Adaptive Power monitoring for self-aware embedded systems

Mohamad El Ahmad, Mohamad Najem, Pascal Benoit, Gilles Sassatelli, Lionel Torres

HAL Id: lirmm-01257519

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01257519v1

Submitted on 17 Jan 2016

Adaptive Power Monitoring For Self-Aware Embedded Systems

Mohamad El Ahmad, Mohamad Najem, Pascal Benoit, Gilles Sassatelli, Lionel Torres

LIRMM - UMR CNRS 5506 - University of Montpellier

Montpellier, France

{*mohamad.elahmad, mohamad.najem, pascal.benoit, gilles.sassatelli, lionel.torres*}@lirmm.fr

Abstract—**Dynamic Thermal and Power Management methods require efficient monitoring techniques. Based on a set of data collected by sensors, embedded models estimate online the power consumption: this task is a real challenge, since models must be both accurate and low cost, but they also have to be robust to variations. In this paper, we investigate a self-aware approach using the performance events and the external temperature. We present a solution (PESel) for the selection of the relevant information. This method is two times faster than existing solutions and provides better results compared to related works. The power models achieve a 96% accuracy with a temporal resolution of 100 ms, and negligible performance/energy overheads (less than 1%). Moreover, we show that our estimations are not sensitive to external temperature variations.**

I. Introduction

ACHIEVING high energy efficiency is a major challenge in the design of integrated circuits, which is confronted with the problem of delivering high performance with a limited power budget. Therefore, modern day microprocessors provide dynamic thermal and power management techniques (DTPM) to address this challenge (*e.g.* scaling the voltage and the frequency, switching to low-power modes, scheduling of tasks, *etc.*). Several techniques use the information of the power consumption to adapt the system behavior (reactive techniques) or to predict undesired future states (proactive techniques). In all cases, the effective management of power and temperature depends critically on the monitoring method, which should provide robust and accurate estimations in a cost-effective way.

Modern systems integrate a specific unit connected to several events for the debugging and profiling of the performance. This unit is called Performance Monitoring Unit (PMU) in ARM processors, and Performance Monitoring Counters (PMCs) in Intel ones; the first notation will be used in the rest of this paper. The events that can be monitored in such system can be decomposed into three main categories: (*i*) local events occurring at the hardware-level inside each core (*e.g.* L1 instruction misses), (*ii*) shared events occurring at the hardware-level on the shared resource inside the cluster (*e.g.* L2 cache accesses), and (*iii*) system events available at the operating system (*e.g.* a task migration). The PMU has a dedicated hardware counter to monitor the occurrence of both local and shared events, while the system event counters are implemented in a software layer managed by a firmware. Indeed, it is impossible to integrate a particular hardware counter for each hardware-level event. That is the reason why, PMUs integrate a few numbers of configurable hardware counters, smaller than the number of available events. It thus raises the question of how the PMU can be configured and used to appraise the global system activity that affects the overall power consumption P_{total}, which is due to the dynamic and static power.

In this paper, we investigate low-cost power models, robust to temperature variations. We use the information from the PMU and an external temperature sensor to track any variation in the power consumption. There are thousands of events at system-level that can be monitored by the PMU [1]. So, if one wants to profile these events, one will easily face a problem of managing a big quantity of data. For this purpose, we develop the algorithm PESel (Performance Events Selection) inspired from Data Mining techniques for the selection of relevant events for power modeling. Furthermore, linear and neural network models are extended taking into account the external temperature to build an efficient monitoring approach.

The remainder of the paper is structured as follows. In Section II the limitations of most relevant power estimation and event selection techniques are discussed to highlight the need for this work. Sections III and IV present the proposed selection method and power models. Section V describes the experiments and in Section VI we draw some conclusions.

