# A Study of Hardware Performance Monitoring Counter Selection in Power Modeling of Computing Systems

Reza Zamani and Ahmad Afsahi
*Department of Electrical and Computer Engineering*
*Queen's University*
*Kingston, Ontario, Canada*
*Email: reza.zamani@queensu.ca, ahmad.afsahi@queensu.ca*

***Abstract*—Power management and energy savings in high-performance computing has become an increasingly important design constraint. The foundation of many power/energy saving methods is based on power consumption models, which commonly rely on hardware performance monitoring counters (PMCs). Various events are provided by processor manufacturers to be monitored using PMCs. PMC event selection has been mainly based on architectural intuitions. However, efficient use of PMCs requires a carefully selected set of events. Therefore, a comprehensive study of PMC events with regards to power modeling is needed to understand and enhance such power models.**

**In this paper, we study the relationship of PMC events with power consumption in the context of single-PMC and multi-PMC power models. Our OpenMP applications are from NAS Parallel Benchmark (BT, CG, LU, and SP) running on an AMD machine. We present the single-PMC selection results for each of our test applications, as well as a unified list for all four applications. Unlike other work that do not consider PMCs as each others' covariates, we present a method to select the most correlated set of PMC events for a given application. Our method finds the desired set of events with 6 times less number of executions compared to a principal component analysis (PCA) method. In addition, we have investigated variability of measurement for correlation coefficients. The 95% confidence interval of power-PMC and PMC-PMC correlation coefficients falls within 1.6% and 2.3% of their measured values, respectively. Furthermore, we study the power and PMC trends in the context of time-series and show that power estimates can be enhanced more than common regression methods. We show that the ARMAX model, a time-series candidate for real-time power estimation, can estimate system power consumption with a mean absolute error (total signal) of 0.1-0.5% in our applications.**

***Keywords*-performance monitoring counters; power modeling; energy saving;**

## I. Introduction

Power consumption and cooling issues are among the important design constraints of current high-performance computing (HPC) systems. High operational and maintenance costs of HPC systems, mainly due to their power consumption, make their power efficiency metrics resemble like their cost indices. Runtime power savings are mostly achieved by a power management (PM) module, which dynamically optimizes the parameters of the system (e.g., processor voltage/frequency) to meet its performance and power requirements. Having access to a real-time power measurement for an adaptive PM module is invaluable. Some of the recent high-end servers provide embedded power and thermal measurements, such as HP [1] and IBM PowerExecutive [2]. In most applications, variations in power consumption happen so quickly that an external power measurement reading cannot provide an in time measurement for the PM module. Applications suffering from power measurement delays, as well as systems without an embedded measurement system, can significanly enjoy an accurate power estimation model. An accurate power estimation model can perform as a control feedback loop for enhancing decision making of the PM module. A delayed real power measurement (e.g., external) can be used to update power estimation model in each step.

In most common processors, the number of PMC events is significantly large compared to the number of events that can be measured *simultaneously*. Previously, architectural intuitions have guided selection of PMCs for modeling workload/power consumption of a system [3]–[9]. However, it is unclear which PMC event "group" selection fits such power models the best when multiple PMCs can be utilized simultaneously in a model.

The authors in [3] have shown that power and performance metrics of a computing system can be estimated and predicted accurately, using a time-series model with PMCs and previous external power measurements as inputs. This work is different from [3] as it focuses on the search for the best PMC set; it also studies PMC selection and filter size in efficiency of the time-series model used in [3]. Bircher et al. [4] have studied single-PMC correlation of 23 PMC events with power consumption. Our work differs from them as we study multi-PMC selections and we consider the effect of PMC covariates in our selection. Lively et al. [10] have studied selection of multi-PMC events on 324 nodes for hybrid MPI/OpenMP programs. They investigate 40 PMCs using a performance-tuned supervised principal component analysis (PCA) [11] method. We use a different approach from [10] as we do not assume that different threads/processes of a parallel application exhibit similar

PMC event rate statistics. In addition, our experiments run multithreaded OpenMP [12] applications on a single node while studying a twice larger PMC event selection pool without requiring a PCA method.

The contributions of this paper includes providing a comprehensive statistical study of available PMC events, despite inherent measurement limitations, with regards to power modeling. A better understanding of this matter is necessary for researchers to enhance PMC-based power models. Furthermore, we propose a method to identify the most correlated set of PMCs for a given application. We also provide the best single-PMC and multi-PMC selection results for BT, CG, LU, and SP applications of NAS parallel benchmark (NPB) [13], and compare their power estimation efficiency using time-series approaches.

The rest of this paper is organized as follows. Section II provides some background information about PMCs and mathematical definitions used in this paper. In Section III, we describe our experimental framework. Section IV studies the measurement variability in PMC/power experiments. In Section V and Section VI, we explain our single-PMC and multi-PMC selection methods along with our results. In Section VII, we use the obtained sets of PMCs for power modeling in a time-series model. We discuss some of the related work to this paper in Section VIII. Section IX concludes the paper.

## II. Background

In this section, we provide some background information regarding PMC measurement along with some of the mathematical definitions and formulas used in this paper.

*1) PMC Modes:* Many of the available PMC drivers, such as *PerfCtr* [14] and *hwpmc* [15], virtualize the PMCs in the system and provide user level support for both system-wide PMC counting and process-private PMC counting. In the process-private mode, after PMCs are attached to a target process, they are counted (or sampled) only when their process is scheduled on a CPU. In the system-wide mode PMC measurement, PMCs are counted regardless of the running processes, and they capture the hardware events for the entire system for each processor (or core).

The process-private mode is more suitable for performance tuning of applications, as it focuses on selected processes and threads in the system. All the processes in the system contribute to power consumption. Therefore, a system-wide PMC measurement, in most cases, is more suitable for relating PMCs to system-level power consumption. In a multiprocessor system, programming PMCs with their hardware events and measuring them for each processor is performed independently from other processors in the system. In addition, the number of available PMC registers on each processor is usually much smaller than the number of available PMC events that can be monitored. For example, for each core of our AMD Opteron processor (refer tp Section III, more than 160 different PMC events are available to be monitored using the four available PMC registers.

System-wide PMC measurement can be performed symmetrically or asymmetrically, with respect to different processors/cores. A symmetric PMC measurement uses an identical set of PMC events on all processors. An asymmetric PMC measurement uses non-identical sets of PMC events on different processors, and therefore the one-to-one associations of a measured PMC on a processor with processes are lost, unless knowning the details of scheduler's decisions. Elaboration of this issue is out of scope of this paper. In short, a symmetric PMC measurement has the benefit of capturing relationship of PMC events with system-level power consumption without being impacted by scheduler, however, it suffers from the limitation on the number of PMCs that can be simultaneously measured.

*2) Vector Projection:* We review projection of a vector on a line in this part. Let $L$ be a line that is the span of a non-zero vector $\mathbf{v} \in \mathbb{R}^n$. Any vector $\mathbf{x} \in \mathbb{R}^n$, can be decomposed as a parallel component to the line $L$, $\mathbf{x}_L^{\parallel}$, and an orthogonal component to the line $L$, $\mathbf{x}_L^{\perp} = \mathbf{x} - \mathbf{x}_L^{\parallel}$. The parallel component, $\mathbf{x}_L^{\parallel}$, is equivalent to the projection of vector $\mathbf{x}$ on line $L$ and it is shown as $\mathrm{Proj}_L \mathbf{x} = \frac{\langle \mathbf{x}, \mathbf{v} \rangle}{\langle \mathbf{v}, \mathbf{v} \rangle} \mathbf{v}$. The inner product of vectors $\mathbf{x}$ and $\mathbf{v}$ is denoted as $\langle \mathbf{x}, \mathbf{v} \rangle$.

