Hot Spot Identification and System Parameterized Thermal Modeling for Multi-Core Processors Through Infrared Thermal Imaging

Sheriff Sadiqbatcha*, Hengyang Zhao*, Hussam Amrouch†, Jörg Henkel†, Sheldon X.-D. Tan*,
* Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA
† Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Abstract—**Accurate thermal models suitable for system level dynamic thermal, power and reliability regulation and management are vital for many commercial multi-core processors. However, developing such accurate thermal models and identifying the related thermal-power relevant spatial locations for commercial processors is a challenging task due to the lack of information and available tools. Existing tools such as HotSpot-like thermal models may suffer from inaccuracy or inefficiency for online applications, primarily because most rely on parameters that cannot be precisely quantified, such as power-traces, while others are numerical methods not suitable for runtime use. In this work, we propose a novel approach to automatically detecting the major heat-sources on a commercial multi-core microprocessor using an infrared thermal imaging setup. Our approach involves a number of steps including 2D discrete cosine transformation filter for noise reduction on the measured thermal maps, and Laplacian transformation followed by K-mean clustering for heat-source identification. Since the identified heat-sources are the thermally vulnerable areas of the die, we propose a novel approach to deriving a thermal model capable of predicting their temperatures during runtime. We apply Long-Short-Term-Memory (LSTM) networks to build a dynamic thermal model which uses system-level variables such as chip frequency, voltage and instruction count as inputs. The model is trained and tested exclusively using measured thermal data from a commercial multi-core processor. Experimental results show that the proposed thermal model achieves very high accuracy (root-mean-square-error: 2.04°C to 2.57°C) in predicting the temperature of all the identified heat-sources on the chip.**

I. Introduction

With rapid technology scaling, today's high performance microprocessors are becoming more thermally constrained due to steadily increasing power densities [1], [2]. Temperature has a profound impact on all the major long-term reliability effects such as electromigration (EM) for interconnects, and bias-temperature-instability (BTI) and hot-carrier-injection (HCI) for CMOS devices [3]. To enhance reliability, many system level thermal/power regulation techniques such as clock gating, power grating, dynamic voltage and frequency scaling (DVFS) and task migration have been proposed in the past [4]–[7]. One critical aspect of the aforementioned algorithms is correctly estimating the full chip temperature profile to properly guide the online thermal management schemes [8], [9]. However, accurate thermal estimation is a difficult task, especially for commercial off-the-shelf multi-core processors. Some of the existing methods depend on the on-chip temperature sensors. However, very few physical sensors are typically available, and they may not be located in close proximity to the true hot-spots on the chip, consequently misleading the temperature regulation decision [10]. Hence, the more popular solution is to supplement the data from the few on-chip sensors with estimated temperatures of all the prominent heat-sources on the chip via thermal models based on estimated power-traces. These methods offer higher spatial resolution as they allow for the temperature of all the heat-sources on the chip to be monitored during runtime [11]–[13].

This work is supported by NSF grant under No. CCF-1527324 and No. CCF-1816361.

Existing approaches consist of several bottom-up numerical methods such as HotSpot [11] based simplified finite difference methods, finite element methods [14], equivalent thermal RC networks [15], and the recently proposed top-down behavioral thermal models based on matrix pencil method [16] and the subspace identification method [17], [18]. However, the existing methods suffer from several drawbacks. First, most of the compact thermal models need accurate power-traces as inputs; but estimating the power of each functional unit (FU) of a practical microprocessor is not a trivial task, if not infeasible [19], [20]. On the other hand, from the system level thermal or power management perspective, the parameters that can be easily accessed are the frequency, voltage, and many other performance metrics natively supported by most commercial processors. Thermal models which are functions of those parameters will be more desirable and practical. Second, calibration of the compact models against the actual chip temperature under different workloads and thermal boundary conditions is very difficult. The reason being, measuring the temperature profile of a working chip under normal operation without the heat sink is a difficult task. Lastly, there is still a lack of a systematic way to determine the exact locations of hot-spots on the chip whose temperature should be monitored for dynamic thermal and power management.

