

# Runtime Power Estimator Calibration for High-Performance Microprocessors

Hai Wang\*, Sheldon X.-D. Tan\*, Xue-Xin Liu\*, and Ashish Gupta†

\*Department of Electrical Engineering, University of California, Riverside, CA 92521

†Intel Corporation, Chandler, AZ 85226

**Abstract**—Accurate runtime power estimation is important for on-line thermal/power regulation on today’s high performance processors. In this paper, we introduce a power calibration approach with the assistance of on-chip physical thermal sensors. It is based on a new error compensation method which corrects the errors of power estimations using the feedback from physical thermal sensors. To deal with the problem of limited number of physical thermal sensors, we propose a statistical power correlation extraction method to estimate powers for places without thermal sensors. Experimental results on standard SPEC benchmarks show the new method successfully calibrates the power estimator with very low overhead introduced.

## I. INTRODUCTION

Chip power performance is critical for today’s high-performance microprocessors as the transistor density has been increasing exponentially. It is directly related to the microprocessor’s energy efficiency, the chip’s thermal reliability and life expectancy. As a result, accurate estimation of power at runtime is crucial for the energy efficiency optimization, dynamic thermal/power management [1], [2], [3], [4] and chip reliability analysis [5], [6].

The coarse runtime power estimation provides total power consumption at the die level and can be used to assist the global power/thermal managements such as fan speed control and dynamic voltage and frequency scaling (DVFS). However, today’s multi-core computer architecture enables the ability to perform more efficient fine-grained management such as task scheduling and computing migration, for which accurate functional-block-level power estimation is required [1], [2]. Although one is able to monitor the total power consumption of the die easily, measuring the runtime power at functional-block-level is extremely difficult [7]. As a result, there are many researches conducted in this area, and most of the proposed methods are performance counter based [8], [9], [10]. The functional-block-level power estimators count the execution numbers of various performance actions for each functional block in a time frame and calculate the power by multiplying the execution numbers with its corresponding performance parameters. However, the power estimators cannot be very accurate due to several reasons. First, not all the executions are counted in the performance counting process



Fig. 1. A simple schematic diagram of the power calibration process.

due to the complex behaviors of the microprocessor at runtime. Second, the performance parameters are not static in general because of the temperature variations and the aging of the chip.

An alliterative way to get more accurate power estimation at runtime is to exploit the thermal-power relation of the chip and utilize the on-chip thermal sensors to calibrate the power estimator. The simple schematic diagram of the proposed power calibration process is shown in Fig. 1. A specially designed compact thermal model takes the power estimation from the power calibrator and calculates the full-chip temperatures with low overhead. The temperatures from the compact thermal model, the physical thermal sensor readings and the estimated power are fed into the power calibrator together, with the calibrated functional-block-level runtime power as the output.

In this paper, we address the accuracy problem of the runtime functional-block-level power estimation by introducing a power calibration method. The main contributions of this paper are:

- 1) First, we show how on-chip physical sensors can be used to compensate the power estimation error by exploiting the thermal-power relationship of the chip.
- 2) Second, we show how to fully utilize the correlations among the power errors of different functional blocks and reach an accurate calibration when the number of thermal sensors is limited.
- 3) Third, we propose a statistical correlation extraction scheme which characterizes the functional block correlations in a systematic way.

The rest of this paper is organized as follows: In Section II, the basic thermal-power relationship of the chip and the power calibration problem are presented. In Section III, we demonstrate the new runtime power calibration method using physical on-chip thermal sensors. Experimental results are

This research was supported in part by NSF grants under No. NSF-0902885, and Semiconductor Research Corporation grant under No. SRC 2009-TJ-1991.

