Blind Identification of Power Sources in Processors

Sherief Reda
School of Engineering
Brown University, Providence, RI 02912
Email: sherief_reda@brown.edu

Adel Belouchrani
Electrical Engineering Department/LDCCP
Ecole Nationale Polytechnique, Algiers, Algeria
Email: adel.belouchrani@enp.edu.dz

***Abstract*—The ability to measure power consumption is at the heart of power and thermal management techniques. Modern processors are equipped with hardware monitoring mechanisms that can measure total power. However, this lumped measurement is not sufficient if there is a need to execute fine-grain thermal and power management techniques. This paper proposes a new direction for identifying the fine-grain sources of power consumption in many-core processors. For the first time, we show that it is possible to simultaneously identify both the power consumption of different cores and the thermal model of the chip from just the measurements of the thermal sensors and the total power consumption measurement. Our identification technique is blind as it does not require design knowledge of the thermal model to identify the power sources. Furthermore, our technique makes no use of the performance counters, which reduces its overhead, and works seamlessly with dynamic voltage and frequency scaling. We implement our technique on a real multi-core CPU-GPU processor-based system, and we show the ability to identify the runtime power consumption of the individual cores using just the total power measurement and the measurements of the thermal sensors under different workloads. We also verify the superior accuracy of our approach using results from a controlled simulation environment.**

I. Introduction

To enable correct thermal and power management, it is necessary to measure physical metrics such as power consumption and temperatures of various cores of the processor. Modern processors use the running average power limit (RAPL) interface to enable applications to measure the power consumption [9]; however, these measurements are typically coarse-grain, only giving the power consumption of all cores, uncore units and total package power. Thermal sensor measurements, however, are usually available on a per-core basis. The goal of this paper is to blindly estimate the power consumption of individual cores from just the total power consumption of all cores and the measurements of the thermal sensors.

Identifying the power consumption of the individual cores will enable a new class of fine-grain power management systems that is able to determine the true impact of dynamic voltage and frequency scaling (DVFS) and scheduling techniques on the power consumption of individual cores. In this paper we propose a *blind* identification approach in which no design time information is required to estimate the power consumption of the cores. The standard approach for thermal and power modeling of many-core processors is the state-space model [1], [6], [15], [16]:

$$\mathbf{t}(k) = \mathbf{A}\mathbf{t}(k-1) + \mathbf{B}\mathbf{p}(k) + \epsilon, \tag{1}$$

where $\mathbf{t}(k)$ and $\mathbf{p}(k)$ are vectors that denote the temperature and power consumption measurements of the cores time k respectively, $\mathbf{A}$ and $\mathbf{B}$ are the two modeling matrices that capture the physical relationship between power and thermal, and ϵ is a vector that represents the noise in the measurement process. Note that if one knows the matrices $\mathbf{A}$ and $\mathbf{B}$, then we can recover the individual powers of the cores over time (i.e., $\mathbf{p}(k)$) relatively easily by just using the measurements of the thermal sensors and applying inversion techniques [7]. However, these matrices require design-time information that is specific to each processor model, and this information is not necessarily available to the users. While one may identify these matrices from on-line measurements [4], these measurements would require either direct knowledge of $\mathbf{p}(k)$ or indirect *a priori models* (e.g., using performance counters) for $\mathbf{p}(k)$ [12]. To summarize, we can say that in previous work, researchers assumed either (1) the availability of $\mathbf{p}(k)$ and sought to identify $\mathbf{A}$ and $\mathbf{B}$ [4], [6], [15], or (2) the availability of $\mathbf{A}$ and $\mathbf{B}$ and sought to identify $\mathbf{p}(k)$ [7].

In contrast to previous work, we seek blind identification of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{p}(k)$, with no assumption or need for any prior design-based models for any of them. That is, our methodology only uses runtime measurements (i.e., total power and thermal sensors measurements) to simultaneously identify $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{p}(k)$. The contributions of this paper are as follows.

- We formulate the *blind power identification* (BPI) problem to estimate the power consumption of individual cores in a multi-core processor together with the processor's thermal model using only the total power measurements and the thermal sensor measurements.
- Existing general blind identification methods suffer from their inability to determine the exact permutation of power estimates, and their estimates can be off by a constant factor [2]. To eliminate these ambiguities, we devise a novel methodology for BPI that exploits the physical characteristics of thermal transfer to provide a unique solution that is consistent with the measurements. Our method handles seamlessly steady-state and transient operation, enabling users to track power consumption during runtime. In addition to the identification of the power of individual cores, a natural byproduct of our methodology is the thermal model that links power consumption and thermal characteristics.
- We implement our methodology on a real quad-core CPU+GPU processor and use it to estimate the power consumption of its cores under various standard benchmarks over time. We also verify the accuracy of our method in a controlled simulation environment.