Hence, in this work, we address all the aforementioned issues with the existing thermal models. Our novel contributions are as follows:

- We establish a lucid infrared (IR) thermal imaging setup with an advanced thermo-electric based rear-mounted cooling technique. This system allows us to obtain accurate online thermal maps of a working commercial multi-core processor.
- Secondly, we propose a novel approach to automatically locating the major heat-sources on a commercial microprocessor. Our approach involves 2D discrete cosine transformation (DCT) for noise reduction on the measured thermal maps, and Laplacian transformation followed by K-mean clustering for heat-source identification.
- We then apply Long-Short-Term-Memory (LSTM) networks to build a system-level hybrid thermal model capable of highly accurate online temperate predictions. The proposed model is parameterized with chip frequency, voltage, and other relevant high-level performance metrics and is trained and tested exclusively using thermal data measured directly from a commercial processor.
- Lastly, while this work is primarily intended for the chip manufacturers, it can also be implemented by the end users

978-3-9819263-2-3/DATE19/©2019 EDAA

as the proposed approach does not rely on any proprietary data such as the processor's floorplan or architectural details. This makes it desirable for applications such as aerospace and defense where mission critical systems with older generation processors and ICs are already deployed in the field.

II. Proposed Thermal Modeling Framework

A. The new thermal modeling and characterization overview

The proposed thermal modeling approach involves several critical steps. First and foremost, it requires an advanced IR thermography setup that is capable of recording lucid thermal maps of the commercial processor while its executing real workloads. This setup will be discussed in detail in the next subsection. The measured thermal maps acquired using this system will then be used to objectively determine the location of prominent heat-sources (or power-sources) on the commercial processor. Our novel approach to locating these heat-sources will be discussed in Sec. III. Once the heat-sources are located, the IR setup will once again be used to record time-series temperature data of all the identified heat-sources while the processor is subjected to a variety of practical workloads. At the same time, a suite of high-level performance metrics will be recorded in synchronous with the capture rate of the IR camera. Once sufficient data is acquired, a specialized Recurrent-Neural-Network (RNN) architecture called Long-Short-Term-Memory (LSTM) network will be employed to train the online thermal model. Once trained, the thermal model will be able to use the performance metrics as inputs to predict the temperature of all the identified heat-sources during runtime. This algorithm flow is illustrated in Fig. 1.

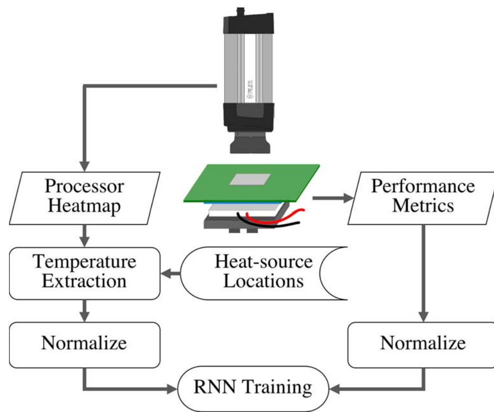

Fig. 1. Proposed algorithm flow

The proposed approach along with a discussion on how it differs from existing performance metrics based methods will be detailed in Sec. IV. As previously mentioned, we will assume that no proprietary information about the architecture or process-specific variables are known about the commercial processor. Our entire methodology will rely exclusively on the data that we measure directly from the processor itself.

B. Our IR thermography setup with rear-mounted cooling

One important aspect of the proposed approach is the acquisition of spatial and temporal thermal information from the commercial processor running a practical load. To achieve this, we have built a specialized IR thermography setup inspired by the recently proposed RAMA thermal imaging system [10]. This state-of-the-art setup features a thermo-electric based rear-mounted heat-extractor which offers a precisely controllable cooling solution where the processor is cooled from underneath, leaving the front side completely exposed to the IR camera. This introduces minimum interference with the IR emissions from the chip which is in stark contrast with the existing cool-liquid or oil-based front-cooling techniques where some sort of compensation or de-embeding is needed [20]. This allows us to capture lucid thermal images of the chip, while maintaining a safe operating temperature comparable to the traditional heat sink-based cooling solutions.