## III. Experimental Framework

All the experiments are conducted on a Dell PowerEdge R805 SMP server. The server has two quad-core 2.0 GHz AMD Opteron processors. The processors have 12 KB shared execution trace cache, and 16 KB L1 shared data cache on each core. The L2 cache available per core is 512 KB. Each processor chip also has a shared 2 MB L3 cache. Our system has 8 GB DDR-2 SDRAM (667 MHz) memory.

Our measurement infrastructure consists of a Keithley 2701/7710 digital multi-meter (DMM), a 10 $\Omega$ shunt resistor, and the node under measurement that performs the profiling task. We measure the power consumption of the node by measuring the voltage of the shunt resistor placed between the wall power outlet and the node. Knowing the value of the resistor, we first calculate the current and then the power and energy consumption of the node. We read 3 AC-voltage samples per second. In the DMM, the signal first goes through an internal analog RMS-converter, where 1000/60 DC samples are read out and averaged for each AC sample. The power measurements are validated with another industry-made power meter, Wattsup, and the measurement error is less than 1%.

The operating system is CentOS Linux, running kernel version 2.6.18 patched with the `perfctr` [14] library version 2.6.42 for PMC measurement purposes. We run our application programs on the node along with the PMC profiling code which is synchronized with the power measurement

Table I
MEAN AND STANDARD DEVIATION OF POWER-PMC AND PMC-PMC CORRELATION COEFFICIENTS (102 MEASUREMENTS)

| | *BT.C* | | *CG.C* | |
|---|---|---|---|---|
| *Correlation* | *Mean* | *S.D.* | *Mean* | *S.D.* |
| $(Power, E_1)$ | 0.490 | 0.016 | 0.974 | 0.013 |
| $(Power, E_2)$ | 0.454 | 0.018 | −0.967 | 0.014 |
| $(Power, E_3)$ | 0.410 | 0.017 | 0.953 | 0.019 |
| $(E_1, E_2)$ | 0.757 | 0.004 | −0.995 | 0.002 |
| $(E_1, E_3)$ | 0.778 | 0.003 | 0.956 | 0.003 |
| $(E_2, E_3)$ | 0.992 | 0.000 | −0.944 | 0.010 |
| | *LU.C* | | *SP.C* | |
| *Correlation* | *Mean* | *S.D.* | *Mean* | *S.D.* |
| $(Power, E_1)$ | 0.618 | 0.037 | 0.485 | 0.040 |
| $(Power, E_2)$ | 0.560 | 0.044 | −0.470 | 0.024 |
| $(Power, E_3)$ | 0.520 | 0.024 | −0.365 | 0.055 |
| $(E_1, E_2)$ | 0.759 | 0.006 | −0.764 | 0.007 |
| $(E_1, E_3)$ | 0.797 | 0.004 | −0.370 | 0.007 |
| $(E_2, E_3)$ | 0.580 | 0.010 | 0.169 | 0.020 |

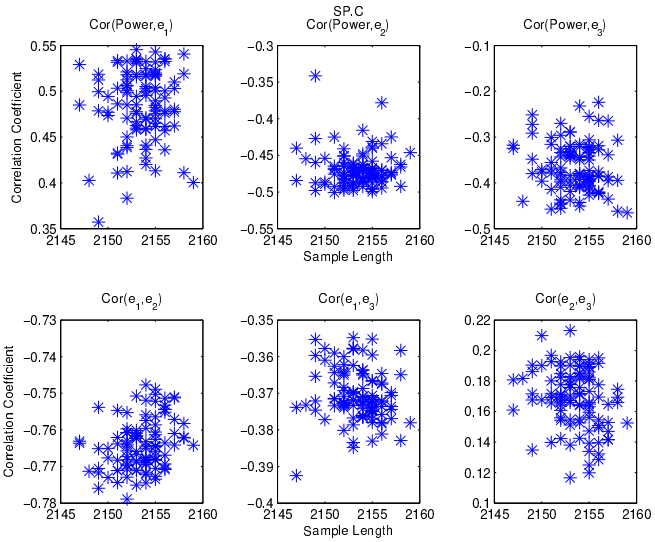


Figure 1. Correlation Measurement Variability for SP.C

software. The overhead of the PMC profiling code and the power measurement software is shown to be minimal. All the PMC events used in this paper are normalized with their number of cycles for each measurement (i.e., events/cycle).

We use four multithreaded OpenMP [12] applications from the NAS parallel benchmarks (NPB) [13] suite in our study. These NPB-3.3-OMP applications consists of *BT.C*, *CG.C*, *LU.C*, and *SP.C* running with eight threads. We have chosen those applications in class C of NPB-3.3 that run for longer than 300 seconds on our system, in order to have sufficient samples to be able to calculate an accurate correlation between our signals. The power estimations in this paper are performed offline in MATLAB.

## IV. MEASUREMENT VARIABILITY

In a real system, there is always a variability between multiple executions of a given application. This variability can be seen in different PMC metric measurements, as well as power consumption. Measurement variability can happen for many reasons, such as time variability, operating system interrupts, processor temperature change, etc. For example, variations in a processor temperature can change its leakage power and therefore change its power curve. Measurement variability has been studied before in other fields, such as phase detection [16]. In this section, we study the variability in measuring power-PMC and PMC-PMC correlations. To ensure model accuracy, it is critical that variability of the measured correlations over repeated tests is not significantly large. Evaluation of correlation variance becomes more important for models that do not use an adaptive approach. One of the noticeable variations in our repeated tests is variation in execution time of applications (sample length).

We have studied the power-PMC and PMC-PMC correlations of three PMC events for BT.C, CG.C, LU.C, and SP.C applications. These are the PMCs that individually have the highest correlation with power consumption for each application. We have measured the variance of correlation calculations over 102 runs for each application. The average and standard deviation of their correlation coefficients are shown in Table I. The selected three PMCs for each application are the first three PMCs in Table II, Table III, Table IV, and Table V. Due to limited space, we only show variability of correlation coefficients for SP.C in Figure 1.

Assuming measurements of correlation between our metrics are normally distributed, we can calculate the confidence interval for the mean of these measurements. The correlation coefficients measured between power consumption and the top three single PMCs for BT.C, CG.C, LU.C, and SP.C applications have a 95% confidence interval that spreads around their measured mean value up to 0.8%, 0.3%, 1.5%, and 1.6%, respectively. The 95% confidence interval for the correlation coefficients between the top three PMC metrics in each application spreads around their measured mean value up to 0.1%, 0.1%, 0.3%, and 2.3% for BT.C, CG.C, LU.C, and SP.C, respectively.

