Run-time Probabilistic Detection of Miscalibrated Thermal Sensors in Many-core Systems

Jia Zhao, Shiting (Justin) Lu, Wayne Burleson and Russell Tessier
Department of Electrical and Computer Engineering
University of Massachusetts, Amherst, MA, USA 01003-9284

***Abstract*—Many-core architectures use large numbers of small temperature sensors to detect thermal gradients and guide thermal management schemes. In this paper a technique to identify thermal sensors which are operating outside a required accuracy is described. Unlike previous on-chip temperature estimation approaches, our algorithms are optimized to run on-line while thermal management decisions are being made. The accuracy of a sensor is determined by comparing its readings to expected values from a probability distribution function determined from surrounding sensors. Experiments show that a sensor operating outside a desired accuracy can be identified with a detection rate of over 90% and an average false alarm rate of $< 6\%$, with a confidence level of 90%. The run time of our method is shown to be around 3x lower than a recently-published temperature estimation method, enhancing its suitability for run-time implementation.**

I. Introduction

On-chip sensors are widely used in many-core processors to closely monitor system temperature, performance, and supply power fluctuation, among other environmental conditions. Light-weight thermal sensors are commonly used in microprocessors due to their low hardware cost. However, large variations in process parameters and power supply voltage are common in processors and can significantly affect the behavior of these light-weight sensors. *Inaccurate* measurements are defined as sensor readings which are outside a temperature range under a predetermined confidence level. The main cause of inaccurate measurements is thermal sensor miscalibration. Measurements from miscalibrated thermal sensors may cause unnecessary performance penalties by negatively impacting dynamic voltage and frequency scaling (DVFS) and other thermal management, leading to long-time reliability degradation [1]. The accuracy of on-chip sensor measurement therefore is a major concern for many-core systems. In this paper, the identification of miscalibrated thermal sensors at run-time is targeted.

For our first contribution, we propose a model for the fast determination of expected temperature readings of a specific thermal sensor based on process parameters. For our second contribution, we introduce a method for the effective and efficient run-time detection of inaccurate thermal measurements. This method considers temperature correlations between multiple sensors based on their distance, their sampled values, and their expected values. We determine an uncertainty model for a sensor by not only using readings from the sensor but also using readings from adjacent sensors. A probability distribution function (pdf) estimate for a specific sensor is determined from the pdfs of its neighbors and the distances between the neighbors and the target sensor using linear opinion pooling [2]. The pdf estimate is then used to evaluate sensor measurements to determine the level of sensor accuracy.

978-3-9815370-0-0/DATE13/©2013 EDAA

II. Background

Several methods for constructing temperature profiles from a limited number of on-chip thermal sensors have been introduced. In Liu [3], the Kriging method is used to construct temperature profiles. Although this approach was shown to be effective in terms of on-chip temperature estimation accuracy, it has a high complexity of $O(N^3)$ for N sensors. In Sharifi *et al.* [1], results from power traces are filtered to estimate temperatures at on-chip locations. The filtering part alone has a complexity of $O(N^3)$ for N sensors. In Cochran *et al.* [4], spectral techniques were proposed for temperature profile estimation. This method was shown to be more effective in constructing temperature profiles from on-chip thermal sensors than the Kriging method but it also has high complexity. Both approaches attempt to estimate unknown values (e.g., temperature) at locations without sensors from known ones reported by sensors in different locations. The method explained in this paper is fundamentally different from these approaches. All measurement values reported by sensors are initially assumed to be unreliable. The measurement values are later tested to see if they are within the acceptable range. Additionally, our approach is designed to be performed at run-time, so execution time performance is important.

III. Thermal Sensor Measurement Probability Distribution Model

For each reading from a thermal sensor, a certain amount of measurement uncertainty exists. In this paper, we consider using ring oscillator-based thermal sensors for temperature monitoring. The measurement uncertainty of both calibrated and uncalibrated ring oscillator-based thermal sensors can be represented with a pdf [5] determined from a multi-variable model. The frequency of a ring oscillator (f) is determined by the propagation delay of each inverter, which consists of the transition time from high voltage to low voltage t_{PHL} and the transition time from low voltage to high voltage t_{PLH}, as shown in (1). N is the total number of inverters in the ring oscillator.

$$f = \frac{1}{N(t_{PHL} + t_{PLH})} \quad (1)$$

The transition time t_{PHL} can be calculated using (2).

$$t_{PHL} = \frac{2C}{\mu_n C_{ox}(W/L)_n(V_{DD} - V_t)}\left(\frac{V_t}{V_{DD} - V_t} + \frac{1}{2}\ln\frac{3V_{DD} - 4V_t}{V_{DD}}\right) \quad (2)$$