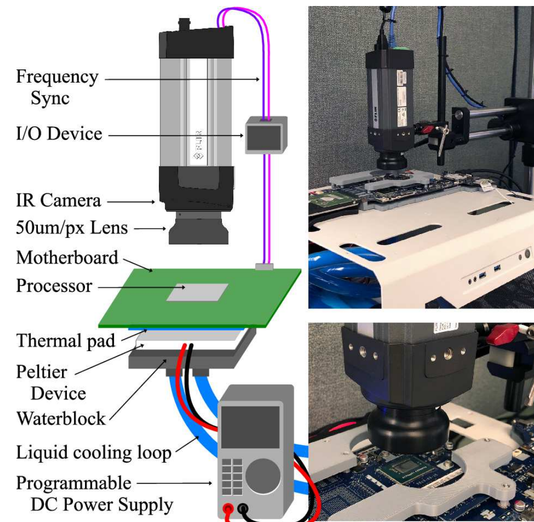

Fig. 2. Our IR thermography setup (Illustration on the left and photos on the right)

The IR setup that we have built (Fig. 2) has a slightly different configuration than what was presented in [10], as it is adapted specifically for the application presented in this study. The new setup consists of a FLIR A325sc IR camera with an image resolution of 16-bit 320x240 pixels and operating frequency of 60Hz. It can measure the temperature range from $0°C$ to $328°C$. Its spectral range (observable wavelength of electromagnetic radiation) is from $7.5\mu m$ to $13\mu m$. It has a microscope lens attachment that is used to achieve the spatial resolution of 50um/pixel. The IR camera has an internal waveform generator that generates a square waveform in synchronous with the capture rate of the camera. An I/O device is used to interface the waveform generator to the processor under test, so that the performance metrics (recorded internally) can be synchronized with the thermal data captured by the camera. The processor under test is an Intel i5-3337U, which has 2 cores with 2 threads per core. Mounted on the PCB directly underneath the processor is the thermo-electric based cooling system which includes a Peltier device powered by a programmable power source. A liquid cooling loop is used to cool the hot-side of the Peltier device.

III. Heat-source identification

One important aspect of building a thermal model for a processor is identifying the major hot-spots. These are the critical areas of the chip for many online or dynamic thermal/power management schemes. Typically those hot-spots are the locations of the major heat/power sources. As a result, locating these prominent heat-sources without the floorplan

and layout information becomes an important problem. In this section we will present our novel approach to locating these heat-sources on a commercial processor exclusively using measured thermal data.

A. Laplacian operation for heat-source identification

We start with the general thermal diffusion equation shown below [21]

$$\rho C_p \frac{\partial T}{\partial t} - \nabla(\kappa \nabla T) = g_T, \tag{1}$$

where T is temperature (K), ρ is the mass density of the material (kg$\cdot$m^{-3}), C_p is the mass heat capacity (J$\cdot$kg$^{-1}\cdot$K^{-1}), κ is the thermal conductivity and g_T is the heat energy generation rate (W $\cdot$ m^{-3}).

Since we deal with the spatial thermal map in two dimensions, we can ignore the transient terms in (1). We then have the steady-state thermal equation with heat-sources in the 2D case (assuming homogeneous material with location independent κ):

$$-\kappa \nabla^2 T(x, y) = g_T(x, y) \tag{2}$$

where ∇^2 is the Laplace operator. From the simplified heat equation (2), we can see that *the negative spatial Laplacian of the temperature distribution across the die is equal to the spatial heat generation*. Therefore, we can perform the 2D spatial Laplacian on a given thermal map to locate the underlining heat-sources $g_T(x, y)$. This method also works even if there is a thin heat-spreader layer with a conductive surface (for example a die with heat-spreader and package). This is due to the fact that, although the heat-spreader will distribute the heat across its surface and dissipate it, the spacial locations of the underlying heat-sources do not change.

To illustrate this idea, we simulate a simple structure in COMSOL Multi-Physics where three distinct heat-sources are placed below a thin heat spreader with a conductive boundary in-between. The simulation results (Fig. 3) show that, by applying 2D Laplacian transformation on the temperature distribution, T(x,y), observed on the heat spreader, the three distinct heat-sources can be easily identified.

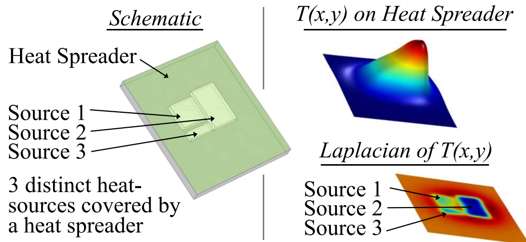

Fig. 3. COMSOL validation of the heat-source identification method

B. Fully automated heat-source identification method

In this subsection, we present our approach to identifying the major heat-sources using measured thermal-maps captured from a commercial processor. First, the raw thermal-map is pre-processed to remove the inherent noise present in measured data. After this, 2D spatial Laplacian is applied to locate the major heat-sources in 2D space. This process is repeated on ten-of-thousands of thermal-maps captured at different time points under different workloads. Lastly, a K-means based clustering algorithm is invoked to find the dominant heat-sources. The proposed method is illustrated in Fig. 4. For clarity, we will demonstrate the algorithm using the following example.