978-3-9815370-8-6/17/$31.00

The organization of this paper is as follows. In Section II we review related work. We give our problem formulation in Section III and our proposed methodology in Section IV. Our experimental results are provided in Section V. The main conclusions of this work are summarized in Section VI.

II. Related Work

To identify the state-space model that links temperatures and power, there are two general approaches: a *design time* approach and a *runtime* approach. The design-time approach requires extensive information of the layout of the chip and its package characteristics [11]. The design-time approach requires the transfer of the state-space models, which are processor specific, to the users to be deployed during runtime. This approach could be prone to errors due to variabilities arising from manufacturing and ambient conditions.

The runtime approach identifies the state-space models from physical measurements during runtime. The processor is treated as a gray or black box and machine learning or system identification techniques are used to identify the state-space models from the thermal sensor measurements [1], [4], [8], [14], [16]. A key assumption in all previous runtime modeling approaches is that there are sensors for the power sources [1], [4], [14], [16]. However, modern processors lack fine-grain power sensors. For instance, the RAPL interface provides the total power consumption for individual domains (e.g., all cores and uncore units), but it does not provide power measurements for the individual cores [9]. Beneventi *et al.* developed a regression-based model to estimate the power consumption of individual cores assuming that when a core is active, it is fully busy running at the maximum instructions per cycle [3]. This method does not work well in practice because (1) workloads have a large impact on power consumption, (2) modern processors automatically adjust the voltage and frequency depending on the number of active cores, and (3) per-core leakage power increases when more cores are activated because of thermal coupling.

In this paper we make no assumptions about the availability of power sensors or prior power models. We simultaneously identify the power sources and the thermal models, while working seamlessly under various processor frequencies. Our technique enables designers to simplify the number of sensors by eliminating the need for physical power sensors, and to instead use measurements of the thermal sensors and total power to derive per-core power consumption. Modern processors rely on internal micro-controllers to collect the measurements of the sensors and to orchestrate thermal and power management decisions [17]. Our technique can be implemented to run on internal micro-controllers or as a software thread on the main processor.

III. Problem Formulation

The transient power estimation problem consists of first identifying the matrices $\mathbf{A}$ and $\mathbf{B}$ of the model (1), then use them latter to get the power profiles $\mathbf{p}(k)$. While matrix $\mathbf{A}$ could be estimated from natural response of the processor; i.e., the measured temperatures when $\mathbf{p}(k) = \mathbf{0}$, the estimation of matrix $\mathbf{B}$ needs the use of the steady-state operation, where $\mathbf{t}(k) = \mathbf{t}(k-1)$, which reduces model (1) to

$$\mathbf{t}(k) = \mathbf{A}\mathbf{t}(k) + \mathbf{B}\mathbf{p}(k) + \epsilon \tag{2}$$

leading to

$$\begin{aligned} \mathbf{t}(k) - \mathbf{A}\mathbf{t}(k) &= \mathbf{B}\mathbf{p}(k) + \epsilon & (3)\\ (\mathbf{I} - \mathbf{A})\mathbf{t}(k) &= \mathbf{B}\mathbf{p}(k) + \epsilon & (4)\\ \mathbf{t}(k) &= (\mathbf{I} - \mathbf{A})^{-1}(\mathbf{B}\mathbf{p}(k) + \epsilon) & (5)\\ \mathbf{t}(k) &\approx \mathbf{R}\mathbf{p}(k) & (6) \end{aligned}$$

where $\mathbf{R} = (\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ is the *thermal transfer matrix* and $\mathbf{I}$ the identity matrix. If $\mathbf{R}$ and $\mathbf{A}$ are identified, matrix $\mathbf{B}$ can be calculated by

$$\mathbf{B} = (\mathbf{I} - \mathbf{A})\mathbf{R}. \tag{7}$$

Note that the model of equation (6) is similar to the one commonly used in array signal processing, particularly in blind source separation [2]. The latter consists of blindly identifying the matrix $\mathbf{R}$, i.e. by resorting only tn the information carried by the measured temperatures. Before proceeding, it is important to specify the notion of blind identification.