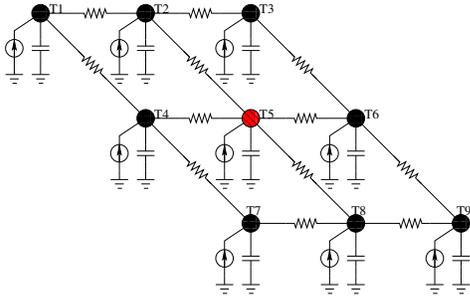


Fig. 2. A nine-grid equivalent thermal circuit. Each grid has a thermal node  $T_i$  denoted as a solid circle (black or red dashed), a thermal capacitor and a current source representing the power dissipation at the grid. There is also a thermal resistor between each pair of the adjacent thermal nodes. A thermal sensor, denoted as the red dashed circle ( $T_5$ ), is placed at the center grid.

reported in Section IV and Section V concludes this paper.

## II. BACKGROUND

### A. Thermal-power relation of the chip

In order to utilize on-chip physical sensors in the power calibration process, the thermal-power relation should be analyzed first.

The heat differential equation of the chip can be spatially discretized using finite difference method in the three dimensional space to generate an equivalent thermal circuit [11]. A two dimensional nine-grid equivalent thermal circuit example is shown in Fig. 2. As shown in the figure, each grid has a thermal node  $T_i$ , a thermal capacitor and a current source representing the power dissipation at the grid. There is also a thermal resistor between the adjacent thermal nodes. One thermal sensor, denoted as the red dashed circle, is placed at the center grid in this example.

Mathematically, if there are  $n$  discretized grids with specific boundary conditions, the equivalent thermal circuit can be modeled using an ordinary differential equation [11]

$$C \frac{dT(t)}{dt} + GT(t) = BU(t) \quad (1)$$

where  $T(t) \in \mathbb{R}^n$  is the temperature vector containing the temperatures of the  $n$  thermal nodes,  $C \in \mathbb{R}^{n \times n}$  is the thermal capacitance matrix,  $G \in \mathbb{R}^{n \times n}$  is the thermal conductance matrix,  $B \in \mathbb{R}^{n \times p}$  is the position matrix of the input where  $B_{i,j}$  denotes the portion of the  $j$ th functional block power injects into the  $i$ th thermal node and  $U(t) \in \mathbb{R}^p$  contains the power dissipations of the  $p$  functional blocks. The right hand side of (1) is also written as

$$J(t) = BU(t) \quad (2)$$

where  $J(t) \in \mathbb{R}^n$  represents the power dissipations of  $n$  grids.

### B. Power estimator calibration problem

In the thermal model introduced in the previous subsection, the input  $J(t)$  (or  $U(t)$ ) is accurate and the resulting temperature  $T(t)$  is accurate. Assume the power estimation from a

power estimator is  $\bar{J}(t)$  ( $\bar{U}(t)$ ), the system equation with the estimated power is

$$C \frac{d\bar{T}(t)}{dt} + G\bar{T}(t) = \bar{J}(t) \quad (3)$$

where  $\bar{T}$  is temperature estimation using the power estimation  $\bar{J}(t)$ .  $\bar{T}$  is not accurate, but can be used for the power calibration process.

Our goal is to compensate the power estimation error and get an accurate power as close to  $J(t)$  as possible. We will show in the following section that how the compensation is performed with the help of thermal sensors.

## III. RUNTIME POWER ESTIMATOR CALIBRATION METHOD

In this section, we present the runtime power estimator calibration method. First, a power error compensation method is presented with the assumption of infinite number of thermal sensors. Then, a statistical correlation extraction method is proposed to make the power error compensation applicable with limited number of thermal sensors.

