. 3). The mean bars exclude Cluster 16.

dependent (*Cluster 1* has $0.7\times$, while *Cluster 7* has $0.01\times$, Fig. 6). However, the gem5 model predicts $1.7\times$ L1 DTLB misses. The model has $1.1\times$ predicted branches (0x12) and this is relatively consistent between clusters (Cluster 1 has $1.32\times$ while Cluster 10 has $0.93\times$). However, the gem5 model has a significantly higher number (mean: $21\times$) of branch mispredictions (0x10). In fact, for Cluster 16 (not shown in Fig. 6) the model has $1402\times$ branch mispredictions than HW. While the MPE is correlated highly with modelled L2 ITLB accesses, these occur less often in the model; branch mispredictions occur more often.

In HW, the BP has a mean prediction accuracy of 96% while in the gem5 model it is 65%. The lowest prediction accuracy in the gem5 model is 0.86%, which occurs for the same workload (*par-basicmath-rad2deg*, Cluster 16) that achieves the highest prediction accuracy in HW (99.9%). This workload has an execution time MPE of -268% (at 1 GHz).

The number of active CPU cycles (0x11, Fig. 6) closely follows the per-cluster errors (Fig. 3). On average, the gem5 model only executes $1.1\times$ more instructions speculatively than HW, meaning that the overestimated execution time is largely due to stalled cycles (also identified in Section IV-C). Despite this, there are over $2\times$ more L1I accesses in the model, assumed to be due to gem5 accessing it for every instruction, as opposed to decoding the entire cache line, but this is a topic for further investigation. Other events that diverge significantly are the L1D_REFILL_WR:0x43 ($9.9\times$) and L1D_WB:0x15 ($19\times$). The number of L2 prefetches are also significantly overestimated by the gem5 model.

This section has concurred with findings from the previous sections that did not rely on matched PMC and gem5 events, and quantified ITLB misses and BP performance metrics.

F. Summary

This section has presented several methods that together evaluate the gem5 model performance and identify sources of error without requiring detailed CPU information or direct equivalents between the HW and modelled events, and demonstrated them on the existing *ex5_big.py* model as an example.

While microbenchmarks found the modelled DRAM memory latency to be too low and discrepancies in the operation latencies (Section IV-A), the statistical techniques found that the most significant source of error was the branch predictor, which in turn caused a large number of L2 ITLB accesses (L1 ITLB misses). While the BP is the cause, the MPE could be exacerbated by large L2 ITLB access penalties. When comparing the HW PMC events directly to the gem5 events, there were a significantly lower number of ITLB misses in gem5 than in HW (while the DTLB accesses were similar, on average). This can be explained with the CPU's documentation [23], which shows that the TLB hierarchy in the hardware differs significantly from the one specified in the model. A 64-entry L1 ITLB is specified in gem5 when HW has a 32-entry one (also highlighted in [5]). However, changing this to the correct value results in a significantly larger MAPE, as expected, due to the BP errors present. This also suggests that the penalty of accessing the L2 ITLB is too high. The HW Cortex-A15 has a shared 512-entry 4-way set-associative L2 TLB whereas the model has two separate 1 KB 8-way set-associative caches simulating the L2 TLBs (one for instructions, one for data). The latency of these caches (4 cycles) appears high compared to the Cortex-A7 L2 TLBs (also 1 KB, latency of 2 cycles, albeit it is 4-way set associative) as well as the L1D cache (2 cycles, 32 KB, 4-way set-associative). Additionally, as they are not unified they will have a lower combined hit ratio than a single TLB of double the size.

There is interaction between the components of the model and changes to each part of the system have knock-on effects. It is therefore important to work on each component individually, and evaluate the full system after each change. It is also necessary to address the most significant sources of error first, otherwise changes to other parts of the system may not show a representative difference. It is therefore important to have a methodology to identify sources of error and a way of automating it (e.g. GemStone). While many of the graphs and stages of analysis generated by GemStone have been omitted for brevity, the key types of analysis and uses of them have been demonstrated. The key source of error was identified to be the BP and Section VII discusses improvements that have since been made to it.