Challenges in Blind Identification. In the blind context, a full identification of the matrix $\mathbf{R}$ from model (6) is impossible because the exchange of a fixed scalar factor between a given source signal (a power source) and the corresponding column of $\mathbf{R}$ does not affect the observations (i.e., thermal measurements), as is shown by the following relation:

$$\mathbf{t}(k) = \mathbf{R}\mathbf{p}(k) = \sum_{n=1}^{N} \frac{\mathbf{r}_n}{\alpha_n}\alpha_n p_n(k), \tag{8}$$

where α_n is an arbitrary factor, $\mathbf{r}_n$ denotes the n-th column of $\mathbf{R}$, and N denotes the number of cores. Note also that labelling in the sum of equation (8) is arbitrary. Hence the blind identification of $\mathbf{R}$ can be performed up to permutation and scaling factor of its columns using blind source separation algorithms[1], but this blind identification is not sufficient for our needs since we would like to resolve the powers of the individual cores and map them to the exact cores.

In the sequel, we propose to (1) take advantage of the particular physical characteristics of thermal transfer to solve the permutation ambiguity, and to (2) solve the scaling ambiguity by using the total power measurements.

IV. Proposed Methodology

First, we start by estimating the natural response matrix $\mathbf{A}$. This is conducted by forcing $\mathbf{p}(k) = 0$ after raising the temperatures of the cores using a workload. From equation (1) and $\mathbf{p}(k) = 0$, one has

$$\mathbf{t}(k) = \mathbf{A}\mathbf{t}(k-1) + \epsilon \tag{9}$$

An estimate of $\mathbf{A}$ is obtained by the least square minimization

$$|\mathbf{t}(k) - \mathbf{A}\mathbf{t}(k-1)|^2 \text{ under the constraint } \mathbf{A} \succeq 0 \tag{10}$$

where $\mathbf{A} \succeq 0$ denotes that the entries of $\mathbf{A}$ are non-negative.

[1]In this paper, we will be using the NMF [13] and fast ICA [10] algorithms.

Second, the thermal transfer matrix $\mathbf{R}$ is obtained up to a permutation and scaling factor of its columns by using the NMF (Non Negative Matrix Factorization) algorithm [13] initialized by the fast ICA algorithm [10]. The NMF algorithm is considered to cope with the positivity constraint of the Matrix $\mathbf{R}$ and the power profiles $\mathbf{p}(k)$. The outcome of blind identification algorithms is the estimation of the power profiles up to permutation and scaling.

Let us define by $\tilde{\mathbf{R}}$ and $\tilde{\mathbf{p}}(k)$ the estimates up to permutation and scaling of $\mathbf{R}$ and $\mathbf{p}(k)$. The contribution of the n-th power source to the thermal measurements is given by

$$\mathbf{t}_n(k) = \tilde{\mathbf{r}}_n \tilde{p}_n(k) \tag{11}$$

where $\tilde{\mathbf{r}}_n$ and $\tilde{p}_n(k)$ are the n-th column of matrix $\tilde{\mathbf{R}}$ and the n-th entry of the vector $\tilde{\mathbf{p}}(k)$, respectively.

Since the identification is blind, one has:

$$\tilde{\mathbf{r}}_n = \alpha_n \mathbf{r}_{\eta(n)} \tag{12}$$

$$\tilde{p}_n(k) = \frac{1}{\alpha_n} p_{\eta(n)}(k) \tag{13}$$

where $\mathbf{r}_{\eta(n)}$ and $p_{\eta(n)}$ are the column of thermal transfer matrix matrix $\mathbf{R}$ and the power profile $\mathbf{p}(k)$ for some permutation $\eta(n)$ of the core index n, respectively, and α_n denotes the scaling ambiguity factor. According to equations (12) and (13), Equation (11) reads:

$$\mathbf{t}_n(k) = \mathbf{r}_{\eta(n)} p_{\eta(n)}(k) \tag{14}$$

where the scaling ambiguity disappears, but we still have the permutation ambiguity. To solve this ambiguity, one resorts to the physical characteristics of the thermal transfer matrix. The latter has the characteristics that highest thermal coupling occurs within each core (i.e., self coupling) with smaller thermal coupling from the neighboring cores; i.e., the largest values of the thermal-transfer matrix should be at the diagonal. Hence, the correct core index is recovered by looking to the position of the maximum value of the entries of the reconstructed temperature vector $\mathbf{t}_n(k)$.

Once the permutation ambiguity is solved, the sorted estimates of the thermal transfer matrix and the power profile vector, denoted $\mathbf{R}'$ and $\mathbf{p}'(k)$, respectively, are equal to:

$$\mathbf{r}'_n = \alpha_n \mathbf{r}_n \tag{15}$$

$$p'_n(k) = \frac{1}{\alpha_n} p_n(k) \tag{16}$$

where $\mathbf{r}'_n$ and $p'_n(k)$ are the n-th column of matrix $\mathbf{R}'$ and the n-th entry of the vector $\mathbf{p}'(k)$, respectively.