### A. Power error compensation process

As briefly introduced in Section I, error is inevitable for the runtime power estimators. In order to obtain the power compensation term using thermal sensor information, we have to first simulate the thermal system numerically using the inaccurate power estimation as input. The simulation is performed by discretizing (3) in time domain. We use Backward Euler (BE) here for illustration. By choosing an appropriate time step  $h$ , BE discretizes (3) in time domain as

$$\left(\frac{C}{h} + G\right)\bar{T}(t+h) = \frac{C}{h}\bar{T}(t) + \bar{J}(t+h) \quad (4)$$

Through inverting  $\left(\frac{C}{h} + G\right)$  to the right hand side, (4) is also written as

$$\bar{T}(t+h) = \left(\frac{C}{h} + G\right)^{-1} \left(\frac{C}{h}\bar{T}(t) + \bar{J}(t+h)\right) \quad (5)$$

Given the initial value  $\bar{T}(0)$  and the input  $\bar{J}(t)$  for all time points, the subsequent temperature  $\bar{T}(t)$  can be calculated iteratively using (5).

However, the temperature  $\bar{T}(t)$  calculated from (5) is inaccurate due to the inaccurate input  $\bar{J}$ . Assume the actual power input is  $J = \bar{J} + \delta J$ . The real system response  $T$  can be calculated from

$$\left(\frac{C}{h} + G\right)T(t+h) = \frac{C}{h}T(t) + J(t+h) \quad (6)$$

1) *Power error compensation with sufficient thermal sensors:* We would like to compensate the power estimation error with the feedback from thermal sensors.

In the ideal case, assume there are thermal sensors everywhere on the chip, that is, we have the accurate temperature information  $T(t)$  already. We define the temperature estimation error  $\delta T$ , power estimation error  $\delta J$  as

$$\delta T(t) := T(t) - \bar{T}(t) \quad (7)$$

$$\delta J(t) := J(t) - \bar{J}(t) \quad (8)$$

Then subtract (4) from (6) and neglect the small second order term, we have

$$\left(\frac{C}{h} + G\right)\delta T(t+h) = \frac{C}{h}\delta T(t) + \delta J(t+h) \quad (9)$$

Because of the low-pass filter property of thermal system [2], the temperature estimation error over two successive time steps does not change too much, that is  $\delta T(t+h) \approx \delta T(t)$ . Therefore, (9) becomes

$$\left(\frac{C}{h} + G\right)\delta T(t) \approx \frac{C}{h}\delta T(t) + \delta J(t+h) \quad (10)$$

We define the *error compensation term*, determined at time  $t+h$ , as

$$\epsilon := \delta J(t+h) \quad (11)$$

and from (10), the error compensation term  $\epsilon$  can be approximately solved as

$$\epsilon \approx G\delta T(t) \quad (12)$$

We do not express  $\epsilon$  as a variable of  $t$  since it will not be calculated repeatedly at every time point.

After we obtain the error compensation term, the power estimations of all the future time points are updated as

$$J(t+ih) = J(t+ih) + \epsilon \quad (13)$$

where  $i = 1, 2, \dots$

Note the error compensation term  $\epsilon$  is accurate as long as the power estimation error statistics do not change too much. In this case, one compensation/calibration is enough for the whole estimation time. If the condition is not satisfied, we can perform the error compensation process (12) and (13) periodically or at the time when the temperature errors at the thermal sensors exceed a threshold.

2) *Power error compensation with limited number of thermal sensors*: We have shown we are able to fully compensate the power estimation error to generate accurate power estimation in the ideal case with sufficient number of thermal sensors. However, we cannot put thermal sensors all over the chip in reality. The number of sensors is always limited and as a result, it is impossible to obtain all the elements of  $\delta T(t)$  in (12). In this subsection, we show how to exploit the power estimator and limited thermal sensor information and approximately recover the full-chip temperature.

Assume there are  $n_s$  thermal sensors placed on chip. For convenience, we first perform matrix permutation on (1) to group the thermal nodes with thermal sensors together as

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \begin{bmatrix} \frac{dT_s(t)}{dt} \\ \frac{dT_u(t)}{dt} \end{bmatrix} + \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} T_s(t) \\ T_u(t) \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U(t) \quad (14)$$

and

$$\begin{bmatrix} J_s(t) \\ J_u(t) \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} U(t) \quad (15)$$

where  $T_s(t) \in \mathbb{R}^{n_s}$  represents the temperatures at the nodes where thermal sensors are placed and  $T_u(t) \in \mathbb{R}^{n-n_s}$  represents the temperatures at the nodes without thermal sensors.