In this equation, μ_n is the electron mobility for NMOS, C_{ox} is the capacitance per unit gate area, $(W/L)_n$ is the width and length ratio for the NMOS inverter, C is the effective load capacitance, V_{DD} is the supply voltage and V_t is the threshold voltage. Parameters W, L and C_{ox} are subject to process variation and V_{DD} fluctuates at run-time. Assuming these parameters to be Gaussian random variables [5], we performed a Monte Carlo simulation with randomized parameter values. The values of the parameters in (2) are evaluated for 45nm technology. We optimistically assume 1% standard deviations for W, L, C_{ox} and V_{DD}. The results of the Monte Carlo experiment indicate that the resultant pdf of the measured temperature matches a Gaussian distribution with more than a 99% correlation coefficient (also noted by Zhang *et al.* [5]). The standard deviation of this Gaussian distribution changes as the true temperature changes. As a result, the standard deviation of measured sensor output temperature, $T^{measured}$, can be estimated as follows. For simplicity, as in [5], the t_{PLH} and t_{PHL} are considered the same in the following analysis. The measured temperature is given by (3).

$$T^{measured} = F(W, L, C_{ox}, V_{DD}) = \frac{a}{2Nt_{PHL}} + b \quad (3)$$

In this equation, W, L, C_{ox} and V_{DD} are variables whose uncertainties lead to the variation of the measured temperature. Taylor expansion is performed around the nominal value of these variables (which leads to T_0), as shown in (4) (we use T in (4) and (5) instead of $T^{measured}$ for simplicity). The variance of the measurement can be determined by (5).

$$T \approx T_0 + \frac{\partial T}{\partial W}dW + \frac{\partial T}{\partial L}dL + \frac{\partial T}{\partial C}dC + \frac{\partial T}{\partial V_{DD}}dV_{DD} \quad (4)$$

$$\sigma_T^2 = \left(\frac{\partial T}{\partial W}\right)^2\sigma_W^2 + \left(\frac{\partial T}{\partial L}\right)^2\sigma_L^2 + \left(\frac{\partial T}{\partial C}\right)^2\sigma_C^2 + \left(\frac{\partial T}{\partial V_{DD}}\right)^2\sigma_{V_{DD}}^2 \quad (5)$$

The standard deviation was calculated using (5) over the T^{true} temperature range of 50-100°C (323-373K), which is the common processor working temperature range. Results from Monte Carlo simulation with 100,000 samples for each temperature showed that the standard deviation of the measured temperature decreases as the true temperature increases. This trend is almost linear within the temperature range (50-100°C) that is of interest. A linear curve fits the standard deviation trend with more than a 95% correlation coefficient. A *simplified* thermal sensor measurement probability distribution function is summarized in (6) and (7).

$$T^{measured} \sim \mathcal{N}\left(T^{true}, \sigma(T^{true})\right) \quad (6)$$

$$\sigma\left(T^{true}\right) = cT^{true} + d \quad (7)$$

In this model, the measured temperature is a Gaussian distribution with the expectation as the true temperature and the standard deviation as a linear function of the true temperature. The value for the two parameters c and d in (7) can be estimated via curve-fitting, given the expectations and standard deviations for W, L, C_{ox} and V_{DD}. This simplified model for $T^{measured}$ and σ_T in (6) and (7) significantly reduces the total number of multiplications and additions versus the previous approach in [5]. The pdf of a thermal sensor measurement is denoted as $p(T^{measured})$.

Algorithm 1 Inaccurate Measurement Detection

Input: The reported temperature $T_i^{reported}$, $1 \le i \le N$ and coordinates of each of the N sensors.
Output: For each sensor, is its measurement inaccurate or not?
Initilization: Set all reported temperature to be NOT inaccurate.

1: **for** each sensor i such that $1 \le i \le N$ **do**
2: **for** each adjacent sensor k, $1 \le k \le N$ and $k \ne i$ **do**
3: **if** $T_k^{reported}$ is NOT inaccurate **then**
4: Calculate the probability distribution of sensor i's measurement $p_{i,k}(T_i^{measured})$ based on sensor k's reported temperature ($T_k^{reported}$) and the distance between sensor i and k (See Section IV-A).
5: **end if**
6: **end for**
7: Combine probability distributions $p_{i,k}(T_i^{measured})$ calculated from adjacent sensors using linear opinion pooling to get an aggregated distribution of sensor i's measurement $p_i(T_i^{measured})$(See Section IV-B).
8: Test sensor i's reported temperature, $T_i^{reported}$, against its aggregated distribution $p_i(T_i^{measured})$, to see if it is inaccurate or not (See Section IV-C).
9: **end for**

IV. A Method For Detection Of Inaccurate Temperature Measurements

We first describe pdf generation for a specific target sensor in which the thermal sensor's pdf is estimated from adjacent sensors' measured temperatures (not pdf) and their distance from the target sensor. In Section IV-C, we describe how the pdf can then be used to determine whether a temperature reading is accurate or inaccurate for the target sensor.