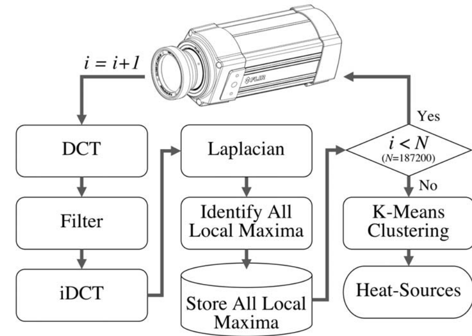

Fig. 4. Illustration of our novel heat source identification flow

1) Pre-processing for noise reduction via DCT: We start with a temperature map (i.e. Fig. 5) captured from the dual core processor under test.

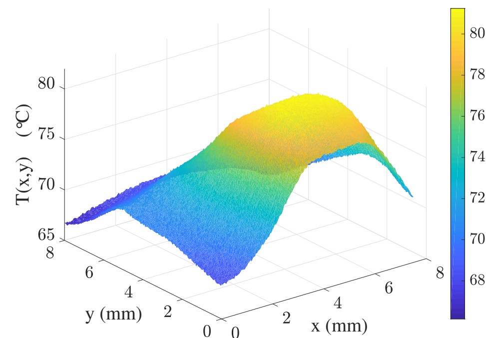

Fig. 5. Heatmap of the Intel i5-3337U captured using our IR system

The raw thermal map may contain noise, which must be removed as a pre-processing step. This step is crucial because the 2D discrete Laplacian

$$\begin{aligned}\nabla^2 f(x, y) = f(x-1, y) + f(x+1, y) + f(x, y-1) \\ + f(x, y+1) - 4f(x, y),\end{aligned} \tag{3}$$

is very sensitive to local difference of adjacent pixels [1]. 2D discrete cosine transformation (DCT) filter is an effective method for eliminating high-frequency noise, by transforming the heatmap into spatial frequency domain, masking the high-frequency components, and then transforming back to the original space domain. A 2D DCT consists of two separate 1D DCT operations, which can be denoted as

$$f_k = \frac{a_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{i=1}^{N-1} a_i \cos \frac{(2i+1)k\pi}{2N}, \; 0 \leq k < N, \tag{4}$$

where vector $\{a_i\}$ is the original data, and $\{f_k\}$ is the result of 1D DCT. A 2D DCT is completed by applying 1D DCT on each column and then on each row of the matrix. With the heatmap $T(x, y)$ transformed into its 2D frequency domain $F(x, y)$, a filtered frequency map $\mathcal{F}(x, y)$ can be obtained by applying a mask

$$\mathcal{F}(x, y) = F(x, y)m(x, y), \tag{5}$$

[1]For a heatmap with 177×166 pixels, with temperature ranging from 65°C to 80 °C, the laplacian range is approximately at ± 0.025°C/pixel^2. While with noise introduced, the laplacian can easily go up to ± 1.0°C/pixel^2, which is much higher than the useful laplacian component.

where $m(x, y)$ is the mask map valued 0 at high frequencies and 1 at low frequencies. The filtered heatmap $\mathcal{T}(x, y)$ is then obtained by taking the inverse 2D DCT on the filtered frequency map $\mathcal{F}(x, y)$. Similar to its forward counterpart, the inverse 2D DCT consists of two separate inverse 1D DCT steps on the rows and columns respectively. The inverse 1D transformation of (4) is

$$a_i = \frac{f_0}{\sqrt{N}} + \sqrt{\frac{2}{N}} \sum_{k=1}^{N-1} f_k \cos\frac{(2i+1)k\pi}{2N}, 0 \leq i < N. \quad (6)$$

This operation performed on the noisy heatmap previously shown in Fig. 5 results in the filtered heatmap shown in Fig. 6.