From Equation (16), one can get the following expressions:

$$\alpha_n p'_n(k) = p_n(k) \tag{17}$$

$$\sum_{n=1}^{N} \alpha_n p'_n(k) = \sum_{n=1}^{N} p_n(k) \tag{18}$$

$$\alpha^T \mathbf{p}'(k) = c(k), \quad k = 1, .., K \tag{19}$$

where $\alpha = [\alpha_1 \cdots \alpha_N]^T$ and $c(k) = \sum_{n=1}^{N} p_n(k)$ denotes the measured total power. Equation (19) can be rewritten as:

$$\alpha^T \mathbf{P} = \mathbf{c}^T \tag{20}$$

where $\mathbf{P} = [\mathbf{p}'(1) \cdots \mathbf{p}'(K)]$ and $\mathbf{c} = [c(1) \cdots c(K)]^T$. Solution to Equation (20) and hence to the scaling ambiguity problem is then given by:

$$\alpha = \mathbf{P}^{\dagger} \mathbf{c} \tag{21}$$

where † denotes the pseudo inverse operator.

An estimate of the thermal transfer matrix $\mathbf{R}$ is obtained my sorting and re-scaling the columns of the corresponding estimates provided by the blind identification. Hence,

$$\mathbf{R} = \mathbf{R}' \mathbf{Diag}[\alpha]^{-1} \tag{22}$$

where $\mathbf{Diag}[\alpha]$ denotes a diagonal matrix whose diagonal is the entries of vector α. Then the forced response matrix $\mathbf{B}$ is estimated through equation (7): $\mathbf{B} = (\mathbf{I} - \mathbf{A})\mathbf{R}$.

During online tracking, the core power profiles at every instance of time are obtained by solving the following quadratic programming periodically:

$$||\mathbf{B}\mathbf{p}(k) - (\mathbf{t}(k) - \mathbf{A}\mathbf{t}(k-1))||_2, \quad \text{such that } \mathbf{p}(k) \succeq 0 \tag{23}$$

Procedure: Blind Identification of Power Profiles
Input: Temperatures $\mathbf{t}(k)$, Total Power $c(k)$
Output: Natural Response Matrix $\mathbf{A}$, Thermal Transfer Matrix $\mathbf{R}$, Forced Response Matrix $\mathbf{B}$, Power Profiles $\mathbf{p}(k)$

Off-line training:

1) Find the Natural Response Matrix $\mathbf{A}$ through the least square minimization: $|\mathbf{t}(k) - \mathbf{A}\mathbf{t}(k-1)|^2$ under the constraint $\mathbf{A} \succeq 0$
2) Find $\tilde{\mathbf{R}}$ the thermal transfer matrix and $\tilde{\mathbf{p}}(k)$ the power profiles up to permutation and scaling using the NMF [13] and the fast ICA [10] algorithms
3) Solve the Permutation Ambiguity:
   - Calculate the contribution of core n to the thermal measurements: $\mathbf{t}_n(k) = \tilde{\mathbf{r}}_n \tilde{p}_n(k)$
   - Recover the correct core index by finding the position of the maximum value of the entries of the vector $\mathbf{t}_n(k)$
   - Sort the columns of $\tilde{\mathbf{R}}$ and entries of $\tilde{\mathbf{p}(k)}$ to obtain: $\mathbf{R}'$ and $\mathbf{p}'(k)$
4) Solve the Scaling Ambiguity:
   - Form $\mathbf{P} = [\mathbf{p}'(1) \cdots \mathbf{p}'(K)]$ and $\mathbf{c} = [c(1) \cdots c(K)]^T$
   - Find the scaling: $\alpha = \mathbf{P}^{\dagger}\mathbf{c}$
5) Find the thermal transfer matrix $\mathbf{R}$: $\mathbf{R} = \mathbf{R}'\mathbf{Diag}[\alpha]^{-1}$
6) Find the forced response matrix $\mathbf{B}$: $\mathbf{B} = (\mathbf{I} - \mathbf{A})\mathbf{R}$

Runtime estimation:

7) Solve quadratic programming: $||\mathbf{B}\mathbf{p}(k) - (\mathbf{t}(k) - \mathbf{A}\mathbf{t}(k-1))||_2$, such that $\mathbf{p}(k) \succeq 0$
8) Return solution of the quadratic programming $\mathbf{p}(k)$

Fig. 1. Blind Power Identification Algorithm.