V. Power Modelling

This section presents empirical power models designed to use the output statistics from gem5 to calculate the power consumption. The PMC-based model building methodology and corresponding *Powmon* software presented in [8] is used, which automates the two key stages of the methodology: PMC event selection and model formulation. The models are first developed and validated on HW using PMC events before

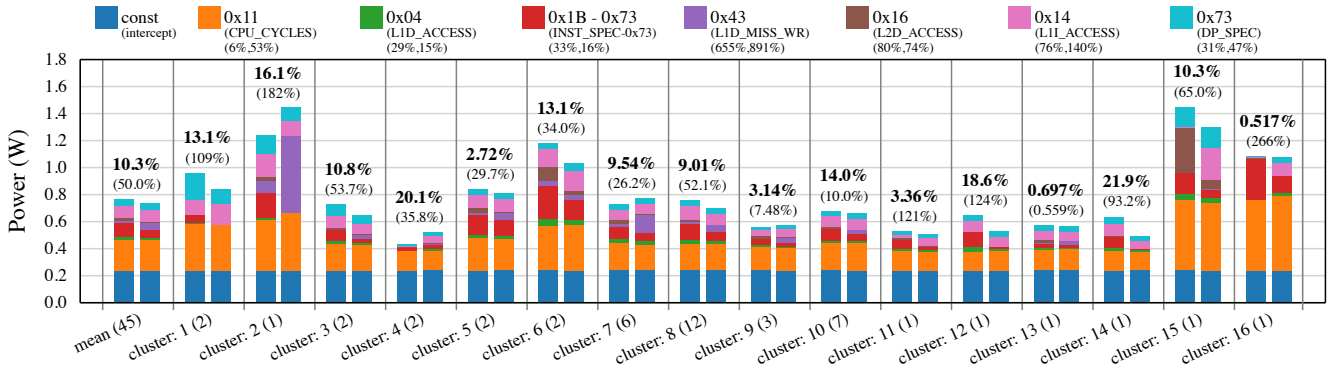


Fig. 7. Comparison of estimated power between using the HW PMC events (left bar) and the gem5 events (right bar) for each cluster (number of workloads in cluster in brackets in X-Axis labels) as per Fig 3 (Cortex-A15 CPU). The power MAPE is above each bar pair in bold, with the energy MAPE below it in brackets. The bars are colour-coded to show the power contribution from each model component.

being integrated into gem5 using modelled events. The models are therefore developed using PMC events found to have accurate and reliable gem5 equivalents.

The authors of [8] claim that the produced models are robust and maintain their accuracy even with unforeseen CPU workload patterns. The first step was to verify the effectiveness of the existing models by using the published model coefficients with the data collected from this experiment, which uses workloads that were not considered in [8] (PARSEC and ParMiBench). A MAPE of 5.6% was achieved, larger than the quoted 2.8%. However, there are several potential reasons for this: the board is not identical and components such as the SoC, power sensors and voltage regulators are subject to variation; the model can be affected by differences in storage media (the read write speeds of the SD card/eMMC); and the ambient temperature conditions have a large effect on power [25]. The coefficients were re-tuned using the same PMC selection from [8] and the data collected from the HW platform. A MAPE of 2.8% was achieved, corroborating the claim that the PMC selection is effective on workload sets that were not used in the selection process.