Accordingly, (12) becomes

$$\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \delta T_s(t) \\ \delta T_u(t) \end{bmatrix} = \begin{bmatrix} \epsilon_s \\ \epsilon_u \end{bmatrix} \quad (16)$$

We know the value of  $\delta T_s$  since thermal sensors are placed at these nodes. However,  $\delta T_u$  is unknown due to the absence of thermal sensors. Since there are  $2n-n_s$  unknowns in (16) with  $n$  equations, (16) is unsolvable (in the normal sense) unless the number of unknowns is reduced. Fortunately, we are able to reduce the number of unknowns by taking advantage of the power correlation among different functional blocks in a chip and introduce a *correlation matrix*  $D \in \mathbb{R}^{(n-n_s) \times n_s}$ . Then, we can represent  $\epsilon_u$  in terms of  $\epsilon_s$  as

$$\epsilon_u = D\epsilon_s \quad (17)$$

The details of forming the  $D$  matrix in a systematic way are presented in Section III-B.

After the introduction of the correlation matrix  $D$ , the number of unknowns has been reduced to  $n$ . Combined with (17), (16) is rearranged as

$$\begin{bmatrix} G_{12} & -I_{n_s \times n_s} \\ G_{22} & -D \end{bmatrix} \begin{bmatrix} \delta T_u(t) \\ \epsilon_s \end{bmatrix} = \begin{bmatrix} -G_{11}\delta T_s(t) \\ -G_{21}\delta T_s(t) \end{bmatrix} \quad (18)$$

where  $I_{n_s \times n_s}$  is an identity matrix with dimension  $n_s$ . After  $\epsilon_s$  is solved from (18) and  $\epsilon_u$  is obtained from (17), the error compensation is performed with the permuted form of (13).

## B. Statistical correlation extraction

In this subsection, we provide a systematic method to extract the error compensation correlation and form the  $D$  matrix.

Our idea is based on the observation that many functional blocks in a chip are highly correlated in their power consumptions. For instance, when a integer register file is busy, most likely the integer ALU and nearby cache memory will also be busy. As a result, if we properly place the thermal sensors so that more correlated functional blocks are clustered around those thermal sensors, we should be able to have a good guess of the compensation errors around the thermal sensors. Specifically, based on the placement of the  $n_s$  thermal sensors, the chip is divided into  $n_s$  blocks by combining the correlated functional blocks around each thermal sensor. We call this kind of block as *sensor block*. The compensation errors of different nodes inside the same sensor block are correlated and the correlation can be characterized, mainly because the power consumptions of these functional blocks inside the same sensor block rely strongly on a small number of common performance parameters [10] such that the power estimation errors are dependent on each other statistically. For example, in (17), each column of  $D$  shows the correlation of the compensation errors within a specific sensor block.

Please note that instead of finding the error relation for each thermal node, it is only necessary to find the correlation among functional blocks since the powers of the nodes inside each functional block are extremely correlated and are usually considered to be the same or follow a static distribution. As

a result, we only need to find the relation of the total power error

$$\delta U_u = D_p \delta U_s \quad (19)$$

and the fine-grind power error relation (17) can be easily calculated.

There are three steps in the statistical correlation extraction process. The first step is to collect sample data, both from measurement and power estimator simulation. The second step is to group the functional blocks into sensor blocks according to the results of a correlation test. The final step is to find the exact formulation of the correlated power errors of the functional blocks in each sensor block using simple linear regression method.