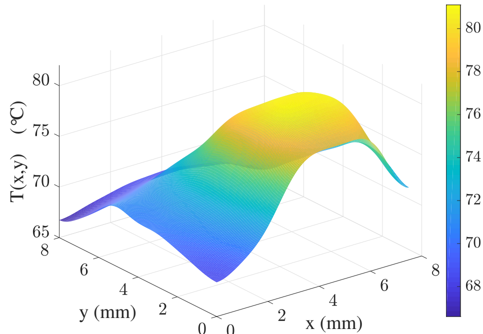

Fig. 6. The noise-reduced heatmap of the Intel i5-3337U

2) Temperature Laplacian for heat-source identification: The Laplacian operation in (2) can now be applied to the noise-less heatmap, which reveals the location of the internal heat-sources that were active during the time this particular heatmap was captured. These heat-sources can be identified by locating all the local maxima in the negative Laplacian of the temperature distribution as shown in Fig. 7.

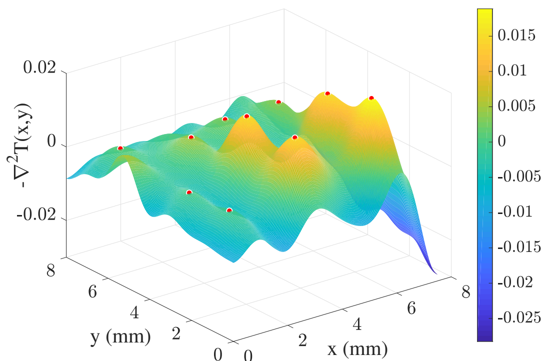

Fig. 7. Laplacian of the the heatmap with all local maxima identified

3) K-means clustering method for heat-source localization: While the above step can be used to identify the heat-sources that were active during the time the heatmap shown in Fig. 5 was recorded, there is no guarantee that all the prominent heat-sources within the chip were active during that time. In fact, many of the heat-sources are disabled at any given time due to the extreme power and clock gating used in modern processors. In order to ensure that most, if not all, of the prominent heat-sources on the chip are identified, we repeat the aforementioned heat-source identification process on many (about 187200) heatmaps that were collected while the processor is subjected to a multitude of different workloads with varying execution patterns. This process increases the chances of activating all the prominent heat-sources on the chip, at least once, so that their thermal signature can be recorded. The aggregate of the local maxima identified using this method is shown in Fig. 8.

This method results in dense clusters of heat-sources. However, it is not possible to track the temperature of each point in the cluster. Instead, we use the K-means clustering algorithm (using the "elbow criterion" to determine the value of k) to identify the centroids of the clusters. We will, from this point, refer to these centroids as our distinct heat-sources. In total, we were able to identify 18 prominent heat-sources on the dual core processor which has only 2 on-chip temperature sensors.

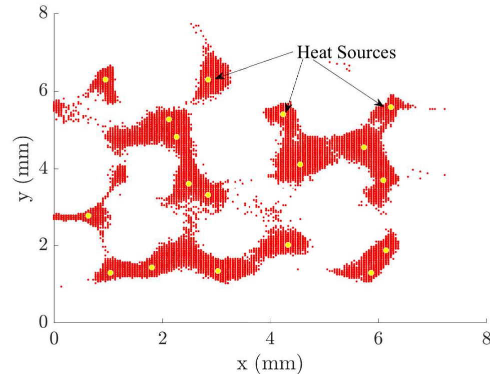

Fig. 8. Clusters (red) of local-maxima extracted from 187200 heatmaps of the Intel i5-3337U and heat-sources (yellow) identified using k-means

With all the heat-sources on the chip identified, our next goal is to derive a model that can be used to monitor their temperatures during runtime. The derivation of this model will be detailed in the next section.

IV. System-level Parameterized Thermal Modeling

In this section, we present the implementation of the recurrent neural network (RNN) based regression model as well as the acquisition and preparation of the training and testing data sets.

A. Runtime temperature

The heat-source identification method discussed in Sec. III allowed us to locate 18 distinct heat-sources on the dual core processor that was used for this study. With the thermal model, our goal is to accurately predict the temperature of these heat-sources during runtime. Hence, in order to train the regression based model, we need time-series temperature data of these 18 heat-sources. With the IR thermography setup (Fig. 2) we can directly record the temperature of the identified heat-sources while the processor is subjected to a variety of workloads. This temperature data measured directly from the processor gives us a great advantage in developing an accurate thermal model, as opposed to relying on another, previously established model to acquire the training and testing data-sets. In this study, we strictly use measured data for training, and later for testing the performance of the trained model.