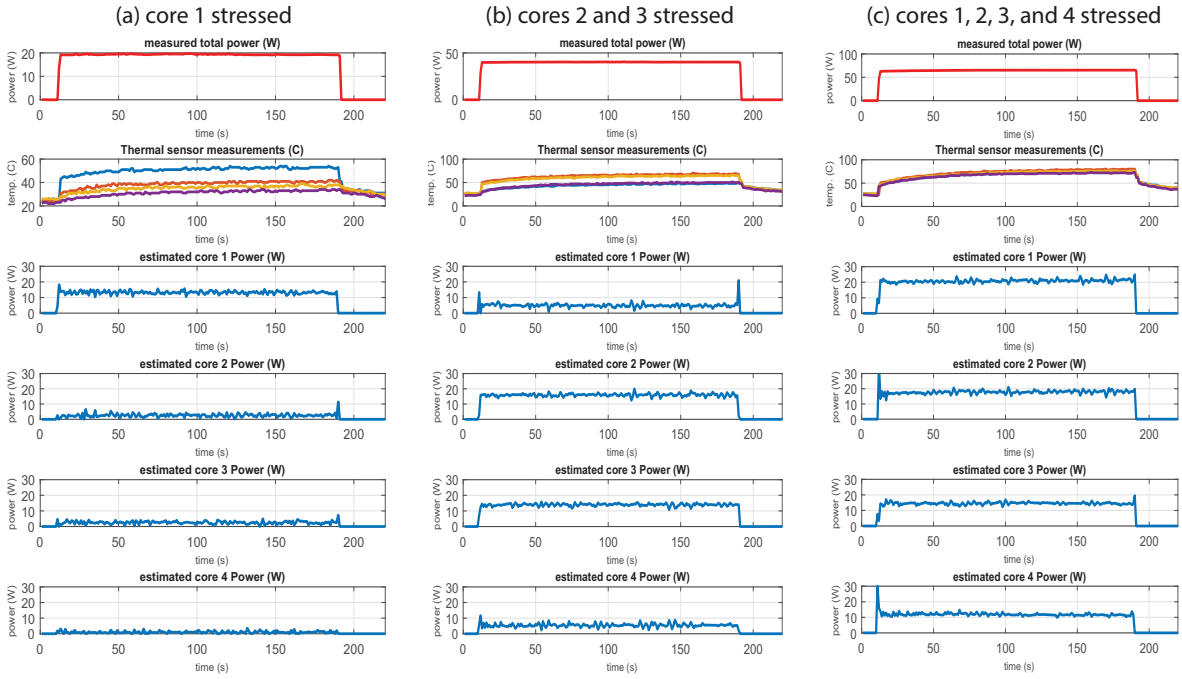

Fig. 2. Illustration of successful operation of BPI in which various cores are stressed and the power of the cores are blindly identified. In (a) one core is stressed (core 1), (b) two cores are stressed (cores 2 and 3), and (c) all cores are stressed.

The identification procedure is summarized in Figure 1. Note that all the steps, except for steps 7 and 8, are done once in the initial identification stage. Once $\mathbf{A}$ and $\mathbf{B}$ are identified, steps 7 and 8 are the only ones that need to be executed during runtime power estimation, which minimizes the overhead of our method during runtime to a great extent.

V. Experimental Results

We use a Linux-based system with an Intel Haswell processor Core i7-4790K (Devil Canyon) which features four cores, an integrated GPU, and a L3 cache of 8 MB. The RAPL interface enables us to read the total power consumption of all the cores. We use the `lmsensors` module v3.3.4 to read the thermal measurements of the four cores. The sampling rate of the RAPL power and thermal sensor measurements is 1 second. We use Intel's speed driver to control the frequencies and voltages of the cores. The driver adjusts the frequencies of the cores automatically depending on the load and the available thermal/power envelope to a maximum of 4.2 GHz. Thus, the frequency is variable during our experiments. While we have fixed the fan speed in our experiments, a variable fan speed can be incorporated in our technique by repeating our modeling approach under various fan speed settings, and then looking up the correct model during power tracking depending on the actual fan speed. Using a large collection of execution traces, the initial phase of our BPI algorithm is run once to blindly estimate the state-space model matrices. During runtime, our light-weight power estimation (Steps 7 and 8 in algorithm Figure 1) takes about than 4.75 ms per-sample to compute the per-core power estimates. Our source code is available at `github.com/scale-lab/BPI`.