Assume there are  $b$  benchmarks with steady power configurations. First, we run the benchmarks using the power estimator and record the power results

$$\hat{U} = [\hat{U}^1, \hat{U}^2, \dots, \hat{U}^b] \quad (20)$$

where, for example, the  $i$ th sample

$$\hat{U}^i = [\hat{u}_1^i, \hat{u}_2^i, \dots, \hat{u}_p^i]^T \quad (21)$$

since there are  $p$  functional blocks. Next, run the benchmarks on the test chip until the temperatures reach steady state, measure the steady state temperatures as  $T$ . The real power of the chip is reversely calculated as

$$U = [U^1, U^2, \dots, U^b] \quad (22)$$

using the measured temperatures. Note that all these steps should be performed off-line, such that the error can be better controlled and no overhead is introduced at runtime. Please see [12], [13] for details of the reverse power calculation. The errors of the functional block powers are obtained as

$$\delta U = U - \hat{U} \quad (23)$$

Also assume the functional blocks with thermal sensors are permuted to the first few blocks, such that we can also write

$$\delta U = \begin{bmatrix} \delta U_s \\ \delta U_u \end{bmatrix} \quad (24)$$

where the  $i$ th sample of  $\delta U_s$  is

$$\delta U_s^i = [\delta \hat{u}_1^i, \delta \hat{u}_2^i, \dots, \delta \hat{u}_{n_s}^i]^T \quad (25)$$

and the  $i$ th sample of  $\delta U_u$  is

$$\delta U_u^i = [\delta \hat{u}_{n_s+1}^i, \delta \hat{u}_{n_s+2}^i, \dots, \delta \hat{u}_p^i]^T \quad (26)$$

remember that  $n_s$  is the number of thermal sensors.

The next step is to determine the sensor blocks using a correlation test, such that functional blocks with high power error correlations can be identified and put into one sensor block. The correlations are tested first using the data samples  $\delta U$  through forming the correlation matrix (27) shown on top of the next page, where  $\mu_i$  is the expected value of  $\delta u_i$ . (27) can be also divided into blocks like

$$corr_{\delta u} = \begin{bmatrix} E_{ss} & E_{us}^T \\ E_{us} & E_{uu} \end{bmatrix} \quad (28)$$

By definition, correlation matrix is a symmetric matrix containing the correlation values of each random variable pair. The correlation value is a number between  $-1$  and  $1$  which reveals the dependence of a random variable pair, where  $1$  and  $-1$  indicate the two random variables are fully dependent and  $0$  means totally independence. By investigating  $E_{us}$  which contains the correlation values of all the  $\delta U_s$  and  $\delta U_u$  pairs, we can easily determine which sensor block does the  $i$ th functional block without thermal sensors belongs to: for the  $i$ th row in  $E_{us}$ , simply take the column number of the element with the largest absolute value as the sensor block number.

For the final step, we use the linear regression method to find the relations among the functional blocks within each sensor block. Assume  $i$ th functional block is associated with the  $j$ th functional block (which has thermal sensor placed), the relation

$$\delta u_j = a_j \delta u_i \quad (29)$$

is found using the sample data information  $[\delta u_i^1, \delta u_i^2, \dots, \delta u_i^b]$  and  $[\delta u_j^1, \delta u_j^2, \dots, \delta u_j^b]$ . With (29) for each functional block without thermal sensors, i.e.  $j = 1, 2, \dots, n - n_s$ ,  $D_p$  in (19) is populated with  $a_j$  and the correlation matrix  $D$  in (17) is derived subsequently.

### C. Compact thermal modeling and practical implementation considerations

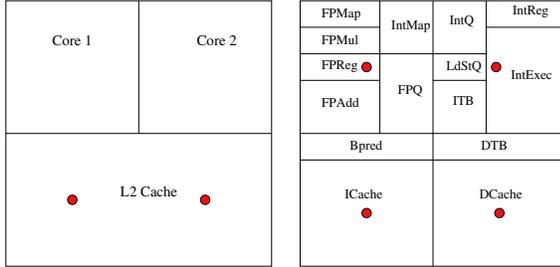
Thermal model is used in our power calibrator to connect the power and thermal. However, at the same time, it introduces overhead and degrades the system performance. The overhead can be significant especially when the full-chip thermal model is used. Model order reduction (MOR) technique, which reduces the size of large dynamic system models, can be used to generate a compact thermal model and reduce the runtime overhead. In this work, Krylov subspace based approach is used with structure preservation [14] to generate the compact thermal model as we need to preserve the structure of (14). Interested readers are referred to [15] for a comprehensive MOR introduction.