If a given application shows significantly different statistics during each runtime, a model cannot be based on an arbitrary execution and its statistics. Thus, it is crucial to verify the variability of statitics used in our method, such as power-PMC and PMC-PMC correlation coefficients. Knowing with 95% confidence that our measurements have an error of less than 1.6% for power-PMC correlation coefficients, and an error of less than 2.3% for PMC-PMC correlation coefficient (the value of the application/metrics

with the widest confidence interval) allows us to calculate correlation coefficient metrics based on measurements of one execution, as in the rest of this paper.

## V. Single PMC Selection

In this section, we compare different PMC events with respect to their correlation with power consumption for a given application. We provide the list of highest correlated PMCs for each application. However, the top correlated PMC events for each application is different from other applications. Therefore, in order to compare the overall correlation coefficient of different PMC events among our applications, we use a rank median approach: we rank the PMC events based on their correlation with power for each application and the overall rank of each PMC event is considered as the median of its ranks for different applications. Then, we provide a unified list of PMC events, ordered based on their overall rank (lower ranks refer to higher correlation coefficients).

### A. Correlation in Event Space

There are many PMC events available for a given modern processor, however, not all of them have a significant correlation with power consumption. After extensive experimentation and selection, 86 events have been chosen in this work that are relevant to power consumption modeling (power-PMC correlation coefficient of larger than 0.10). Let $n_e$ be the total number of available power-relevant PMC events ($n_e = 86$). Let $n_r$ be the number of available PMC registers. For the AMD Opteron processors used in this work $n_r = 4$. Measurements of the $i$-th PMC event, $e_i$, where $i \in \{1, \cdots, n_e\}$, with $k$ samples in time are denoted as vector $\mathbf{E_i} \in \mathbb{R}^k$. Their corresponding power measurements are represented by $\mathbf{P} \in \mathbb{R}^k$.

In this paper, sample correlation coefficient between measurement vectors $\mathbf{X}$ and $\mathbf{Y}$ is denoted as $r_{\mathbf{X},\mathbf{Y}}$. We find the power-PMC correlation coefficients for all the available power-relevant PMC events, $r_{\mathbf{E_i},\mathbf{P}}$, $i \in \{1, \cdots, n_e\}$. Having only $n_r$ PMC registers available for this measurements, this search takes $n_e/n_r$ runs for each application benchmark (in this work, $n_e/n_r$ = 86/4, thus 22 runs are required). This comparison does not consider PMC-PMC correlations (we investigate this part in Section VI). The correlation coefficient of each event with power consumption is measured for BT, CG, LU, and SP applications and the top correlated events for each application are reported in Table II, Table III, Table IV, and Table V, respectively.

Table II
BT.C - Top 25 Correlated Events

| Event Name | $r$ |
|---|---|
| L2 Fill/Writeback | 0.487 |
| Retired SSE Operations | 0.453 |
| Retired Instructions | 0.429 |
| Retired uops | 0.427 |
| Retired Move Ops | 0.427 |
| Retired x87 Floating Point Operations | 0.407 |
| Dispatched FPU Operations | 0.406 |
| Microarchitectural Early Cancel of an Access | 0.396 |
| Canceled Store to Load Forward Operations | 0.394 |
| Retired MMX/FP Instructions | 0.384 |
| Instruction Cache Fetches | 0.383 |
| Data Cache Accesses | 0.377 |
| L1 DTLB Hit | 0.371 |
| L1 DTLB and L2 DTLB Miss | -0.363 |
| Retired Mispredicted Branch Instructions | 0.340 |
| Retired Taken Branch Instructions Mispredicted | 0.334 |
| Cycles with FPU operation $\geq$ 1 in FPU | -0.331 |
| Instruction Fetch Stall | -0.329 |
| Dispatch Stalls | -0.326 |
| Dispatch Stall for Reservation Station Full | -0.317 |
| Return Stack Hits | 0.311 |
| Memory Controller Bypass Counter Saturation | -0.311 |
| Data Cache Refills from L2 | 0.301 |
| Retired Near Returns | 0.298 |
| Decoder Empty | -0.297 |

Table III
CG.C - Top 25 Correlated Events

| Event Name | $r$ |
|---|---|
| CPU Clocks not Halted | 0.975 |
| Memory Requests by Type | -0.972 |
| L1 DTLB and L2 DTLB Miss | 0.962 |
| Retired uops | 0.946 |
| L3 Cache Misses | 0.935 |
| Read Request to L3 Cache | 0.933 |
| Dispatch Stall Waiting for All Quiet | -0.932 |
| HyperTransport Link 1 Transmit Bandwidth | 0.932 |
| DRAM Accesses (hit, miss, conflict) | 0.932 |
| HyperTransport Link 2 Transmit Bandwidth | 0.931 |
| Data Cache Refills from the Northbridge | 0.931 |
| MFENCE Instructions Retired | 0.930 |
| Data Cache Lines Evicted | 0.930 |
| HyperTransport Link 0 Transmit Bandwidth | 0.929 |
| Data Cache Refills from L2 | 0.927 |
| Memory Controller DRAM Command Slots Missed | 0.927 |
| CPU to DRAM Requests to Target Node | 0.927 |
| L1 DTLB Miss and L2 DTLB Hit | 0.927 |
| Retired MMX/FP Instructions | 0.925 |
| Data Cache Misses | 0.925 |
| Memory Controller Turnarounds | 0.921 |
| Retired Branch Instructions | 0.921 |
| Microarchitectural Late Cancel of an Access | 0.920 |
| IO to DRAM Requests to Target Node | 0.919 |
| local CPU requests to local/remote Memory/IO | 0.918 |

### B. Rank in Application Space

In this section, our objective is to find a set of events that each provides a good correlation with power for most applications. We use a rank median approach to identify these PMCs for our applications. First, each PMC event is ranked for each application based on its correlation coefficient with power consumption. Stronger correlation coefficients are represented by smaller ranks. The overall rank of each event is calculated as its rank median among our applications. The overall ranks for the top 24 PMCs for our applications are provided in Table VI.

Table IV
LU.C - TOP 25 CORRELATED EVENTS

| Event Name | $r$ |
|---|---|
| Data Cache Lines Evicted | 0.604 |
| L2 Fill/Writeback | 0.592 |
| Retired Move Ops | 0.507 |
| Locked Operations (cycles spent) | -0.468 |
| Dispatch Stall for Serialization | 0.450 |
| Pipeline Restart Due to Instruction Stream Probe | -0.444 |
| Requests to L2 Cache | 0.424 |
| Memory Controller Bypass Counter Saturation | 0.409 |
| Canceled Store to Load Forward Operations | 0.400 |
| Data Cache Misses | -0.382 |
| Retired MMX/FP Instructions | 0.380 |
| Dispatch Stall for Reservation Station Full | 0.375 |
| Dispatched FPU Operations | 0.349 |
| LFENCE Instructions Retired | 0.329 |
| Microarchitectural Early Cancel of an Access | -0.324 |
| Data Cache Accesses | 0.320 |
| Retired SSE Operations | 0.299 |
| L2 Cache Misses | 0.299 |
| L1 DTLB Hit | 0.298 |
| Microarchitectural Late Cancel of an Access | 0.297 |
| Probe Responses | 0.282 |
| L3 Cache Misses | 0.282 |
| HyperTransport Link 2 Transmit Bandwidth | 0.264 |
| Octwords Written to System | 0.263 |
| HyperTransport Link 1 Transmit Bandwidth | 0.260 |