II. Related works

Several works assume that either the internal activity, or the ambient temperature are constant, while modeling the power consumption. In [2], when estimating the power consumed by the ARM big.LITTLE platform, the internal activity is supposed to be constant. In [3], a few performance events were manually selected based on a previous knowledge about the system behavior, to build a simple linear model. In [4], the authors present a power model based on dynamic and static contributions, but there is no mention on the impact of the external temperature. The selection of events is a problem addressed in [5], using the Pearson correlation criterion, or the Spearmans rank correlation in [6]. A projection method based on the principle component analysis (PCA) is proposed in [7] that takes into account the relation between the elements before the selection of performance events for the Dell PowerEdge Opteron processor.

This paper proposes a new algorithm for the selection of events (PESel), which is faster than existing works [7], and takes into account the parallel execution of the applications by the multiprocessor platforms. The originality of our algorithm comes from the consideration of the interactions between events before the selection, that were not taken into account classical statistical methods such as in [5], [6]. Furthermore, our cost-effective power models are robust to temperature variations while tracking the consumption.

III. Performance Events Selection

The main goal is to produce a robust and accurate power model that is able to track power variations for any activity at any ambient temperature. Here, we propose to use the performance events from the PMU to appraise the global system activity. But, due to the limited number of hardware counters, the most relevant events for power modeling purposes must be selected. For example, the ARM CortexA9 processor has 6 hardware counters, and consequently, must be configured with 6 among the 62 available local events. Furthermore, it is not possible to construct a unique database containing all available events with the power under a unique time-sample domain, limiting the utilization of the standard selection methods such as PCA analysis, features selection, *etc*. For this reason, we developed a sub-space method inspired from the features selection algorithms from Data Mining. It greedily searches for the best set of events, and uses the Correlation Features Selection (CFS) metric (*merit*) [8] for the evaluation shown in the following equation:

$$M = \frac{p \times \overline{r}_{E_i,p}}{\sqrt{p + p \times (p-1) \times \overline{r}_{E_i,E_j}}} \tag{1}$$

where p is the number of events in the candidate subset, E_i and E_j are two events in the candidate subset, $\overline{r}_{E_i,p}$ is the average correlation between each event E_i and the power P, and $\overline{r}_{E_i,E_j}$ is the average correlation among events in the candidate subset. The subset having the highest merit is the best solution found, and corresponds to a set of events uncorrelated with each other, which highly correlate with the power.

The proposed method requires several iterations (k), each one having two main steps. At the end of each iteration, only one event is selected among the resulting available events. In the first step, at the iteration k, the $k-1$ selected events will occupy several hardware counters. Therefore, the remaining available counters are used to configure the remaining available events. For this purpose, several runs are required to finish this step at iteration k. A subset C of events with the corresponding power P is generated for each run.

The second step is to use the proposed $PESel$ algorithm to identify the most relevant event at the iteration k (e_k). The pseudo-code of the proposed algorithm is shown in the Algorithm 1. It operates on all subsets C generated previously for each run to find the best event e_k at the current stage k. Let C_{cand} be the set of events candidate extracted from C. For each C_{cand}, we measure the impact of each event candidate e over the solution S_{k-1} found at the stage $k-1$. This evaluation is done by the calculation of the merit of the union between S_{k-1} and e. Then, the event with the highest merit is selected. At the first iteration $k=1$, where $S_0 = \emptyset$, *PESel* selects the event the most correlated with the power, as long as the equation (1) is reduced to a simple correlation for $p = 1$. At the end of the stage k, the event found by the proposed $PESel$ algorithm is added to the solution pool: $S_k = S_{k-1} \cup \{e_k\}$, and $k = k + 1$.