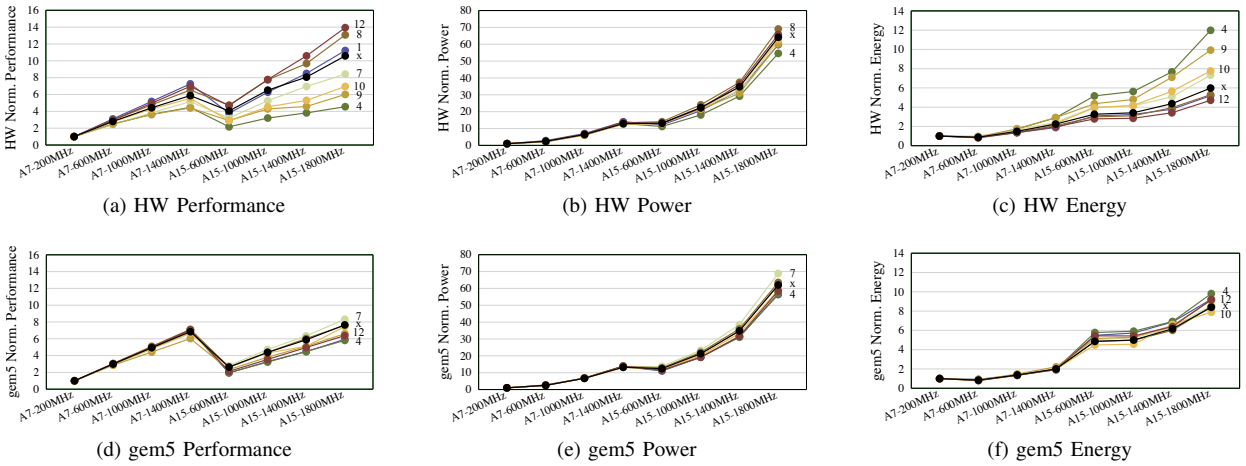
However, there were some PMCs in the original selection that were not readily available in the gem5 statistics (e.g. unaligned memory accesses) or that were found to be particularly inaccurate. The PMC selection experiment and algorithm (from [8]) was run with no restrictions on which PMC events could be chosen to obtain a baseline model. Because this experiment uses different workloads to [8] the PMC selection differed. The new model achieved a slightly lower MAPE of 4%, but the R^2 value, which measures the *goodness-of-fit* and is the metric that the model building process is optimising for, is improved, as expected. However, there were some issues with the chosen PMC events: 0x15 (L1 data cache writebacks) had an MPE of over 1000% for both the total and rate; 0x75, (floating-point speculatively executed) were misclassified in the gem5 model as SIMD floating point instructions. The approach used was to remove PMCs from the selection pool if it was not readily available in gem5 or if it had a significant error and there was an alternative event available. A trade-off had to be made as selecting only events that were modelled well in gem5 resulted in selecting events with a large inter-correlation (resulting in a poor model). The chosen events for the Cortex-A15 model are shown in the legend of Fig. 7. Event 0x1B has 0x73 subtracted from it to reduce multicollinearity. The model was built using all 65 workloads (Section III) and validated against the HW platform. It achieved a MAPE of 3.28%, standard error of regression (SER) of 0.049 W, and Adjusted R^2 of 0.996. The mean Variance Inflation Factor (VIF) across all model inputs was 6, indicating a low level of inter-correlation, as required. The *p-values* for all coefficients were lower than 0.0001 with the exception of two, which were lower than 0.02. Substituting PMC events with ones that were readily available and modelled with reasonable accuracy caused some degradation of the model but its accuracy and VIF is still within an acceptable level. The maximum MAPE out of all 621 observations was 14% (*parsec-canneal-4* at 1400 MHz).

The Cortex-A7 model achieves an adjusted R^2, MAPE and SER of 0.992, 6.64%, and 0.014 W, respectively.

This section has built power models for the Cortex-A7 and Cortex-A15 clusters using PMC events suitable for the gem5 models and validated them using power measurements from the hardware platform. While the model parameters are omitted from this paper for brevity, all the parameters, an extended set of model quality statistics as well as software implementing them are made available.