Table V
SP.C - TOP 25 CORRELATED EVENTS

| Event Name | $r$ |
|---|---|
| Data Cache Misses | 0.508 |
| Memory Controller Bypass Counter Saturation | -0.473 |
| Decoder Empty | -0.470 |
| Dispatch Stall for Reorder Buffer Full | 0.427 |
| Retired Taken Branch Instructions | -0.405 |
| MFENCE Instructions Retired | 0.396 |
| Memory Controller Requests | -0.388 |
| Retired Branch Instructions | -0.374 |
| L1 ITLB Miss, L2 ITLB Miss | -0.343 |
| Microarchitectural Early Cancel of an Access | 0.279 |
| CPU Clocks not Halted | 0.275 |
| Memory Requests by Type | -0.274 |
| Dispatch Stall Waiting for All Quiet | -0.270 |
| Data Cache Lines Evicted | 0.257 |
| CPU to DRAM Requests to Target Node | -0.255 |
| Dispatch Stall for FPU Full | -0.253 |
| Retired x87 Floating Point Operations | 0.248 |
| L2 Cache Misses | 0.242 |
| Ineffective Software Prefetch | -0.240 |
| Upstream Requests | -0.239 |
| Data Cache Refills from L2 | 0.237 |
| DRAM Accesses (hit, miss, conflict) | -0.231 |
| L3 Cache Misses | -0.219 |
| Cache Block Commands | -0.214 |
| Requests to L2 Cache | 0.212 |

The best PMC for power modeling depends on the application. In this section, we investigated the relationship of each PMC with power consumption of our applications. We found that events such as *microarchitectural early cancel of an access* and *data cache lines evicted* are overall among the top events for applications used in this paper, representing a better choice than commonly used intuition based events, such as *retired uops*. Many PMCs show a significant covariance with each other. An example of this is shown in Table I where $E1$, $E2$, and $E3$ show an absolute correlation coefficient of 0.16-0.99 with each other.

It is essential to consider PMC covariance when selecting more than one PMC for a power-PMC model. In the next section, we turn our attention to multi-PMC selection and take into account the PMC covariance.

Table VI
TOP 24 UNIFIED EVENTS (RANK MEDIAN)

| Event Name | Rank |
|---|---|
| Microarchitectural Early Cancel of an Access | 12.5 |
| Data Cache Lines Evicted | 13.5 |
| L2 Fill/Writeback | 14.0 |
| Data Cache Misses | 15.0 |
| Retired MMX/FP Instructions | 15.0 |
| Memory Controller Bypass Counter Saturation | 15.0 |
| Retired uops | 20.0 |
| Dispatched FPU Operations | 21.5 |
| Data Cache Refills from L2 | 22.0 |
| L3 Cache Misses | 22.5 |
| Retired SSE Operations | 23.0 |
| Canceled Store to Load Forward Operations | 24.0 |
| Dispatch Stall for Reservation Station Full | 25.0 |
| Retired x87 Floating Point Operations | 25.0 |
| Requests to L2 Cache | 26.0 |
| Memory Requests by Type | 26.5 |
| CPU Clocks not Halted | 26.5 |
| L2 Cache Misses | 26.5 |
| Octwords Written to System | 28.0 |
| Retired Move Ops | 29.0 |
| Data Cache Accesses | 30.0 |
| Dispatch Stall Waiting for All Quiet | 30.0 |
| DRAM Accesses (hit, miss, conflict) | 30.0 |
| Cache Block Commands | 30.0 |

## VI. MULTIPLE PMC SELECTION

In this section, we take into account the correlation of PMCs with each other in order to find the best set of PMCs for simultaneous measurement in system power modeling. If it was possible to measure all the available PMC events simultaneously (e.g., 86 PMC events), finding the best set of PMC events would have been straightforward using principal components analysis. The most important PMC events for an application would have been the PMC events that have the largest component in the most significant eigenvector (i.e., the eigenvector that is associated with the largest eigenvalue). However, due to the limited number of PMC registers available on each processor core (e.g., four registers), while using a symmetric PMC measurement method for data collection, it is not feasible to measure many PMC-power trends simultaneously. Furthermore, variations among the collected data of repeated experiments for a given application do not allow us to aggregate them and to use them in their time-sample domain at the same time.

Instead of using signals in time from different executions of a test, one could have measured the cross correlation of all the possible pairs of events and power to obtain their covariance matrix. It is possible to perform PCA calculations directly from a covariance matrix. The challenge in this approach is that a large number of cross correlation measurements is required to be done. For example, in our study that uses 86 PMC events, the number of pairs of PMCs that their covariance has to be measured is $\binom{86}{2} = 3655$ (not considering power measurements here). Each execution of an application can measure 4 PMCs and therefore it can capture up to $\binom{4}{2} = 6$ of the required 3655 pairs. The total number of executions for a given application is much larger than $3655/6$ (our estimate is between 1162 and 1247 experiments, which are the limits for 85 and 88 events using a recursive function $f(n) = f(n-3) + n - 3$, $f(4) = 1$), due to unavoidable repeated pairs. This large number of tests discourages us in using a PCA approach.

Here, we propose a sub-space projection method that searches for the best combination of PMC events without using a PCA method. In a PCA method, the first eigenvector provides the significance of contribution of all the PMC events. However, even if the order of significance of all of PMC events were available we cannot measure more than four PMCs at a time. We use this fact as a leverage to reduce the calculations needed to find the top four PMCs. Our proposed sub-space projection method uses more than 6 times less number of executions than a PCA method to find the top four PMCs (176 runs for sub-space projection method, in contrast to more than 1162 runs for PCA). In the following, we explain our proposed method and present the results.

### A. Sub-Space Projection Method

In this section, we are searching for the most correlated set of PMCs with power consumption, denoted as $S$. Our method requires $n_r$ stages and each stage has four steps. In this section, $k$ represents the stage number ($1 \leq k \leq n_r$). The set of all PMC events is denoted as $T$, $|T| = n_e$. Let $S_k$ represent the set of PMCs found at the end of stage $k$ and $R_k$ be the set of the remaining PMC events for search at the end of stage $k$. $R_k$ is equivalent to the set difference of $T$ and $S_k$, $R_k = T \backslash S_k$. We perform the following four steps for each stage $k$.

*1) Simultaneous Measurement:* In this step, at stage $k$, we measure each member of $R_{k-1}$ simultaneously with all of the members of $S_{k-1}$. The members of $S_{k-1}$ will occupy $k-1$ of the available $n_r$ PMC registers. Therefore, $n_r - k + 1$ registers are available for assigning to the members of $R_{k-1}$ ($|R_{k-1}| = n_e - k + 1$). The number of times required to run each application to finish this step at stage $k$ is $\lceil (n_e - k + 1)/(n_r - k + 1) \rceil$. For $n_e = 86$, $n_r = 4$, the number of runs for stage 1 to 4 is 22, 29, 42, and 83 (total 176 runs), respectively. For $k = 1$, $S_0 = \emptyset$ and $R_0 = T$.

*2) Signal Decorrelation:* The second step is to decorrelate the signals of all of the PMC events in our remaining search pool (members of $R_{k-1}$) and their corresponding power measurements against the signals of the previously selected PMC events that are simultaneously measured with them (all members of $S_{k-1}$). The results of this step are the residual PMC and power signals. This step is skipped during the first stage.