Each thermal sensor i reports a measurement (at one sample point), which is denoted as $T_i^{reported}$ (this is a *fixed value*, not a random variable). We would like to know if this measurement is inaccurate based on adjacent sensors' measurements. Our method estimates the probability distribution of each thermal sensor's (sensor i) measurement, which is denoted as $p_i(T_i^{measured})$. $T_i^{measured}$ is a random variable. This probability distribution is calculated based on adjacent sensors' reported temperatures. Then, the reported temperature ($T_i^{reported}$) for sensor i is tested against its estimated probability distribution ($p_i(T_i^{measured})$) to find out if it is inaccurate. If a sensor is determined to be inaccurate, it is excluded from subsequent accuracy calculations for its neighbors. Our method for pdf generation for N sensors is summarized in Algorithm 1.

TABLE I
THE TRUE AND MEASURED TEMPERATURES OF SENSORS 1 AND 2

Sensor	True Temp.	Measured Temperature
1	T_1^{true}	$T_1^{measured} = T_1^{true} + \Delta T_1$
2	T_2^{true}	$T_2^{measured} = T_1^{true} + \Delta T + \Delta T_2$

A. Inclusion of Spatial Correlation in the Distribution Model

As an example, we calculate the probability of sensor 1's measurement $p_{1,2}(T_1^{measured})$ based on sensor 2's measurement. The difference between the measured and the true temperature (ΔT_1 and ΔT_2 for sensor 1 and 2 in Table I) therefore can be modeled as a Gaussian distribution with zero mean and the same standard deviation in (7).

Assuming that the difference between the true temperature at sensor 1 and sensor 2 is $\Delta T_{1,2}$ (which we consider to be a random variable since it is unknown), the difference between the measured temperature at sensor 1 and sensor 2, denoted as $D_{1,2}$, is shown in (8).

$$T_2^{measured} - T_1^{measured} = D_{1,2} = (\Delta T_2 - \Delta T_1) - \Delta T_{1,2} \tag{8}$$

$$p_{1,2}(T_1^{measured}) = p_{1,2}(T_2^{measured} - D_{1,2}) \approx p_{1,2}(T_2^{reported} - D_{1,2}) \tag{9}$$

The probability distribution of the measurement at sensor 1 $p_{1,2}(T_1^{measured})$ thus could be calculated using (9). Although $T_2^{measured}$ is a random variable, as explained in Section III, here we replace it with the reported temperature from sensor 2 ($T_2^{reported}$) to provide an estimation, as shown in (9). In order to calculate the probability distribution of $D_{1,2}$, we consider $(\Delta T_2 - \Delta T_1)$ and $\Delta T_{1,2}$ to be independent, thus it can be calculated as the convolution of these two parts. This is a reasonable assumption since the first part $(\Delta T_2 - \Delta T_1)$, represents the subtraction of the uncertainty of sensor 2 and sensor 1, while $\Delta T_{1,2}$ represents the true temperature difference between these two sensors. These two variables are not physically related.

Monte Carlo simulation shows that the probability distribution of $(\Delta T_2 - \Delta T_1)$ is a Gaussian distribution with the standard deviation almost linearly decreasing as the true temperature increases through 50-100°C, similar to the one shown in (6) and (7). Since the true temperature is unknown, we use the average of the reported temperatures at sensor 1 and 2 ($T_1^{reported}$ and $T_2^{repored}$) to calculate the pdf of $(\Delta T_2 - \Delta T_1)$ in (8).