VI. Performance, Power and Energy Evaluation

This section combines the gem5 model analysed in Section IV with the hardware-validated power models presented in Section V and evaluates the effect of gem5 modelling errors on the resulting power and energy estimations. The power model application software tool (Section III, Fig. 2) applies the same power model to PMCs collected from HW and gem5 modelled event statistics. The resulting power and energy is then compared. The gem5 estimated power is not compared to the hardware measured power for two reasons: 1) the power sensors do not provide accurate power readings for short



(a) HW Performance (b) HW Power (c) HW Energy

(d) gem5 Performance (e) gem5 Power (f) gem5 Energy

Fig. 8. Performance, power and energy scaling normalised to 200 MHz on the Cortex-A7 CPU. Cluster numbers correspond to Fig. 3

workload durations and repeated workloads behave differently to single runs; 2) the measured power depends heavily on the temperature and voltage conditions which are not modelled in gem5. Therefore, a fair comparison is achieved by using the equivalent PMC events and modelled gem5 events with the same model, with the same voltage-frequency lookup values. The power model uses the event *rate* to estimate the power consumption, as opposed to the total number of events. The chosen events and their MAPEs are shown in the legend of Fig. 7 (first number is the rate MAPE, the second is the total MAPE).

The estimated power from the HW PMC events and the gem5 modelled events was compared for each cluster (Fig. 7). Despite large errors in the gem5 modelled events, the mean power MPE and MAPE for the 45 workloads are 3.3% and 10%, respectively. The reason for this is clear when observing each predicted power component; the largest components are the intercept (static and constant dynamic power) and 0x11 rate (which has a low MPE). Many of the other components cancel each other out. For example, in Cluster 13, 0x43 in the gem5 model is 9.7× larger than the measured HW equivalent. However, the error in this component is offset by 0x1B-0x73 and 0x16 being 2× and 6.7× larger, respectively, in the HW PMC model and a power MAPE of just 0.7% is calculated.

While the power error is low, the energy errors, which are dependent on the estimated execution time, are significantly larger. The energy MPE and MAPE are -43.6%, and 50.0%, respectively. The energy MAPE of each cluster varies significantly; from as low as 0.6% (Cluster 13) to as high as 266% (Cluster 16) (Fig. 7 [energy MAPE in brackets below the power MAPE]). Moreover, a cluster can have a very low power error but a very large energy error.

The Cortex-A7 model achieves a power MPE and MAPE of -5.48%, 7.97%, respectively, and an energy MPE and MAPE of 5.85%, and 14.6%, respectively. The Cortex-A7 model achieves lower power and energy errors due to the higher accuracy of the Cortex-A7 gem5 model (it is a simpler, in-order CPU).

The trade-offs between DVFS levels and different cores (e.g. in an Arm big.LITTLE [23] system) are important for many investigations. The performance, power and energy, normalised to the lowest frequency (200 MHz) of the Cortex-A7 cluster was calculated to see how the scaling of the gem5 model compared with HW (Fig. 8). Selected clusters from the HCA were also considered to see how different workload types scaled. A key observation is that the modelled Cortex-A15 performance is lower, with respect to the Cortex-A7, than measured from HW.

When considering only the Cortex-A15 scaling, the mean speedup running at 1800 MHz compared to 600 MHz is 2.7× and 2.9× for HW and the model, respectively, showing that the model accurately estimates the mean speedup. However, the model does not capture the workload diversity; the speedup range is 2.1× to 3.2× for HW and 2.8× to 3.0× for the model. The minimum speedup is Cluster 9 in both cases, but the maximum speedup is Cluster 2 for HW and Cluster 11 for the model. The energy increase estimated on HW has a range of 1.7× to 2.3× (mean: 1.8×), while the model estimates a range of 1.6× to 1.9× (mean: 1.7×). This would need to be considered for studies that consider the scaling of frequencies, or trading off between the 'little' and the 'big' CPU.