For example, at stage $k = 4$, any of the remaining PMC events in the search pool, such as $x \in R_3$, will have a PMC measurement signal of $\mathbf{X}$ and a power measurement of $\mathbf{P}$. Let $S_3 = \{a, b, c\}$. The PMC signals of $a$, $b$, and $c$ events that are simultaneously measured with event $x$ are shown as $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$. The residual signal of PMC event $x$ when decorrelated against PMC signals of events $a$, $b$, and $c$, denoted as $\mathbf{X}^{\perp}_{\mathbf{ABC}}$, is calculated as follows:

$$\mathbf{X}^{\perp}_{\mathbf{A}} = \mathbf{X} - \mathrm{Proj}_{\mathbf{A}}\mathbf{X} \tag{1}$$

$$\mathbf{X}^{\perp}_{\mathbf{AB}} = \mathbf{X}^{\perp}_{\mathbf{A}} - \mathrm{Proj}_{\mathbf{B}}\mathbf{X}^{\perp}_{\mathbf{A}} \tag{2}$$

$$\mathbf{X}^{\perp}_{\mathbf{ABC}} = \mathbf{X}^{\perp}_{\mathbf{AB}} - \mathrm{Proj}_{\mathbf{C}}\mathbf{X}^{\perp}_{\mathbf{AB}} \tag{3}$$

The residual signal of power measurement associated with the measurement of PMC event $x$ after being decorrelated against $a$, $b$, and $c$, denoted as $\mathbf{P}^{\perp}_{\mathbf{ABC}}$ is calculated similarly to $\mathbf{X}^{\perp}_{\mathbf{ABC}}$.

*3) Residual Correlation:* In the third step, correlation coefficients between the PMC measurement residuals and power measurement residual for every $x \in R_{k-1}$ are calculated. For example, for $k = 4$ (similar to Section VI-A2), we calculate the correlation coefficient between $\mathbf{X}^{\perp}_{\mathbf{ABC}}$ and $\mathbf{P}^{\perp}_{\mathbf{ABC}}$, denoted as $r_{\mathbf{X},\mathbf{P}}$, for every $x \in R_3$.

*4) PMC Event Selection:* The fourth step is to find the PMC event that has the largest absolute residual correlation coefficient with power. For example, at stage $k$, we are looking for event $y \in R_{k-1}$ with decorellated PMC and power measurements of $\mathbf{Y}$ and $\mathbf{P_y}$ such that for every other event $x \in R_{k-1}$ with decorellated PMC and power measurements of $\mathbf{X}$ and $\mathbf{P_x}$: $|r_{\mathbf{Y},\mathbf{P_y}}| \geq |r_{\mathbf{X},\mathbf{Px}}|$.

After finding the PMC event $y$ with the strongest correlation coefficient with power it is moved from the search pool $R_{k-1}$ to the pool of selected PMCs and we go to the next stage: $S_k = S_{k-1} \cup \{y\}$, $R_k = R_{k-1} \backslash \{y\}$, and $k = k + 1$.

### B. Results

We apply the proposed sub-space projection method to our applications and we find the most significant set of PMC events related to power consumption. These events for BT.C (in order of significance and the most significant event first) are *L2 fill/writeback*, *dispatch stall for FPU full*, *retired move ops*, and *MFENCE instructions retired*. For CG.C these are *CPU clocks not halted*, *LS buffer 2 full*, *L1 ITLB miss, L2 ITLB hit*, and *LFENCE instructions retired*. Similarly for LU.C the events are *data cache lines evicted*, *dispatch stall for reorder buffer full memory controller DRAM command slots missed*, and *instruction cache fetches*. For SP.C

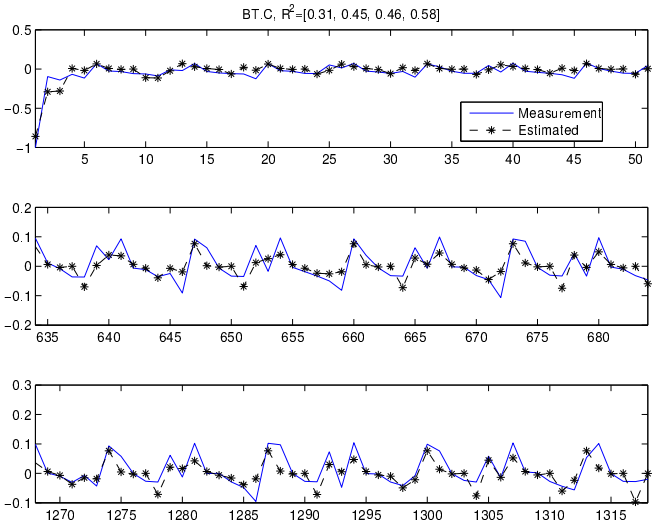


Figure 2. Zero-mean-norm. power variations captured by PMCs (BT.C)

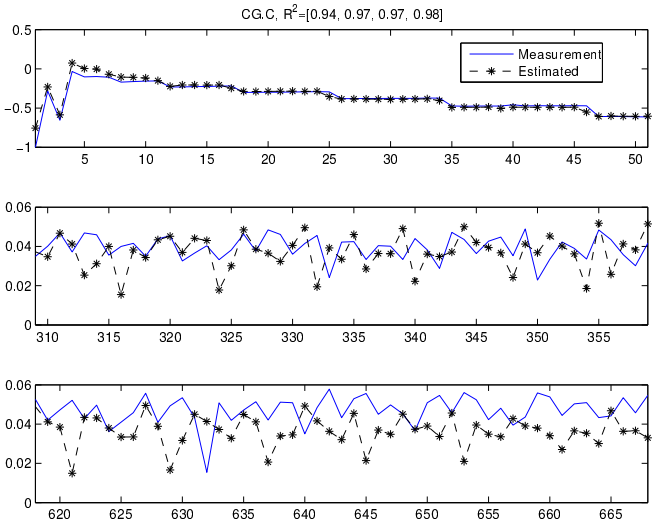


Figure 3. Zero-mean-norm. power variations captured by PMCs (CG.C)

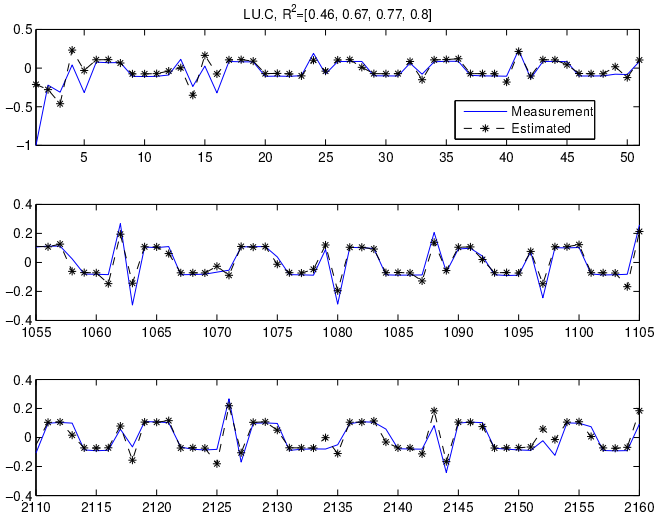


Figure 4. Zero-mean-norm. power variations captured by PMCs (LU.C)