B. Bounding the True Temperature Difference

Assuming that the maximum true temperature difference is T_{diff}, it is possible to assume that within the bounds ($\pm T_{diff}$), the probability of the temperature difference is uniform. Thus, the probability distribution of the true temperature difference between sensor 1 and sensor 2 ($\Delta T_{1,2}$) can be modeled as a uniform distribution shown in (10).

$$p(\Delta T_{1,2}) = \begin{cases} \dfrac{1}{2T_{diff}}, & -T_{diff} < \Delta T_{1,2} < T_{diff} \\ 0, & \Delta T_{1,2} > T_{diff} \text{ or } \Delta T_{1,2} < -T_{diff} \end{cases} \tag{10}$$

Our estimation of the maximum true temperature difference between sensor 1 and 2 (T_{diff}) is based on (11) [6].

$$T_{diff}(r) = T_{diff-max} \times \left(1 - e^{-\frac{2r}{K}}\right) \tag{11}$$

In this equation, r is the distance between two thermal sensors. $T_{diff-max}$ is the difference between the maximum and minimum temperature values when the heat source (which is located at either sensor 1 or sensor 2s position) allows T_{diff} to reach its local maximum power density. K is the thickness of the processor package.

C. Combining Distributions from Adjacent Sensors

To combine the pdfs from surrounding sensors together (e.g. for sensor 1) we use the linear opinion pooling method [2]:

$$p_1\left(T_1^{measured}\right) = \sum_{k=2}^{8} \omega_k p_{1,k}\left(T_1^{measured}\right) \tag{12}$$

Linear pooling allows for the weighted combination of multiple pdfs forming an aggregated pdf. We choose to have the weight assigned to each probability distribution from adjacent sensors decrease exponentially as the distance (denoted as $dis(1,i)$) between them increases, as shown in (13). The parameter h in this equation is a constant. Our weight determination supports the general intuition that the closer two sensors are, the higher the correlation of temperatures.

$$\omega_i = \frac{h}{e^{2 \times dis(1,i)/K}} \quad \text{here, } \sum_{i=2}^{8} \omega_i = 1 \tag{13}$$

In the last step, the reported temperature from sensor 1, $T_1^{reported}$, is compared against the aggregated probability distribution of sensor 1's measurement $p_1(T_1^{measured})$. It is considered to be inaccurate when it falls into a region with temperatures that happen with very low probability (either the reported temperature is too high or too low). We use the following two equations to test the reported temperature. In these two equations, γ is the confidence level.

$$Prob\{T_1^{measured} < T_1^{reported}\} < \frac{1-\gamma}{2} \tag{14}$$

$$Prob\{T_1^{measured} > T_1^{reported}\} < \frac{1-\gamma}{2} \tag{15}$$

The reported temperature $T_1^{reported}$ is determined to be inaccurate if it falls into one of the two tails of the pdf (e.g. $T_1^{reported}$ is in the region on the far left of the curve (Eq. (14), or $T_1^{reported}$ is on the far right (Eq. (15)).

V. EXPERIMENTS AND RESULTS

We simulate our inaccurate measurement detection method using a 128-core system based on clusters of 90nm, 8-core UltraSPARC T1 architectures which encompass $115mm^2$ each [7]. After technology scaling to 45nm, the total area of the 128-core system is estimated to be $460mm^2$. We assume that thermal sensors are located in the center of the function blocks for the purpose of calculating distances. For the purposes of

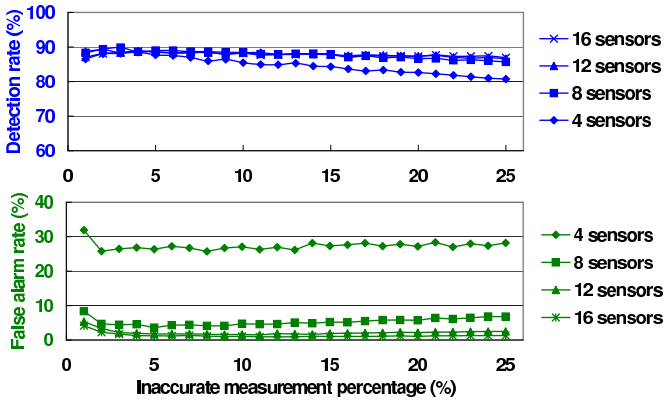

Fig. 1. Detection rates and false alarm rates at different thermal sensors per core count (4, 8, 12 and 16) with a confidence level (γ) of 90%. Inaccurate measurements are at least $\pm 10°$C off the true temperature

experimentation, the true temperatures are the values determined by HotSpot. Uncertainty is randomly added to values for each sensor based on the thermal sensor measurement probability distribution function described in Section III. An accepted value for $T_{diff-max}$ (35°C) [6] is used along with a K value of 6 mm (scaled from the 180nm technology in [6] to 45nm technology) in (11). The confidence level γ used in the algorithm in Section IV is set to 90%, which has been found to be optimal in our simulation. Only adjacent sensors in the same core are considered for inaccurate measurement detection (explained in Section IV) since these sensors are spatially local and share the same V_{DD}.