VII. Improvements to the gem5 Models

Its active development community means that gem5 is constantly being updated and improved. Since the analysis in Section IV identified significant errors in the BP, a change (bug fix) has been made to the BP used in the *ex_big* (Cortex-A15) CPU model. Running GemStone with the new model results in a significantly improved performance MAPE and MPE of 18% and +10%, respectively, meaning that new gem5 model underestimates execution time on average. The energy MAPE improved from 50% to 18%. As well as affirming the identified errors in the BP, this also underlines a key motivation for a tool such as GemStone, that automatically evaluates a gem5 model against a fixed HW platform. In this case, a researcher would see very different results for their study depending on when they downloaded gem5. The GemStone tool can be run after

a change has been made to the simulator to verify the model behaviour against the HW reference (i.e. ensuring no major bugs have been introduced). It can also be run by the user to ensure the model gives the required level of accuracy and is suitable for their use-case. Furthermore, a model correction can cause a larger MAPE due to other errors present (e.g. increasing the L1 ITLB size), necessitating a tool that analyses sources of error. Remaining sources of error can be reduced by iteratively making changes and analysing the result with GemStone.

VIII. Conclusion

This work has presented a systematic methodology for comparing CPU performance models to reference hardware platforms and identifying sources of error, allowing such models to be improved, extended to other CPUs, validated after changes, and applicability tested for specific use-cases. It employs statistical analysis and does not require detailed CPU specifications to be known. The GemStone open-source software tool has been presented, which automates the methodology with Arm platforms and gem5. Accurate energy analysis has been enabled in gem5 by developing and integrating empirical power models of an Arm Cortex-A7 and Arm Cortex-A15, achieving MAPEs of 6.6% and 3.3%, respectively. Furthermore, a tool that allows power and energy analysis to be retrospectively applied to gem5 simulations is presented. Additionally, the effect of errors in the gem5 models on the performance, power and energy has been analysed, including the scaling with DVFS levels and between core types. It was also shown how a low power MAPE can be achieved despite significant errors on certain model inputs (highlighting the importance of considering individual workloads and model components, as well as the average) but significant energy MAPEs can still occur. This work has also highlighted many aspects of the gem5 models to analyse in more detail, including the TLB hierarchy, classification of floating-point and SIMD operations, and how the L1I cache is accessed. The methodology identified significant errors in the existing Arm Cortex-A15 gem5 model and found the branch predictor to be the key source of error. A later version of gem5 included a fix for a branch predictor bug and the mean percentage error swung from -51% to $+10\%$, further motivating the use of the presented methodology to validate simulator changes.

Acknowledgement

This work was supported by Arm Ltd. and EPSRC Grant EP/K034448/1 (the PRiME Programme).

The authors thank the IRIDIS High Performance Computing Facility, and associated support services at the University of Southampton.

Experimental data: DOI 10.5258/SOTON/D0420

References

[1] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[2] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *42nd Annu. IEEE/ACM Int. Symp. Microarchitecture*, Dec 2009, pp. 469–480.

[3] S. L. Xi, H. Jacobson, P. Bose, G. Y. Wei, and D. Brooks, "Quantifying sources of error in McPAT and potential impacts on architectural studies," in *IEEE 21st Int. Symp. High Performance Computer Architecture (HPCA)*, Feb 2015, pp. 577–589.

[4] T. Nowatzki, J. Menon, C. han Ho, and K. Sankaralingam, "gem5, GPGPUSim, McPAT, GPUWattch, "your favorite simulator here" considered harmful," 2014.

[5] A. Gutierrez, J. Pusdesris, R. Dreslinski, T. Mudge, C. Sudanthi, C. Emmons, M. Hayenga, and N. Paver, "Sources of error in full-system simulation," pp. 13–22, 03 2014.

[6] W. Lee, Y. Kim, J. H. Ryoo, D. Sunwoo, A. Gerstlauer, and L. K. John, "PowerTrain: A learning-based calibration of McPAT power models," in *IEEE/ACM Int. Symp. Low Power Electronics and Design (ISLPED)*, July 2015, pp. 189–194.