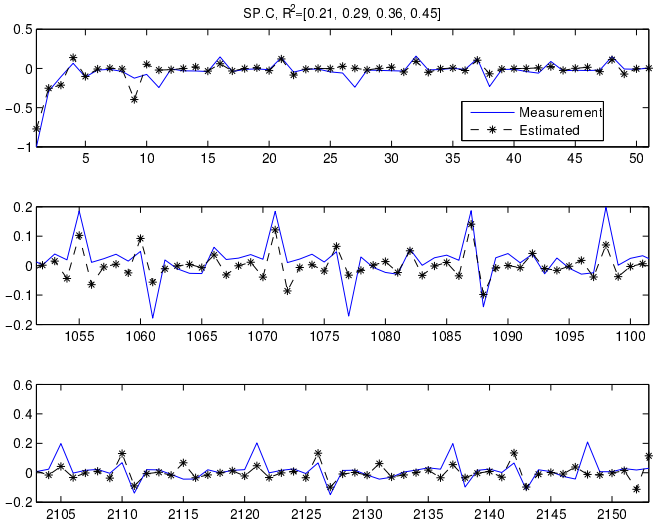


Figure 5. Zero-mean-norm. power variations captured by PMCs (SP.C)

the events are *data cache misses*, *microarchitectural early cancel of an access*, *L3 fills caused by L2 evictions*, and *memory controller bypass counter saturation*.

After finding the set of four PMCs for each application, denoted as $\{a, b, c, d\}$, we now present the projection of power consumption on these PMC events, $\mathbf{P}^{\parallel}_{\mathbf{ABCD}} = \mathbf{P} - \mathbf{P}^{\perp}_{\mathbf{ABCD}}$, in Figure 2, Figure 3, Figure 4, and Figure 5, respectively for BT, CG, LU, and SP applications. In these figures, the first, middle, and last 50 samples of the execution time and their estimated value (both normalized) only based on projection on PMC signals are provided for each application. The y-axis is the power consumption normalized to absolute value of 1 and mean of zero. The goodness of fit, $R^2$, for each of the projected signals is given on top of each figure. The $R^2$ is presented after adding projection on each significant PMC (in significance order). From left to right the presented $R^2$ value is associated with $\mathbf{P}^{\parallel}_{\mathbf{A}}$, $\mathbf{P}^{\parallel}_{\mathbf{AB}}$, $\mathbf{P}^{\parallel}_{\mathbf{ABC}}$, and $\mathbf{P}^{\parallel}_{\mathbf{ABCD}}$, respectively. The power projection on all PMC events, $\mathbf{P}^{\parallel}_{\mathbf{ABCD}}$, can be calculated as:

$$\mathrm{Proj}_{\mathbf{A}}\mathbf{P} + \mathrm{Proj}_{\mathbf{B}}\mathbf{P}^{\perp}_{\mathbf{A}} + \mathrm{Proj}_{\mathbf{C}}\mathbf{P}^{\perp}_{\mathbf{AB}} + \mathrm{Proj}_{\mathbf{D}}\mathbf{P}^{\perp}_{\mathbf{ABC}}$$

The best $R^2$ achieved for BT, CG, LU, and SP applications with their most significant PMC sets are 0.58, 0.98, 0.80, and 0.45, respectively. Using only the most significant PMC for power projection (i.e., $\mathbf{P}^{\parallel}_{\mathbf{A}}$) achieves $R^2$ of 0.31, 0.94, 0.46, and 0.21, similarly.

We extend our investigation to temporal analysis of $R^2$ in our applications. We present the $R^2$ of $\mathbf{P}^{\parallel}_{\mathbf{ABCD}}$ for each 64-sample segment of the execution of our applications in

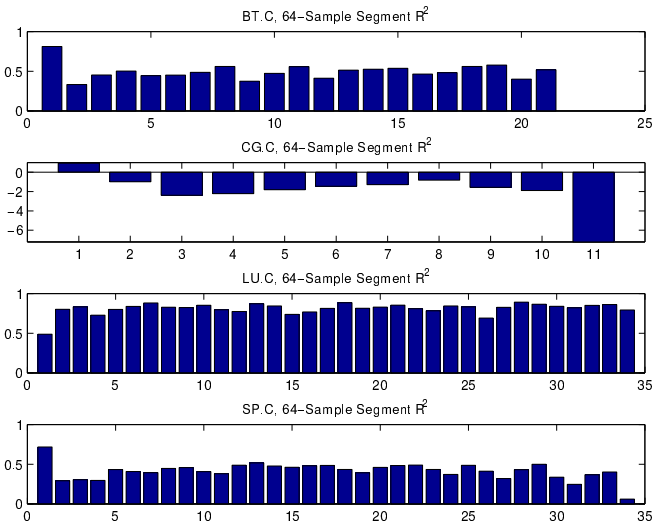


Figure 6. $R^2$ for 64-sample segments

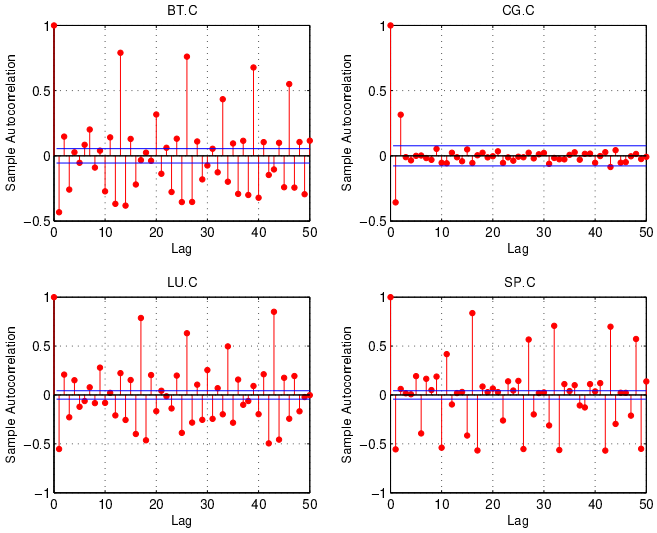


Figure 7. Sample autocorrelation function of differenced power

Figure 6. One can notice a significant difference in goodness of fit in the first segment of execution time (warm up phase). Except for LU, other applications show a much better $R^2$ in their first execution segment. It should be noted that our applications are launched on an idle system and the range of change in power consumption from idle to busy is much larger than between other phases of an application, thus gaining a better $R^2$. We believe that a minimal power consumption variation for CG ($\approx 4\%$ variation, see Figure 3) is the reason behind exhibiting a negative $R^2$ (except the first segment). A negative $R^2$ shows that the mean of the signal as an estimator performs better than the used method,

In this section we presented the best combination of PMCs and the portion of power variations that can be captured using these events. Our best combination PMCs can produce estimates of power with an $R^2$ of 0.45-0.98 for our applications. In the next section, we improve our estimation model using time dependence of our measurements.