IV. Power Consumption Models

The total power consumption can be calculated as:

```
input   : S_{k-1} and D_k set of simultaneous measurement
output  : e_k best event found at the end of the stage k
M_Best = 0;
e_k = {};
for C in D_k do
    C_cand ← C \ S_{k-1};
    for e in C_cand do
        L_e ← S_{k-1} ∪ {e};
        M = Merit(L_e);
        if M > M_Best then
            M_Best = M;
            e_k ← e;
        end
    end
end
return e_k;
```

Algorithm 1: PESel Algorithm

$$P_{total} = P_{dyn} + P_{stat} = (P_{sw} + P_{sc}) + P_{stat}$$
$$= (\alpha \cdot C \cdot V_{dd}^2 \cdot f + V_{dd} \cdot I_{sc}) + V_{dd} \cdot I_{leakage} \tag{2}$$

Where α is the activity factor, C is the switching capacitance, V_{dd} is the power supply, f is the frequency and I_{sc} is the short-circuit current. This one occurs when pull-up and pull-down networks are conducting simultaneously. In the rest of this paper, the short-circuit term is neglected.

The dynamic power component dominates during the active mode of the system, while the static power accompanies the total power as long as the system is powered on. The static component increases exponentially with the chip temperature (T_{chip}). This temperature dependence can result in a positive feedback loop, because the temperature is also dependent on power consumption. With this assumption, the static power (P_{Stat}) can be expressed as:

$$P_{static} = P_0 \cdot e^{-k/T_{chip}} \tag{3}$$

where P_0 and k are process dependent constants. After that, we can write the updated equation for the power dissipation as follows:

$$P_{total} = \alpha \cdot C \cdot V_{dd}^2 \cdot f + P_0 \cdot e^{-k/T_{chip}} \tag{4}$$

According to [9], the external temperature T_{ext} (room temperature) and T_{chip} (operating temperature) may be found in a same equation by using the equivalent RC circuit for modeling the temperature, and it can be written as follows:

$$T_{chip} = T_{ext} + R_\theta \cdot P_{Total} \tag{5}$$

where R_θ is the equivalent thermal resistance of the package ($^\circ C/W$). This equation (5) clearly shows that a change of the ambient temperature has a direct impact on T_{chip}. This highlights the need to account for this parameter in the power models for tracking the consumption.

A. Linear Approximation

In mathematics, a functional equation not identically null where $\mathscr{F}:]0, +\infty[\Rightarrow \mathbb{R}$ like Cauchy function can verify the following property $\mathscr{F}(xy) = \mathscr{F}(x) + \mathscr{F}(y)$ for all x and y strictly positive. This formal definition can be applicated on the equation (4), since the power domain meets the interval of $\mathscr{F}$ and will never be null or negative when the system is powered on. We can apply this property to the representation of P_{Dyn}, P_{Stat} and P_{Total}. The total power can than be approximated as:

$$P_{Total} = P_0 + \sum_{i=1}^{N} w_i \cdot e_i + w_f \cdot f + K \cdot T_{chip} \tag{6}$$

where P_0 is a constant that corresponds to the idle power consumption, w_i is the weight of the contribution of the event e_i on power variation. T_{chip} corresponds to the chip temperature which results from P and T_{ext}, K is a process dependent constant and w_f is the weight of the frequency. In the subsequent experiments, the DVFS mechanism was deactivated in order to put the focus on the temperature impact on power consumption. For this reason, the terms $w_f.f$ can be considered here as a constant term.

B. Neural Network Estimation

Modeling the power with a highly reduced subset of events might be innaccurate with the afore mentioned linear approximations. Neural network modeling could be an interesting alternative to investigate. For this purpose, multilayer feed forward neural networks model which has been extensively used in data mining for supervised learning could be suitable to mode non-linear behaviors.

Neural network consists of neurons, that are arranged in layers. The number of neurons at the first layer is equal to the length of the input set (events and temperature), while the output layer contains only one neuron that produces the estimation of the power. The layers between have a configurable number of neurons, and each neuron has direct connections to all neurons of the subsequent layer. In general, the hidden layer determines the reliability of the model and the number of neurons is usually equal to the number of inputs [3].