B. Runtime performance metrics

While the IR setup allows us to capture the temperature of the processor externally, the other major part of the data-set comes from the internal state of the processor itself. One

way to monitor the internal state of the processor during runtime is through online performance monitoring software, which are supported by most, if not all, major manufacturers of commercial microprocessors. In this work we use high-level performance metrics provided by tools such as Intel's Performance Counter Monitor (PCM) [22]. These provide a high-level overview of the processor utilization with metrics such as the current frequency of the cores, instruction count, cache hit/miss rates, sleep-state residency, temperature from the internal sensors, etc. In total, PCM provides 80 performance metrics (P_1 to P_{80}) for the Intel i5-3337U used in this study. Since these performance metrics are a good representation of the processor's utilization, we can train a model which can accurately predict the temperature of the heat-sources using these metrics as inputs. Note, it is important to ensure that these performance metrics are captured in synchronous with the thermal data captured by the IR camera. Hence, as previously mentioned, the IR camera's internal waveform generator, along with an I/O device is used to synchronize the capture rate of the camera and the performance metrics recorded on the test system. This setup ensures that, at a frequency of 60Hz, one set of performance data is recorded in tandem with each set of temperature data captured by the IR camera.

Thermal models based on runtime performance metrics have been demonstrated in the past [23]–[25]. However, the existing methods are not practical to implement in modern commercial processors for several reasons. First, for each FU on the chip, the low-level performance metrics that have significant correlation with the power of the given FU must be manually identified. However, this is under the assumption that micro-benchmarks can be used to target a single FU in isolation so that the correlation can be determined, this is not feasible in modern processors. Second, even if these correlations can be found, the number of low-level performance metrics that can be recorded in parallel is limited by the number of programmable registers available in the processor. In the case of the Intel i5-3337U, only 11 registers were available. Since more than one metric is typically needed to model the temperature of a single FU, it is not possible to track the temperature of all the FUs on the chip in parallel. Alternatively, in this study we use high-level performance metrics offered by performance monitors such as Intel's PCM. The correlation between the transient behavior of the high-level performance metrics and the thermal response of the previously identified heat-sources are automatically learned through training. This make the proposed method more practical for modern commercial processors.

C. LSTM network

Since predicting the runtime temperature of a microprocessor is very much a time-series problem, we want to use a neural-network (NN) that has a feedback mechanism from the output back to the input, and is capable of maintaining several preceding time-steps of memory in its internal state. Recurrent-neural-networks (RNN) are the classical architecture designed for such a task. However, it has been well known that generic RNNs suffer from the so-called "vanishing gradient problem" which prevents them from handling most applications that require a substantial temporal resolution. Hence, in this study we will utilize a specialized subset of RNNs, called Long-Short-Term-Memory (LSTM) network, which uses gated internal states capable of processing much longer sequences of data. For brievity we refer the readers to [26] for detailed discussions and analysis of LSTM networks.

For our purposes, we will be utilizing the network shown in Fig. 9. The network has four hidden layers, where the first three are LSTM layers consisting of 100, 80, and 60 neurons respectively. The fourth is a dense layer with 18 neurons, matching the number of output dimensions. The LSTM nodes are configured to use the hyperbolic tangent (tanh) activation function, whereas the nodes in the final dense layer use linear activation functions. Dropout regularization of 20% is used after each of the LSTM layers to prevent the model from overfitting. The network takes in the 80 performance metrics (P_1 to P_{80}) as inputs, and outputs the estimated temperatures ($T_{HS\#1}$ to $T_{HS\#18}$) of the 18 heat-sources identified in Fig. 8.