The practical implementation of the new power calibration scheme needs to be considered carefully to avoid overhead as much as possible. The full thermal model generation, statistical correlation extraction, model order reduction process, and the pre-factorization of the compact thermal system matrices are performed off-line. The on-line computation should only contain the temperature calculation with the pre-factorized compact thermal system matrices and the power error compensation.

## IV. EXPERIMENTAL RESULTS

The experiments are conducted using Matlab on a Linux server with Intel 3.0GHz quad-core CPU and 16GB memory. In order to validate the new power estimator calibration method, we build a dual-core processor with a shared L2 cache which is shown in Fig. 3 (a). The size of the processor is  $10mm \times 10mm \times 0.7mm$ . The core architecture shown in Fig. 3 (b) is similar to the Alpha ev6 processor. There are 10 thermal sensors placed on chip in total, 4 for each core and 2

$$corr_{\delta u} = \begin{bmatrix} \frac{E[(\delta u_1 - \mu_1)(\delta u_1 - \mu_1)]}{\sigma_{\delta u_1}^2} & \frac{E[(\delta u_1 - \mu_1)(\delta u_2 - \mu_2)]}{\sigma_{\delta u_1} \sigma_{\delta u_2}} & \dots & \frac{E[(\delta u_1 - \mu_1)(\delta u_p - \mu_p)]}{\sigma_{\delta u_1} \sigma_{\delta u_p}} \\ \frac{E[(\delta u_2 - \mu_2)(\delta u_1 - \mu_1)]}{\sigma_{\delta u_2} \sigma_{\delta u_1}} & \frac{E[(\delta u_2 - \mu_2)(\delta u_2 - \mu_2)]}{\sigma_{\delta u_2}^2} & \dots & \frac{E[(\delta u_2 - \mu_2)(\delta u_p - \mu_p)]}{\sigma_{\delta u_2} \sigma_{\delta u_p}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{E[(\delta u_p - \mu_p)(\delta u_1 - \mu_1)]}{\sigma_{\delta u_p} \sigma_{\delta u_1}} & \frac{E[(\delta u_p - \mu_p)(\delta u_2 - \mu_2)]}{\sigma_{\delta u_p} \sigma_{\delta u_2}} & \dots & \frac{E[(\delta u_p - \mu_p)(\delta u_p - \mu_p)]}{\sigma_{\delta u_p}^2} \end{bmatrix} \quad (27)$$



(a) The dual-core microprocessor architecture.

(b) The architecture for each core composed of functional blocks.

Fig. 3. The dual-core microprocessor architecture, with two cores and one shared L2 cache. 10 thermal sensors (red solid circle) are placed on chip, 2 on the L2 cache and 4 on each core.

TABLE I  
SENSOR BLOCKS DETERMINED BY THE STATISTICAL EXTRACTION.

Sensor block #	Functional blocks in the sensor block
1	L2 Cache Left
2	Core 1: ICache, Bpred
3	Core 1: DCache, DTB
4	Core 1: FPAdd, FPReg, FPMul, FPMap, FPQ
5	Core 1: IntMap, IntQ, LdStQ, ITB, IntExec
6	L2 Cache Right
7	Core 2: ICache, Bpred
8	Core 2: DCache, DTB
9	Core 2: FPAdd, FPReg, FPMul, FPMap, FPQ
10	Core 2: IntMap, IntQ, LdStQ, ITB, IntExec

for the L2 cache as shown in Fig. 3. The power information is obtained using the power estimator Wattch [16] by running the standard SPEC benchmarks [17]. One core of the dual-core processor is assumed to be active and the other one is assumed to be idle, they can be switched when the temperature on one core is too high. The power estimations given by the power estimator is modeled with up to 20% mean value error with the correlations similar to the one reported in [18]. The original order of the thermal model is 3200 and the reduced model, which is used in our power calibrator, has the order of 106. The simulation time step  $h$  is chosen to be 0.1s to balance the speed and accuracy.