1. Demonstration Using Controllable Stress Generator. We first demonstrate that our BPI technique produces correct results on the real system. We design a multi-threaded stress generation application that enable us to control the number of threads and the exact cores that are being stressed when the application is executed. For space considerations, we demonstrate three cases out of the possible 16 cases: (a) one core is stressed (core 1), (b) two cores are stressed (cores 2 and 3), and (c) all cores are stressed. The results are given in Figure 2, where we report the total power, the measurements from the thermal sensors, and per-core power estimates using our BPI technique for the three cases. We observe from the plots that our technique is able to break-down the power consumption and map it to the four cores correctly as known from the controlled scheduling. While some of the inactive cores are correctly estimated to consume a small amount of power, this power is mainly attributed to leakage power, since our technique identifies the total power (dynamic and leakage). Furthermore, we can see that the cores do not consume the same exact power when they are all active (case c). This can be attributed to leakage power which depends on the thermal profile and process variability [5].

We also implemented the regression-based approach reported in [3], which only works in steady state and assumes that active cores are fully busy. We found that it can lead to up 11.4% deviation in power estimation compared to the actual steady-state total power. This deviation results because the method does not account for the automatic changes in operational voltage-frequency and leakage power when more cores are activated.

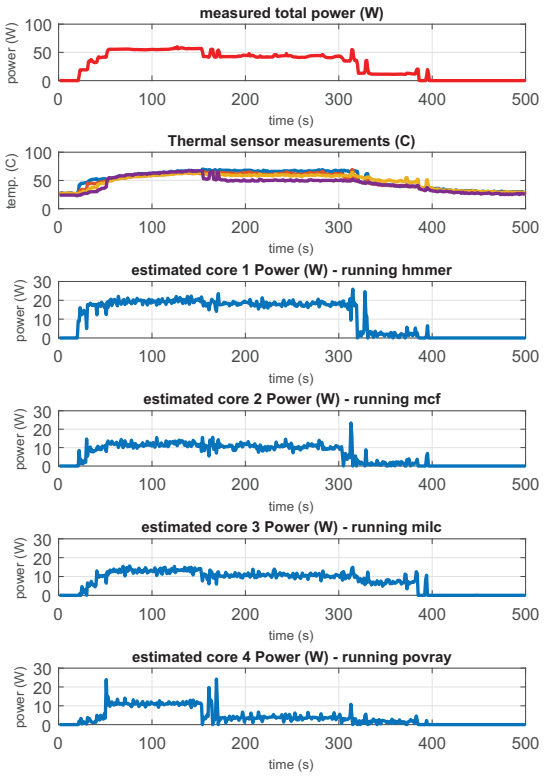

Fig. 3. Demonstration of BPI using a mix of SPEC CPU 2006 benchmarks. Four benchmarks are launched with one benchmark per core.

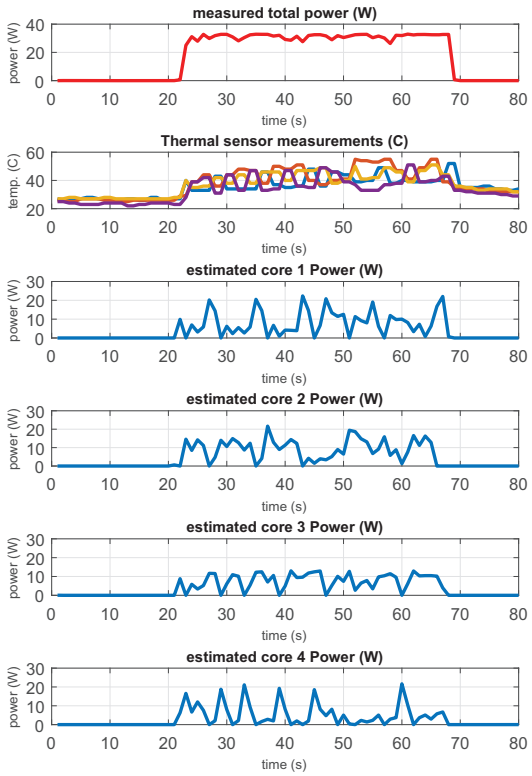

Fig. 4. Demonstration of BPI using `bodytrack` from PARSEC benchmark configured to run using two threads.

2. Demonstration Using Standard Benchmarks. We conduct two experiments using SPEC CPU 2006 and PARSEC benchmarks to demonstrate the ability to track the power consumption of general benchmarks. In the first experiment, we launch four benchmarks of the SPEC CPU 2006: `hmmer`, `mcf`, `milc`, and `povray` on cores 1, 2, 3 and 4 respectively. We wait about 10 seconds between launching two consecutive benchmarks. Figure 3 gives the per-core estimates from our BPI algorithm. The plot correctly shows that the estimated power of core 4 spikes at about 40 seconds when `povray` was launched on it, and that the core is almost inactive right after `povray` terminates at time 168 seconds. In a similar trend, the power estimates of every core are very low, as expected, right after the completion of the SPEC benchmark running on the core. Furthermore, Core 1 is displaying the highest power consumption among all cores, as it executes `hmmer`, which is the most CPU intensive benchmark.