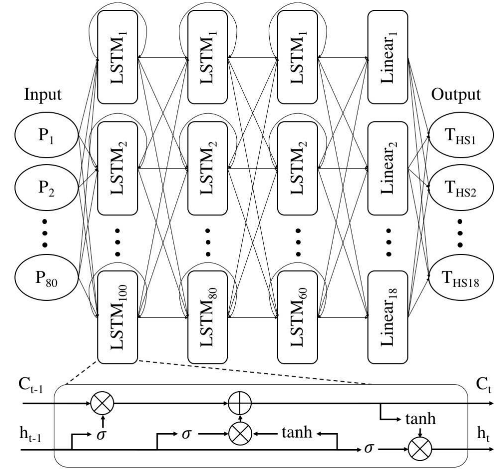

Fig. 9. LSTM network architecture

V. Experimental results and discussion

In this section, we present the results from the proposed system-level parameterized thermal modeling approach. For this study a total of 187200 data points were collected which, considering the capture rate of 60Hz, constitutes to 52 minutes of continuous runtime. Each data point consists of 80 performance metrics captured internally on the test system, and the temperatures of the previously identified 18 heat-sources captured via the thermal imaging setup. During this time, the processor was subjected to a variety of realistic workloads. These range from lightweight loads like idling, and word processing, to intensive loads like data compression. Some workloads were primarily compute-intensive tasks while others were memory-intensive. The idea is to task all the different functional units in the processor so that their thermal response can be recorded. Once all the data is acquired, the NN, shown in Fig. 9, was trained for a total of 100 epochs with 60 timesteps used for the LSTM layers. Out of the 187200 data-points collected for this study, 60% were used for training, 15% were used for validation, and the remaining 25% were used for testing. The training and testing data-sets were kept completely isolated from each-other in-order to ensure that no testing data is used in the training process.

Formal testing and validation carried out on the final model shows that it performs exceptionally well. The results presented in Fig. 10 shows the model predicting the runtime temperature of heat-source #1 for a duration of about 8 minutes. The measured temperature from the IR camera is overlaid on top of the prediction for comparison. For brevity we only show this plot for one heat-source, however the

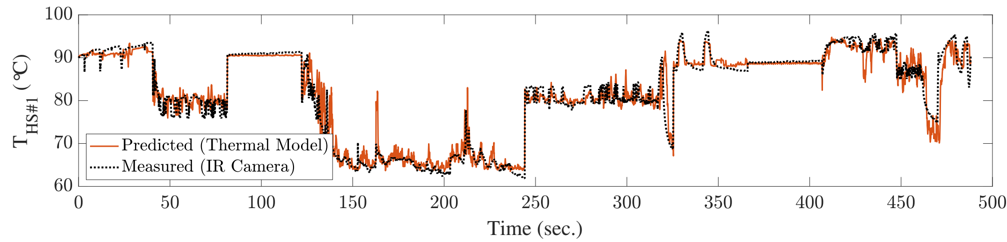

Fig. 10. Predicted vs measured runtime temperature of heat-source #1

root-mean-square-error (RMSE) computed for all 18 heat-sources is presented in Table. I. In summary, the highest RMSE was 2.57°C (HS #10) while the lowest was 2.04°C (HS #11). Considering the observed dynamic range of 58.73°C to 101.35°C, these constitute to a relative RMSE of 6.03% and 4.79% respectively.

TABLE I
ROOT-MEAN-SQUARE-ERROR FOR EACH HEAT-SOURCE (HS)

HS	RMSE	HS	RMSE	HS	RMSE
HS #1	2.36°C	HS #7	2.08°C	HS #13	2.46°C
HS #2	2.55°C	HS #8	2.27°C	HS #14	2.15°C
HS #3	2.49°C	HS #9	2.28°C	HS #15	2.38°C
HS #4	2.19°C	HS #10	2.57°C	HS #16	2.38°C
HS #5	2.49°C	HS #11	2.04°C	HS #17	2.55°C
HS #6	2.56°C	HS #12	2.29°C	HS #18	2.29°C

VI. CONCLUSION

In this article, we have proposed a novel method of systematically identifying all prominent heat-sources on a commercial processor and deriving a dynamic thermal model to predict the temperature of the identified heat-sources during runtime. Unlike many existing studies, this work exclusively utilizes measured data from a commercial off-the-shelf processor. Additionally, the proposed approach inherently avoids all the major obstacles faced by traditional methods that currently exist in literature, allowing it to be easily deployed on any off-the-shelf commercial processor. Experimental results show that the proposed thermal model achieves very high accuracy (root-mean-square-error 2.04°C to 2.57°C) in predicting the temperature of all the identified heat-sources on the chip. These results make the proposed approach very desirable for dynamic thermal management schemes which now rely heavily on the temperature data from a few on-chip temperature sensors.

REFERENCES

[1] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *Micro, IEEE*, vol. 32, no. 3, pp. 122–134, May 2012.

[2] M. Taylor, "A landscape of the new dark silicon design regime," *International Symposium on Microarchitecture (MICRO)*, vol. 33, no. 5, pp. 8–19, October 2013.

[3] "Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS)," 2003, in International Sematech Technology Transfer Document 03024377A-TR, 2003.