The sensor blocks determined by the statistical correlation extraction are shown in Table I. The accuracy comparison of the power density map snapshot of bzip2 benchmark is given in Fig. 4. The real power density map is shown in Fig. 4 (a), the estimated power density map which has significant error is shown in Fig. 4 (b), and the power density map

TABLE II  
RUNTIME AND ACCURACY COMPARISON OF THE POWER CALIBRATION METHOD ON SPEC BENCHMARKS.

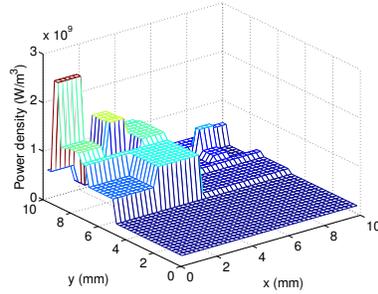
Benchmark	Estimation err	Calibration				
		org time	org err	red time	X	red err
bzip2	14%	0.04	4.1%	0.0011	36X	4.2%
gzip	11%	0.12	4.3%	0.0016	75X	4.3%
mcf	17%	0.06	6.3%	0.0013	46X	6.1%
mgrid	12%	0.04	1.5%	0.0014	29X	2.3%
swim	13%	0.05	2.3%	0.0011	45X	2.4%
galgel	13%	0.09	1.9%	0.0013	69X	2.0%

after the calibration process is demonstrated in Fig. 4 (c). It is clear from the figures that the power calibration process successfully compensated the power estimation errors and generated a much more accurate power density map compared to the directly estimated one. The result with the compact thermal model can be found in Fig. 4 (d). It is almost the same as Fig. 4 which reveals the high accuracy of the compact thermal model.

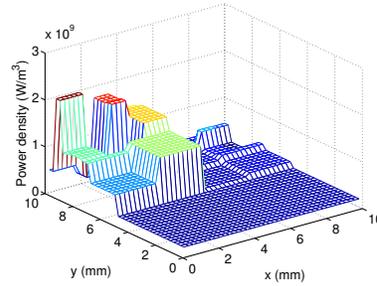
The detailed results on other benchmarks are presented in Table II, where *Estimation* means the inaccurate power estimation, *org* is the calibration with the original thermal model and *red* suggests the compact thermal model is used, *X* shows the speedup of the compact model over the original model. To be fair, all the times are measured as the time spent to calibrate 1 second transient power map, with the unit *s*. Even with the large average power estimation error around 15%, the new power calibration method reduces the average error to around 4%. The overhead of the power estimator is also low, especially with the compact thermal model. Only about 0.0015 seconds are spent to calibrate 1 second transient power map.

## V. CONCLUSION

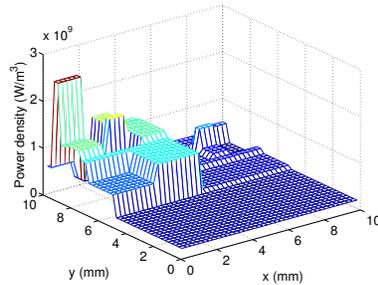
In this paper, we have proposed a new runtime power estimator calibration method for high-performance microprocessors with the assistance of on-chip physical thermal sensors. It is based on a new error compensation method which corrects the errors of power estimations using the feedback from thermal sensors. We also proposed a statistical correlation extraction method to fully utilize the information from limited number of thermal sensors. Experimental results on standard SPEC benchmarks demonstrate the new method successfully calibrates the power estimator with very low overhead introduced.