In the second experiment, we use `bodytrack` from the PARSEC multi-threaded benchmarks. We limit it to a maximum of two threads, and use our BPI algorithm to estimate the per-core power estimates. The results are given in Figure 4. Interestingly, the plot shows activation of all cores; however, not all cores appear active simultaneously. This result perfectly matches `bodytrack` characteristics, which launches one thread per image to analyze 260 image frames. Since we limited `bodytrack` to two threads, the Linux scheduler automatically seeks to balance the launched threads among the cores, and as a result all the cores are used over the course of execution but no more than two cores are active at a time. Our BPI algorithm correctly tracks this behavior during runtime, where the power spikes correspond to the launching and termination of threads on the various cores.

3. Verification of BPI Accuracy Using Simulators. We analyze the accuracy of the per-core power estimates derived from our BPI algorithm, by comparing our per-core power estimates with the actual per-core power consumption. Given that modern processors lack such sensors, we resort to simulation for verification. We create a quad-core processor layout and simulate its thermal characteristics using HotSpot [11], where we execute the simulation with realistic power traces for the individual cores. We then collect the thermal traces from HotSpot together with the total power consumption and give them as inputs to our BPI algorithm. Our BPI algorithm identifies the state-space matrix models and the per-core power estimates. We then compare the per-core power estimates produced from BPI against the actual power consumptions of the individual cores that were used as inputs to HotSpot.

Figure 5 gives the results of our experiment using 32 different power traces applied over time. In particular, Figure 5.a gives the thermal simulation output from HotSpot for the four cores, while Figures 5.b-e give the per-core power traces given as inputs to HotSpot (dashed red line) and the per-core

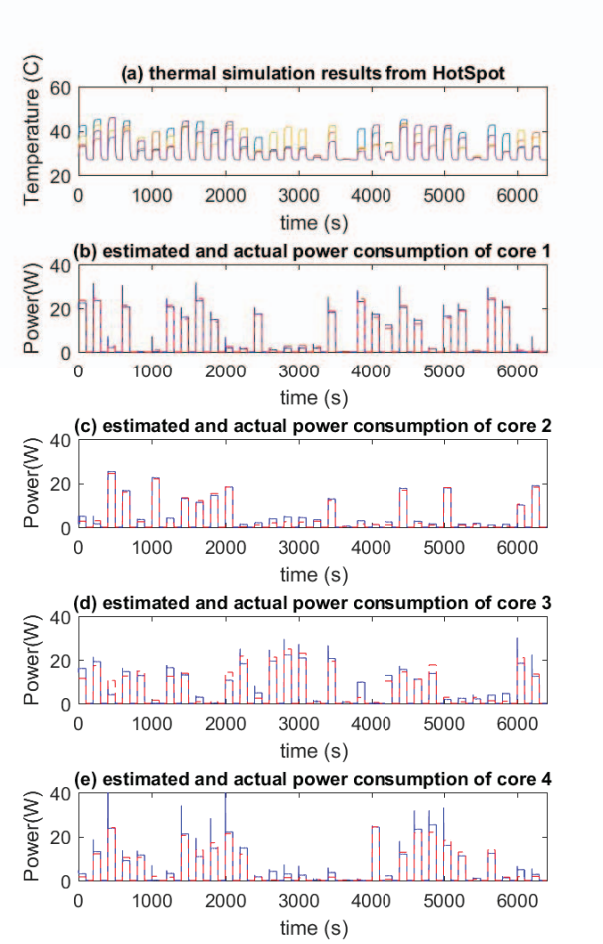

Fig. 5. Verification of BPI using HotSpot. Subfigure (a) gives the thermal measurements of the four cores using HotSpot, subfigures (b-e) give the per-core power estimates from BPI and the input power of each core to HotSpot. Dashed red lines give input power, while blue lines give estimated power.

num of cores cores	average absolute error (W)	average absolute error (%)
4	0.675 W	1.04%
9	0.719 W	1.09%
16	1.395 W	1.64%

TABLE I
SUMMARY OF BPI ACCURACY. THE AVERAGE ABSOLUTE ERROR IN PER-CORE POWER ESTIMATES ARE REPORTED AS A FUNCTION OF THE NUMBER OF CORES. HOTSPOT IS USED FOR VALIDATION OF PER-CORE ESTIMATES AGAINST THE ACTUAL POWER CONSUMPTION OF EACH CORE.

power estimates computed from BPI (solid blue lines) for the four cores. The results in Figure 5 demonstrate that BPI tracks the power accurately as the average absolute errors in the per-core power estimates are equal to 0.39 W, 0.46 W, 1.12 W, 0.72 W for core 1, 2, 3 and 4 respectively. Thus, the average per-core error is about 0.675 W. To understand the scalability of our algorithm as a function of the number of power sources; i.e., cores, we repeat our verification experiment for multi-core processors with 9-core and 16-core configurations. We summarize the average absolute error in the per-core power estimates in Table V. Our results show excellent scalability as the number of cores increases on the chip.