V. EXPERIMENTS

In this section, we evaluate the proposed power models on the SKY-S9500-ULP-CXX snowball PDK development kit . It includes a Cortex A9 ARM dual-core processor integrated into the NovaTMA9500 chip from ST-Ericsson. The on-chip PMUv2 has 6 configurable counters for each cores for local events, one counter per core for the clock cycles and 2 configurable counters for the L2 cache events. In order to vary the system activity, we run on both cores several software benchmarks, including MiBench, Whetstone and Linpack. These one target different application fields, including: (*i*) performance and data management, (*ii*) automotive and industrial control, (*iii*) office and security programs and (*iv*) network and telecom. We carried out the experiments using the Streamline Performance Analyzer of DS-5 v5.21 development studio and the Energy probe provided by ARM for the measurement of the power consumption. DS-5 collects the number of occurrences of events counted by the PMU and the measured power consumption, at $1ms$ data sampling period.

A. Accuracy vs. Performance Events

We first aim at evaluating the proposed *PESel* method. Benchmarks were split into two sets of applications: 70% for the training of the models and 30% for the test and measurement of the errors. We applied *PESel* to the set of training applications and we found the 11 most significant events for power modeling.

We compare *PESel* with the Neural Network (NN) and the Linear Model (LM), to three other methods from the literature that use the LM model to estimate the power. The number of selected events is varied from 1 to 11 considering for each case the best selected events according to each method.
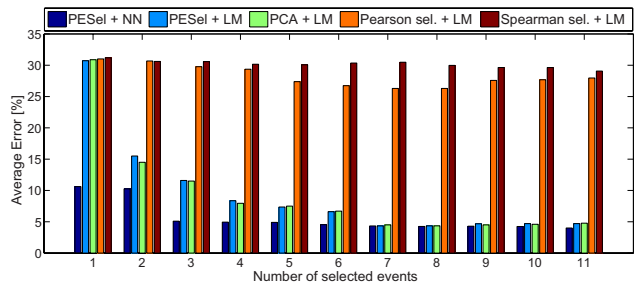

Fig. 1: comparison of the selection methods for different selected subsets

Figure 1 depicts the average for each models. The worst selection is the one that uses the classical statistical correlations (Pearson [5] and Spearman [6] correlations). The PCA based selection in [7] produces a solution closer to the one selected by *PESel* with the LM model. It must also be noticed that *PESel* is two times faster than PCA (129.8s compared to 226.22s at stage $k = 11$). The NN model is clearly more accurate than LM especially when the number of events is lower than 6. The percentage of average error of NN constructed with the 11 events is about 4.85% at a data sampling period equal to $100ms$ and the measured coefficient of determination (R^2) is about 0.887. The obtained error decreases when the data sampling period increases; the average error of NN reaches 3.2% ($R^2 = 0.9084$) compared to 4.1% ($R^2 = 0.8912$) for a sampling period equal to 1 sec. For this purpose, the NN model is chosen to model the power consumption in the rest of this paper.

It can be noticed that the error achieved in [10] at a timing resolution equal to 500 ms is about 8.2% and 10.0% for the small core and big core respectively, of the ARM big.LITTLE processor, while in [11] is about 3% by application.

B. Temperature Impact

For the external temperature, the board was placed in a thermal chamber where the system runs through a series of arbitrary temperatures (from -20 $°C$ to +80 $°C$). Several experiments are therefore conducted under different thermal conditions. The power consumed over time and the corresponding external temperatures are depicted in Fig.2. Plot(a) of this figure shows the reference power consumed by the system at each external temperature. Since the transient phases at each change in the temperature are very long (10min), they were not represented in this figure. Plot(b) shows that for the same applications scenario running at different external temperatures, there is a significant impact on the power consumed, as expected (both on average and amplitude). Since an increase in the external temperature favors the self-heating phenomena. This motivates the need to build a power model robust to temperature variations.