[4] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. IEEE Int. Symp. on High-Performance Computer Architecture (HPCA)*, Jan. 2001, pp. 171–182.

[5] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling," vol. 63, no. 2, pp. 349–360, February 2014.

[6] Z. Liu, X. H. S. X.-D. Tan, and H. Wang, "Task migrations for distributed thermal management considering transient effects," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 23, no. 2, pp. 397–401, Feb. 2015.

[7] H. Wang, J. Ma, S. X.-D. Tan, C. Zhang, H. Tang, K. Huang, and Z. Zhang, "Hierarchical dynamic thermal management method for high-performance many-core microprocessors," *ACM Trans. on Design Automation of Electronics Systems*, vol. 22, no. 1, pp. 1:1–1:21, July 2016.

[8] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *International Symposium on Computer Architecture*, 2003, pp. 2–13.

[9] J. Kong, S. W. Chung, and K. Skadron, "Recent thermal management techniques for microprocessors," *ACM Comput. Surv.*, vol. 44, no. 3, pp. 13:1–13:42, jun 2012. [Online]. Available: http://doi.acm.org/10.1145/2187671.2187675

[10] H. Amrouch and J. Henkel, "Lucid infrared thermography of thermally-constrained processors," in *2015 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, July 2015, pp. 347–352.

[11] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 14, no. 5, pp. 501–513, May 2006.

[12] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, "ISAC: Integrated space and time adaptive chip-package thermal analysis," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 1, pp. 86–99, 2007.

[13] H. Wang, S. X.-D. Tan, G. Liao, R. Quintanilla, and A. Gupta, "Full-chip runtime error-tolerant thermal estimation and prediction for practical thermal management," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, Nov. 2011.

[14] S. P. Gurrum, Y. K. Joshi, W. P. King, K. Ramakrishna, and M. Gall, "A compact approach to on-chip interconnect heat conduction modeling using the finite element method," *Journal of Electronic Packaging*, vol. 130, pp. 031 001.1–031 001.8, September 2008.

[15] Y. C. Gerstenmaier and G. Wachutka, "Rigorous model and network for transient thermal problems," *Mircroelectronics Journal*, vol. 33, pp. 719–725, September 2002.

[16] D. Li, S. X.-D. Tan, E. H. Pacheco, and M. Tirumala, "Parameterized architecture-level dynamic thermal models for multicore microprocessors," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 15, no. 2, pp. 1–22, 2010.

[17] T. Eguia, S. X.-D. Tan, R. Shen, D. Li, E. H. Pacheco, M. Tirumala, and L. Wang, "General parameterized thermal modeling for high-performance microprocessor design," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 2011.

[18] Z. Liu, S. X.-D. Tan, H. Wang, Y. Hua, and A. Gupta, "Compact thermal modeling for packaged microprocessor design with practical power maps," *Integration, the VLSI Journal*, vol. 47, no. 1, January 2014, in press, online access: http://www.sciencedirect.com/science/article/pii/S0167926013000412.

[19] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "Efficient power modeling and software thermal sensing for runtime temperature monitoring," *ACM Trans. on Design Automation of Electronics Systems*, vol. 12, no. 3, pp. 1–29, 2007.

[20] K. Dev, A. N. Nowroz, and S. Reda, "Power mapping and modeling of multi-core processors," in *International Symposium on Low Power Electronics and Design (ISLPED)*, Sept 2013, pp. 39–44.

[21] F. P. Incropera and D. P. DeWitt, *Fundamentals of Heat and Mass Transfer*, 5th ed. New York: John Wiley & Sons, 2002.

[22] Intel, "Intel Performance Counter Monitor (PCM)," https://software.intel.com/en-us/articles/intel-performance-counter-monitor.

[23] K. . Lee and K. Skadron, "Using performance counters for runtime temperature sensing in high-performance processors," in *19th IEEE International Parallel and Distributed Processing Symposium*, April 2005, pp. 8 pp.–.

[24] J. S. Lee, K. Skadron, and S. W. Chung, "Predictive temperature-aware dvfs," *IEEE Transactions on Computers*, vol. 59, no. 1, pp. 127–133, Jan 2010.

[25] H. Wang, S. X.-D. Tan, S. Swarup, and X. Liu, "A power-driven thermal sensor placement algorithm for dynamic thermal management," in *Proc. Design, Automation and Test In Europe. (DATE)*, March 2013, pp. 1215–1220.

[26] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.