## VII. Time-Series

In this section, we investigate the power consumption trend from a time-series perspective. In Figure 7, we assess the degree of dependence in the data by showing the sample autocorrelation function (sample ACF) of differenced power consumption data. The horizontal lines in this figure are the 95% confidence interval for Gaussian white noise process of length $N$, where $N$ is the sample length for each application. For a Gaussian white noise process it is expected to have 95% of the ACF values (lags larger than zero) between the 95% bounds. We can see that power consumption signals for the studied benchmarks have many more than 5% ACF values outside those bounds. This shows that there is a significant relationship in time between our data samples. This suggests using a time-series approach that includes an autoregressive (AR) component might be beneficial for power modeling.

### A. ARMAX Modeling

In this section we use a recursive least squares (RLS) [17] combined with an autoregressive moving-average with exogenous inputs model (ARMAX) [17] to estimate power consumption using PMC measurements and previous power measurements, explained with details in [3].

The relationship between power and PMC events is modeled as an ARMAX model in Equation (4) where $\mathrm{P}[t]$ represents the power measurement of the system at time $t$. There are $j_{max}$ PMCs used in the model ($j_{max} = 4$). The $\mathrm{j^{th}}$ PMC values at time $t$ is shown as $c_j[t]$ ($1 \leq j \leq j_{max}$). In this model, $c_j[t-i]$, the $\mathrm{j^{th}}$ PMC measurement at time $t-i$, is linearly related to the current power consumption measurement $\mathrm{P}[t]$ via a coefficient $\alpha_{i,j}$. A past power measurement at time $t-i$, $\mathrm{P}[t-i]$, is related to the current power measurement via a coefficient $\beta_i$. The time window that Equation (4) covers includes the current and the previous $m$ PMC measurements, as well as the past $n$ power measurements. The values of $\alpha_{i,j}$ and $\beta_i$ are updated at each time step using an RLS algorithm.

$$\mathrm{P}[t] + \sum_{i=1}^{n} \beta_i \mathrm{P}[t-i] = \sum_{i=0}^{m} \sum_{j=1}^{j_{\max}} \alpha_{i,j} c_j[t-i] \qquad (4)$$

We use the above ARMAX model to estimate the power consumption of a computing node for different values of $m$ and $n$. In addition to goodness of fit, $R^2$, we report both the mean absolute error of total signal (MAETS) and the mean absolute error of dynamic signal (MAEDS). Let $x_i$ be an observation and $\hat{x}_i$ its estimated value for $n$ readings ($1 \leq i \leq n$). Let $x_{min}$ and $x_{max}$ be the minimum and maximum

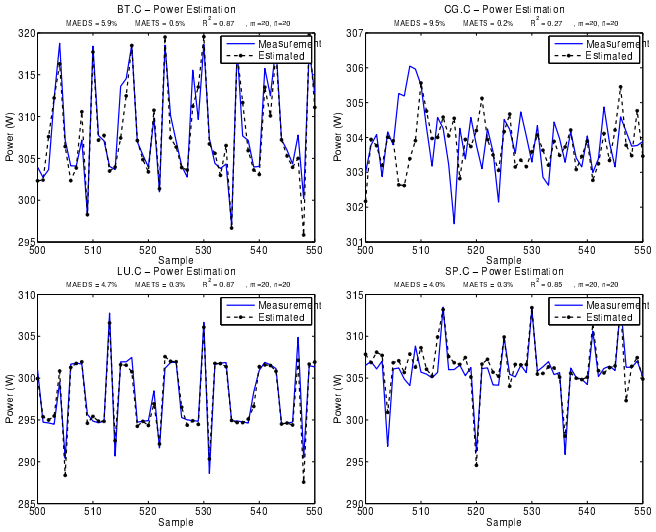


Figure 8. Power Estimation of BT, CG, LU, and SP using ARMAX(20,20)

observed values of $x_i$. MAETS and MAEDS are calculated as $\frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{x}_i - x_i|}{x_{max}}$ and $\frac{1}{n}\sum_{i=1}^{n}\frac{|\hat{x}_i - x_i|}{x_{max}-x_{min}}$, respectively.

In evaluating power consumption modeling errors, the smaller denominator of MAEDS, $x_{max} - x_{min}$, makes it a more accurate measure than MAETS. Unfortunately, many related work still report their error rates without removing the static part of the estimated signals. For example, a 3% MAETS in estimation of a power consumption signal with $x_{min} = 280$ and $x_{max} = 310$ might seem a great result. However, it might not seem so great when translated into its dynamic range and considering the fact that 280W of the signal never changes and is related to idle power. For this example, the dynamic range of signal is $x_{max} - x_{min} = 30$ and therefore its MAEDS is 31% (10.3 times larger).

We use two different sets of PMCs for each application to see their differences in estimation performance. In our first selection, we use the top four correlated single PMCs for each application from Tables II, III, IV, and V, respectively for BT, CG, LU, and SP. We refer to this selection of PMC as "Top PMCs". In our second selection, we use the obtained results from our proposed sub-space projection in section VI-B. We refer to this selection of PMCs as "Best Combination". In the following, we explore estimation efficiency of ARMAX($m$, $n$) model in four different configurations (i.e., $m, n \in \{0, 20\}$).

For a fair comparison, we disregard the first 99 samples of the trace in measuring the performance metrics (i.e., $R^2$, MAETS, and MAEDS), as some of the scenarios can produce estimates only after the $20^{\text{th}}$ sample. A summary of the results in this section is provided in Table VII.

*1) ARMAX (20,20):* This model uses the last 20 power and PMC readings, in addition to the current PMC reading. It achieves an excellent result of 0.2%-0.5% estimation error (MAETS) for the best combination selection of PMCs. Both

Table VII
$R^2$, MAEDS, AND MAETS OF POWER ESTIMATION USING AN ARMAX WITH DIFFERENT PARAMETERS ($m$, $n$)

| *App.* | *Top* | *Best* | *Top* | *Best* | *Top* | *Best* |
|---|---|---|---|---|---|---|
| $(20, 20)$ | $R^2$ | | $DS\%$ | | $TS\%$ | |
| $BT.C$ | 0.88 | 0.87 | 6.0 | 5.9 | 0.4 | 0.5 |
| $CG.C$ | 0.36 | 0.27 | 6.3 | 9.5 | 0.1 | 0.2 |
| $LU.C$ | 0.84 | 0.87 | 5.1 | 4.7 | 0.4 | 0.3 |
| $SP.C$ | 0.79 | 0.85 | 5.3 | 4.0 | 0.3 | 0.3 |
| $(0, 0)$ | $R^2$ | | $DS\%$ | | $TS\%$ | |
| $BT.C$ | $-0.86$ | 0.31 | 27.5 | 15.8 | 2.0 | 1.2 |
| $CG.C$ | $-1.77$ | $-55.03$ | 12.4 | 58.4 | 0.2 | 1.1 |
| $LU.C$ | $-10.33$ | $-7.45$ | 44.0 | 49.6 | 3.1 | 3.5 |
| $SP.C$ | $-15.51$ | 0.08 | 54.2 | 11.2 | 3.4 | 0.8 |
| $(20, 0)$ | $R^2$ | | $DS\%$ | | $TS\%$ | |
| $BT.C$ | 0.88 | 0.87 | 5.9 | 6.0 | 0.4 | 0.5 |
| $CG.C$ | $-0.04$ | $-1.48$ | 8.1 | 18.4 | 0.2 | 0.4 |
| $LU.C$ | 0.85 | 0.87 | 4.7 | 4.4 | 0.3 | 0.3 |
| $SP.C$ | 0.70 | 0.77 | 6.2 | 4.9 | 0.4 | 0.3 |
| $(0, 20)$ | $R^2$ | | $DS\%$ | | $TS\%$ | |
| $BT.C$ | 0.80 | 0.79 | 7.9 | 7.7 | 0.6 | 0.6 |
| $CG.C$ | 0.30 | 0.09 | 6.4 | 10.5 | 0.1 | 0.2 |
| $LU.C$ | 0.81 | 0.87 | 5.6 | 4.6 | 0.4 | 0.3 |
| $SP.C$ | 0.78 | 0.80 | 5.1 | 4.3 | 0.3 | 0.3 |