C. Power Tracking

In this experiment, we aim to assess and compare two power models: the first one NN_{20} was generated under normal conditions of temperature (+20°C), while the second NN_R was trained with external temperature variations (from -20°C to +80°C) and includes T_{ext} as a model variable. The Fig.2 (both plot (c) and (d)) depicts the power consumed tracked by both models at different temperatures. At +20°C (plot (c)), NN_{20} produces a good estimation of
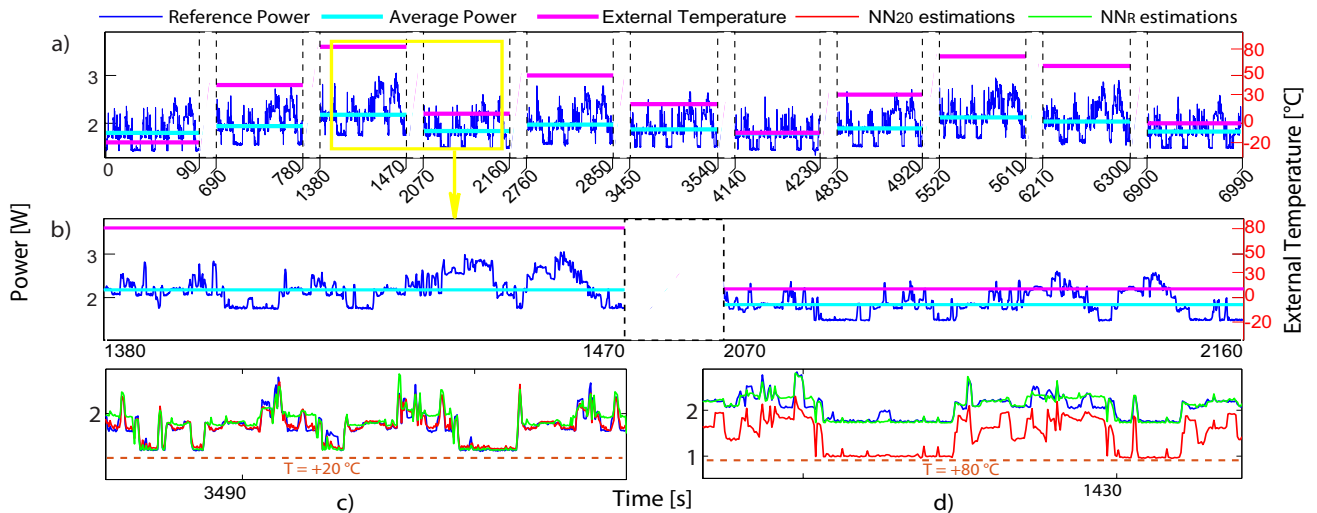
Fig. 2: Reference Power consumed at each external temperature (a) and (b); The tracking of power with both NN_R and NN_{20} models (c) and (d).

the consumption, as expected. Despite small discrepancies, the NN_R model also provides a precise assessment. When switching to $+80°C$ (plot(d)), we observe that the NN_R is still able to accurately follow the power variations, while the NN_{20} clearly not.

D. Overhead

Our methodology allows an online estimation, which is simply based on a set of logic events and an external temperature sensor: we have shown that our solution is able to track the total power consumption with a good accuracy. However, it also requires some processing ressources, which need to be examined. Both models, NN_R and NN_{20}, were programmed and executed by the processor at 1GHz. The maximum overhead is achieved with the 11 selected events, from which the time (t_{NN}) taken by the processor to compute the calculation of the NN_R and NN_{20} equations are equal to 8.036 μs and 7.962 μs respectively.
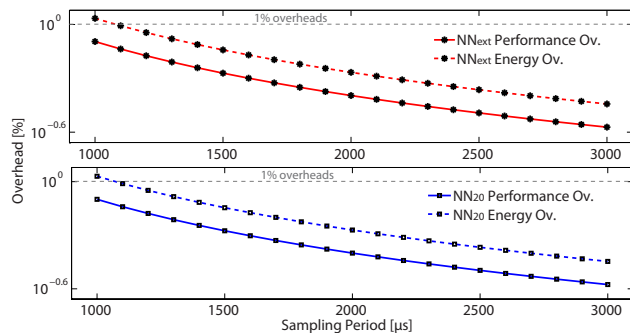