It is well-known that application execution can cause localized supply voltage fluctuations. If a thermal sensor is subjected to this fluctuation, its temperature reading may vary based on the expression shown in (2). The results for the experiment are shown in Fig. 1. The x-axis in Fig. 1 is the fraction of inaccurate measurements within all the measurements (for example, 1024 measurements from 1024 thermal sensors for 128 cores with 8 thermal sensors each). The fraction of inaccurate measurements that can be detected using our method is called the detection rate, as shown in the top figure in Fig. 1. Those measurements that are falsely identified as inaccurate (although they are not) are called false alarms. The false alarm rate is the ratio of the total number of false alarms to the total number of inaccurate measurements, as shown in the bottom figure in Fig. 1. The results are averaged over 200 temperature profiles generated using SESC and HotSpot. Fig. 1 shows the detection rate and the false alarm rate, as the inaccurate measurement percentage (fraction of all measurements which are inaccurate) increases from 1% to 25%. Fig. 1 shows that our method can detect more than 90% of inaccurate measurements for at least 12 sensors per core when the inaccurate measurement percentage is less than 7%. The false alarm rate is less than 6%.

The computational cost is represented as the total number of multiplications and additions involved in the calculation, as shown in Table II. N is the number of thermal sensors per core. For the spectral method, the computational cost is related to the total number of samples M, which is linearly proportional to N [4]. As seen in Table II, the computational cost of our method increases linearly as the number of sensors per core increases, while for the spectral method the relationship is $O(N\log N)$. The computational cost of our method is around 1/3 that of the spectral method [3] while the accuracies of the approaches are in the same range. Although not shown in this table, the computation cost of using the Kriging [3] and filtering methods [1] described in Section II for detecting inaccurate measurements increases in $O(N^3)$, which is not desirable for future many-core systems with many on-chip thermal sensors.

TABLE II
COMPUTATIONAL COST COMPARISON

	Multiplications and additions		Estimated exec. time using one core (ms)	
Sensors	Ours	Spectral [4]	Ours	Spectral [4]
8	92, 000	260, 000	0.38	1.08
12	144, 000	415, 000	0.60	1.73
16	196, 000	578, 000	0.81	2.41
Complexity	13, 000(N-1)+250	40× ($2M\log M$+M) $M = 361$ when $N = 8$	N/A	N/A

VI. CONCLUSION

In this paper, an efficient run-time method of evaluating the measurement uncertainty of on-chip thermal sensors is described. This uncertainty can be calculated relative to a thermal measurement probability distribution function. Linear opinion pooling is used to generate a combined pdf for the target sensor based on thermal measurements and distances between sensors. Inaccurate measurements can be determined with over 90% accuracy. The computational cost of our method is significantly improved versus other techniques making the approach's use at run-time more feasible.

REFERENCES

[1] S. Sharifi and T. S. Rosing, "Accurate Direct and Indirect On-Chip Temperature Sensing for Efficient Dynamic Thermal Management," *IEEE Trans. on CAD*, vol. 29, no. 10, pp. 1586–1599, Oct. 2010.
[2] W. Edwards, R. F. Miles, and D. von Winerfeldt, Eds., *Advances in Decision Analysis: From Foundations to Application*. Cambridge University Press, 2007.
[3] F. Liu, "A General Framework for Spatial Correlation Modeling in VLSI Design," in *Proc. Design Automation Conf.*, Jun. 2007, pp. 817–822.
[4] R. Cochran and S. Reda, "Spectral Techniques for High-resolution Thermal Characterization with Limited Sensor Data," in *Proc. Design Automation Conf.*, Jul. 2009, pp. 478–483.
[5] Y. Zhang and A. Srivastava, "Accurate Temperature Estimation Using Noisy Thermal Sensors for Gaussian and Non-Gaussian Cases," *IEEE Trans. on VLSI Systems*, vol. 19, no. 9, pp. 1617 –1626, Sep. 2011.
[6] K.-J. Lee, K. Skadron, and W. Huang, "Analytical Model for Sensor Placement on Microprocessors," in *Proc. IEEE Int'l Conf. on Computer Design*, Oct. 2005, pp. 24–30.
[7] A. Coskun, J. Ayala, D. Atienza, T. Rosing, and Y. Leblebici, "Dynamic thermal management in 3d multicore architectures," in *Proc. IEEE/ACM Design, Automation and Test in Europe*, Apr. 2009, pp. 1410–1415.

Work funded by the Semiconductor Research Corp. under Task 2083.001