the top PMC selection and the best combination selection perform well and their results are close. For CG, $R^2$ is 0.36 and 0.27 in case of top PMC and best combination selections, respectively. For BT, LU, and SP, $R^2$ remains between 0.79 and 0.88 for both selections. The low $R^2$ value for CG is assumed to be due to its "flat" power curve and meaning its average is a better estimator. The MAETS and MAEDS confirms that errors are small for CG (MAEDS of 6.3% and 9.5%, and MAETS of 0.1% and 0.2%). We present some of the observed and estimated power consumption of BT, CG, LU, and SP applications in Figure 8 .

*2) No time dependence (0,0):* This model uses only the current sample PMC measurements (no previous power/PMC). This is an extreme case for the purpose of comparison to other scenarios and to understand estimation efficiency without looking at the past values. Except for BT with the best combination selection of PMCs, the other applications for either selection of PMCs do not provide a good $R^2$. The dynamic range errors (MAEDS) for either selections of PMCs are between 11% and 59%. It is important to notice, as a *misleading measure*, that even with such a poor estimation performance, the total signal error average (MAETS) is in the range of 0.2%-3.4% and 0.8%-3.5% for top PMCs selection and best combination PMC selection, respectively.

*3) Moving average with exogenous inputs (20,0):* This model uses the past 20 PMC measurements and no previous

power measurement. Except for CG, the $R^2$ measure shows a great estimation performance and its performance is close to ARMAX(20, 20) model. The dynamic range average estimation error (MAEDS) range is 4.7%-8.1% and 4.4%-18.4% for the top PMC selection and best combination PMC selection, respectively. The total signal average estimation error (MAETS) range is 0.2%-0.5% overall for either PMC selections. Both MAETS and MAEDS metrics are close to the performance metrics of ARMAX(20,20).

*4) Autoregressive model with current PMC (0,20):* This model uses only the current PMC measurement and the past 20 power measurements. The $R^2$ metric performs very close to ARMAX(20,20) and ranges between 0.09-0.30 for CG and 0.78-0.87 for BT, CG, and LU. The difference in MAEDS metric between ARMAX(0,20) and ARMAX(20, 20) is less than 2.0% for all the applications/selections. Similarly, the difference of MAETS metrics are less than 0.3%.

In short, using time dependence between power and/or PMC measurements significantly improves the estimation efficiency. Furthermore, error calculations based on total power signal can be misleading and hide model defficiencies, therefore we suggest using metrics such as MAEDS.

## VIII. Related Work

Bellosa [18] has studied the correlation of PMC events on Pentium II for operating-system-directed power management. Bircher et al. [4] have studied correlation coefficients of 23 PMCs with power consumption of a Pentium 4 processor (single-PMC study). They have found instruction per cycle (IPC) metrics (in particular *uops fetched per cycle*) useful in power estimation techniques using linear regression models to estimate the power consumption of a Pentium 4 processor. Lively et al. [10] have developed application-centric PMC-based models for performance and power consumption. Isci et al. [16] have studied the impact of real-system variability on detecting recurrent phase behavior.

## IX. Conclusion

This paper studies single and multi-PMC selection methods for power-PMC modeling purposes. We have provided application specific single-PMCs that are most correlated with power for BT, CG, LU, and SP applications (class C), as well as a unified group of PMCs for all of our applications. Our multi-PMC selection method provides the least redundant PMC selection without facing obstacles faced by PCA-based methods, or their shortcoming in accuracy. We have provided the best combination of PMCs for our applications using our method. In addition, we have shown the time-domain dependence of data in power-PMC models. Utilizing this characteristic, we have presented that ARMAX models can produce power estimates with a mean absolute error in total signal of 0.1-0.5% in our applications.

## References


[1] "HP ProLiant Intel-based 300series G6 and G7 servers," Hewlett-Packard, Tech. Rep., 2010, TC100403TB.

[2] "IBM PowerExecutive," http://www.ibm.com.

[3] R. Zamani and A. Afsahi, "Adaptive estimation and prediction of power and performance in high performance computing," *Computer Science - Research and Development*, vol. 25, pp. 177–186, 2010.

[4] W. L. Bircher, M. Valluri, J. Law, and L. K. John, "Runtime identification of microprocessor energy saving opportunities," in *Intl. Symp. on Low power electronics and design*, 2005, pp. 275–280.

[5] Y. Cho, Y. Kim, S. Park, and N. Chang, "System-level power estimation using an on-chip bus performance monitoring unit," in *ICCAD '08: Proc. of the 2008 IEEE/ACM Intl. Conf. on Computer-Aided Design*, 2008, pp. 149–154.

[6] G. Contreras and M. Martonosi, "Power prediction for Intel XScale processors using performance monitoring unit events," in *ISLPED '05: Proc. of the 2005 Intl. Symp. on Low power electronics and design*. ACM, 2005, pp. 221–226.

[7] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *MICRO 36: Proc. of the 36th annual IEEE/ACM Intl. Symp. on Microarchitecture*. IEEE Computer Society, 2003, p. 93.

[8] T. Li and L. K. John, "Run-time modeling and estimation of operating system power consumption," *SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 160–171, 2003.

[9] K. Rajamani, H. Hanson, J. C. Rubio, S. Ghiasi, and F. L. Rawson, "Online power and performance estimation for dynamic power management," Tech. Rep., 2006.

[10] C. Lively, X. Wu, V. Taylor, S. Moore, H.-C. Chang, C.-Y. Su, and K. Cameron, "Power-aware predictive models of hybrid (mpi/openmp) scientific applications on multicore systems," *Computer Science - Research and Development*, pp. 1–9.

[11] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Boston, USA: Addison-Wesley Longman, 2005.

[12] "The OpenMP API," www.openmp.org.

[13] "The NAS Parallel Benchmarks," www.nas.nasa.gov.

[14] M. Pettersson, "Linux performance counters driver," http://user.it.uu.se/~mikpe/linux/perfctr.

[15] "hwpmc - Linux performance counters driver," FreeBSD manual 4, http://www.freebsd.org.

[16] C. Isci and M. Martonosi, "Detecting recurrent phase behavior under real-system variability," in *Workload Characterization Symp., 2005. Proc. of the IEEE Intl.*, 2005, pp. 13 – 23.

[17] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. New York, NY, USA: John Wiley & Sons, Inc., 1996.

[18] F. Bellosa, "The benefits of event: driven energy accounting in power-sensitive systems," in *Proc. of the 9th workshop on ACM SIGOPS European workshop*. ACM, 2000, pp. 37–42.