VI. CONCLUSIONS

We proposed a new technique for blind identification of power consumption of individual cores in multicore processors with no need for any *a priori* thermal-power models. Our BPI technique simultaneously identifies the state-space thermal model and the power consumption of cores from just the measurements of total power and thermal sensors during runtime. To overcome the challenges in general blind source separation techniques, we proposed methods that exploit the nature of thermal characteristics, and the total power measurements to construct the state-space model correctly with appropriate permutation and scaling factors. We have implemented our technique on a real multi-core processor and we used it to track the exact power consumption of its cores under different workloads. We also verified our technique's superior accuracy using a controlled simulation environment.

Acknowledgments: This work is partially supported by NSF grants #1438958, and by an Arab-American Frontiers Fellowship of the U.S. National Academy of Sciences, Engineering and Medicine.

REFERENCES

[1] A. Bartolini, M. Cacciari, A. Tilli and L. Benini, "A distributed and self-calibrating model-predictive controller for energy and thermal management of high-performance multicores," *IEEE DATE*, pp. 1-6, 2011.
[2] A. Belouchrani, K. Abed-Meraim, J.F. Cardoso and E. Moulines, "A blind source separation technique using second order statistics", IEEE Trans. on Signal Processing, vol. 45, no. 2, pp. 434-444, February 1997.
[3] F. Beneventi, A. Bartolini, and L. Benini, "Static thermal model learning for high-performance multicore servers", in *ICCCN*, pp. 1-6, 2011.
[4] F. Beneventi, A. Bartolini, A. Tilli, and L. Benini, "An Effective Gray-Box Identification Procedure for Multicore Thermal Modeling", in *IEEE Transactions on Computers*, Vol. 63(5), 2014.
[5] S. Borkar, T. Karik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter variations and impact on circuits and microarchitecture", in ACM/IEEE DAC, pp. 338-342, 2003.
[6] R. Cochran and S. Reda, "Consistent Runtime Thermal Prediction and Control Through Workload Phase Detection", *DAC*, pp. 62-67, 2010.
[7] R. Cochran, A. N. Nowroz and S. Reda, "Post-Silicon Power Characterization Using Thermal Infrared Emissions," *ISLPED*, pp. 331-336, 2010.
[8] A. Coskun, T. Rosing and K. C. Gross, "Utilizing Predictors for Efficient Thermal Management in Multiprocessor SoCs", *IEEE Transactions on CAD of Integrated Circuits and Systems*, Vol 28(10), pp. 1503-1516, 2009.
[9] H. David *et al.*, "RAPL: Memory Power Estimation and Capping", in *ISLPED*, pp. 189-194, 2010.
[10] A. Hyvarinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis", IEEE Trans. on Neural Networks 10(3):626-634, 1999.
[11] W. Huang *et al.* "HotSpot: a compact thermal modeling methodology for early-stage VLSI design", IEEE Transactions on VLSI Systems, vol 14(5), pp. 501 - 513, 2006.
[12] C. Isci, G. Contreras and M. Martonosi, "Live, Runtime Phase Monitoring and Prediction on Real Systems with Application to Dynamic Power Management", in *ISCA*, pp. 369-370, 2006.
[13] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization", *Nature*, 401 (1999), pp. 788-791
[14] D. Li, S. X.-D. Tan, E. Pacheco and M. Tirmula, "Parameterized Architecture-level Dynamic Thermal Models for Multicore Microprocessors", ACM TODAES., Vol 15(2), pp. 16:1-16:22, 2010.
[15] S. Sharifi and C.-C. Liu and T. Rosing, "Accurate Temperature Estimation for Efficient Thermal Management", *ISQED*, pp. 137 – 142, 2008.
[16] Y. Wang, K. Ma, and X. Wang, "Temperature-Constrained Power Control for Chip Multiprocessors with Online Model Estimation", *ISCA*, pp. pp. 314-324, 2009.
[17] M. Yuffe *et al.*, "A Fully Integrated Multi-CPU, Processor Graphics, and Memory Controller 32-nm Processor", IEEE JSSC, Vol 47(1), pp. 194-205, 2012.