Fig. 3: The Perfomance and energy overhead of both NN_R and NN_{20} models for different sampling periods

The power consumed (P_{NN}) during this computation is 2.02W. The energy overhead is approximated as the ratio between the extra energy ($t_{NN} \times P_{NN}$) and the minimum energy at idle state (worst case scenario). The performance overhead is the ratio between the computation time (t_{NN}) over the data sampling period (T_s). Energy and performance overheads for both NN_R and NN_{20}, at different sampling periods, are depicted in the Fig.3 introducing the temperature into the NN model has a negligible impact on both overheads. In all cases, the overhead is very low and is almost less than 1%.

VI. CONCLUSION

We have proposed a method robust to temperature variations that can be exploited for the run-time monitoring of the power consumption in modern computing systems. This approach appraises the total power in terms of performance events taking into account the external temperature, thanks to the proposed PESel algorithm. The results show the efficiency of the solution, which selects a few events faster than existing works. Moreover, the power models achieve a very good accuracy ($<4\%$), with a low performance and energy overheads ($<1\%$).

REFERENCES

[1] V. Salapura, K. Ganesan, A. Gara, M. Gschwind, J. C. Sexton, and R. E. Walkup, "Next-generation performance counters: Towards monitoring over thousand concurrent events," *ISPASS 2008 - IEEE International Symposium on Performance Analysis of Systems and Software*, pp. 139–146, 2008.

[2] G. Singla, G. Kaur, A. K. Unver, and U. Y. Ogras, "Predictive Dynamic Thermal and Power Management for Heterogeneous Mobile Platforms," pp. 960–965, 2015.

[3] S. Tamura, M. Tateishi, M. Matumoto, and S. Akita, "Determination of the number of redundant hidden units in a three-layered feedforward neural network," *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, vol. 1, pp. 335–338, 1993.

[4] R. Nath, D. Carmean, and T. S. Rosing, "Power modeling and thermal management techniques for manycores," *2013 IEEE Symposium on Computers and Communications (ISCC)*, pp. 000740–000746, Jul. 2013.

[5] W. Bircher, M. Valluri, J. Law, and L. John, "Runtime identification of microprocessor energy saving opportunities," *ISLPED '05. Proceedings of the 2005 International Symposium on Low Power Electronics and Design, 2005.*, pp. 275–280, 2005.

[6] 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*, vol. 27, no. 4, pp. 245–253, 2012.

[7] R. Zamani and A. Afsahi, "A study of hardware performance monitoring counter selection in power modeling of computing systems," *2012 International Green Computing Conference, IGCC 2012*, 2012.

[8] M. A. Hall and G. Holmes, "Benchmarking attribute selection techniques for discrete class data mining," *IEEE Trans. on Knowl. and Data Eng.*, vol. 15, no. 6, pp. 1437–1447, Nov. 2003.

[9] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. E. I. Huang, S. Velusamy, and D. Tarjan, "Temperature-Aware Microarchitecture : Modeling and Implementation," vol. 1, no. 1, pp. 94–125, 2004.

[10] M. Pricopi et.al., "Power-performance modeling on asymmetric multi-cores," *2013 International Conference on Compilers, Architecture and Synthesis for Embedded Systems, CASES 2013*, 2013.

[11] W. Choi, H. Kim, W. Song, J. Song, and J. Kim, "ePRO-MP : A tool for profiling and optimizing energy and performance of mobile multiprocessor applications," vol. 17, pp. 285–294, 2009.