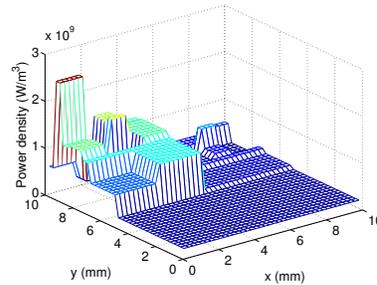
(a) The real power density map of the dual-core processor.



(b) The estimated power density map of the dual-core processor.



(c) The power density map of the dual-core processor after the calibration process with the original thermal model.



(d) The power density map of the dual-core processor after the calibration process with the compact thermal model.

Fig. 4. Comparison of the power density maps of the dual-core processor before and after the power calibration process.

## REFERENCES

- [1] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proceedings of the 33rd annual international symposium on Computer Architecture, ISCA '06*, (Washington, DC, USA), pp. 78–88, IEEE Computer Society, 2006.
- [2] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, pp. 2–13, 2003.
- [3] F. Zanini, D. Atienza, L. Benini, and G. De Micheli, "Multicore thermal management with model predictive control," in *Proc. 19th European Conference on Circuit Theory and Design, (Piscataway, NJ, USA)*, pp. 90–95, IEEE Press, 2009.
- [4] Y. Wang, K. Ma, and X. Wang, "Temperature-constrained power control for chip multiprocessors with online model estimation," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, pp. 314–324, 2009.
- [5] K. Kang, S. P. Park, K. Roy, and M. A. Alam, "Estimation of statistical variation in temporal NBTI degradation and its impact on lifetime circuit performance," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 730–734, 2007.
- [6] Y. Lu, L. Shang, H. Zhou, H. Zhu, F. Yang, and X. Zeng, "Statistical reliability analysis under process variation and aging effects," in *Proc. Design Automation Conf. (DAC)*, pp. 514–519, 2009.
- [7] G. Liao, X. Zhu, S. Larsen, L. N. Bhuyan, and R. Huggahalli, "Understanding power efficiency of TCP/IP packet processing over 10GbE," in *18th Symposium on High-Performance Interconnects*, pp. 32–39, 2010.
- [8] C. Isci and M. Martonosi, "Runtime power monitoring in high-end processors: Methodology and empirical data," in *Proceedings of MICRO*, 2003.
- [9] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "A systematic method for functional unit power estimation in microprocessors," in *Proc. Design Automation Conf. (DAC)*, pp. 554–557, June 2006.
- [10] M. D. Powell, A. Biswas, J. S. Emer, S. S. Mukherjee, B. R. Sheikh, and S. Yardi, "CAMP: A technique to estimate per-structure power at run-time using a few simple parameters," in *Proc. IEEE Int. Symp. on High-Performance Computer Architecture (HPCA)*, pp. 289–300, 2009.
- [11] Y.-K. Cheng, C.-H. Tsai, C.-C. Teng, and S.-M. Kang, *Electrothermal Analysis of VLSI Systems*. Kluwer Academic Publishers, 2000.
- [12] A. Nowroz, G. Woods, and S. Reda, "Improved post-silicon power modeling using AC lock-in techniques," in *Proc. Design Automation Conf. (DAC)*, 2011.
- [13] R. Cochran, A. Nowroz, and S. Reda, "Post-silicon power characterization using thermal infrared emissions," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pp. 331–336, 2010.
- [14] R. W. Freund, "SPRIM: structure-preserving reduced-order interconnect macromodeling," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 80–87, 2004.
- [15] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems*. The Society for Industrial and Applied Mathematics (SIAM), 2005.
- [16] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, pp. 83–94, 2000.
- [17] J. L. Henning, "SPEC CPU 2000: Measuring CPU performance in the new millennium," *IEEE computer*, vol. 1, pp. 28–35, July 2000.
- [18] Y. Zhang, A. Srivastava, and M. Zahran, "Chip level thermal profile estimation using on-chip temperature sensors," in *Proc. IEEE Int. Conf. on Computer Design (ICCD)*, pp. 432–437, 2008.