[7] R. Bertran, M. Gonzalez, X. Martorell, N. Navarro, and E. Ayguade, "Decomposable and responsive power models for multicore processors using performance counters," in *24th ACM Int. Conf. Supercomputing*, ser. ICS '10. New York, NY, USA: ACM, 2010, pp. 147–158.

[8] M. J. Walker, S. Diestelhorst, A. Hansson, A. K. Das, S. Yang, B. M. Al-Hashimi, and G. V. Merrett, "Accurate and stable run-time power modeling for mobile and embedded CPUs," *IEEE TCAD*, vol. 36, no. 1, pp. 106–119, Jan 2017.

[9] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. 27th Int. Symp. Computer Architecture*, June 2000, pp. 83–94.

[10] B. Black and J. P. Shen, "Calibration of microprocessor performance models," *Computer*, vol. 31, no. 5, pp. 59–65, May 1998.

[11] A. Butko, F. Bruguier, A. Gamati, G. Sassatelli, D. Novo, L. Torres, and M. Robert, "Full-system simulation of big.LITTLE multicore architecture for performance and energy exploration," in *IEEE Int. Symp. Embedded Multicore/Many-core Systems-on-Chip*, Sept 2016, pp. 201–208.

[12] F. Bellosa, "The benefits of event-driven energy accounting in power-sensitive systems," in *Proc. 9th Workshop on ACM SIGOPS European Workshop: Beyond the PC: New Challenges for the Operating System*, ser. EW 9. New York, NY, USA: ACM, 2000, pp. 37–42.

[13] A. Butko, R. Garibotti, L. Ost, and G. Sassatelli, "Accuracy evaluation of gem5 simulator system," in *7th Int. Workshop Reconfigurable and Communication-Centric Systems-on-Chip*, July 2012, pp. 1–7.

[14] F. A. Endo, D. Courouss, and H. P. Charles, "Micro-architectural simulation of in-order and out-of-order arm microprocessors with gem5," in *Int. Conf. Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS XIV)*, July 2014, pp. 266–273.

[15] K. R. Basireddy, M. Walker, D. Balsamo, S. Diestelhorst, B. Al-Hashimi, and G. Merrett, "Empirical cpu power modelling and estimation in the gem5 simulator," in *27th Int. Symp. Power and Timing Modeling, Optimization and Simulation*, July 2017.

[16] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *4th Annu. IEEE Int. Workshop Workload Characterization*, Dec 2001, pp. 3–14.

[17] S. M. Z. Iqbal, Y. Liang, and H. Grahn, "ParMiBench - An open-source benchmark for embedded multiprocessor systems," *IEEE Comput. Archit. Lett.*, vol. 9, no. 2, pp. 45–48, Jul. 2010.

[18] L. McVoy and C. Staelin, "LMbench: Portable tools for performance analysis," in *Proc. Annu. Conf. USENIX Annu. Technical Conf.*, ser. ATEC '96, 1996, pp. 23–23.

[19] R. Longbottom, "Roy Longbottom's PC benchmark collection," http://www.roylongbottom.org.uk, Sept 2014, [Online; accessed 2-Jun-2015].

[20] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.

[21] Arm Ltd., "Dhrystone and MIPs performance of ARM processors," 2011.

[22] H. J. Curnow and B. A. Wichmann, "A synthetic benchmark," in *The Computer Journal*, vol. 19, no. 1, 1976, pp. 43–49.

[23] Arm Ltd., "Cortex-A15 MPCore, r3p3," 2013.

[24] R. Fisher, *Statistical Methods for Research Workers*, 1925.

[25] M. J. Walker, S. Diestelhorst, A. Hansson, D. Balsamo, G. V. Merrett, and B. M. Al-Hashimi, "Thermally-aware composite run-time CPU power models," July 2016.