A Learning-Based Autoregressive Model for Fast Transient Thermal Analysis of Chip-Multiprocessors

Da-Cheng Juan, Huapeng Zhou, Diana Marculescu, and Xin Li
Electrical and Computer Engineering, Carnegie Mellon University, PA, U.S.A.
{djuan, huapengz}@andrew.cmu.edu, dianam@cmu.edu, xinli@ece.cmu.edu

ABSTRACT

Thermal issues have become critical roadblocks for the development of advanced chip-multiprocessors (CMPs). In this paper, we introduce a new angle to view transient thermal analysis – based on predicting thermal profile, instead of calculating it. We develop a systematic framework that can learn different thermal profiles of a CMP by using an autoregressive (AR) model. The proposed AR model can serve as a fast alternative for predicting the transient temperature of a CMP with reasonably good accuracy. Experimental results show that the proposed AR model can achieve approximately 113X speed-up over existing thermal profile estimation methods, while introducing an error of only 0.8˚C on average.

I. INTRODUCTION

Power density is increasing in each generation of microprocessors since feature size and frequency are scaling faster than the operating voltage. Power density directly translates into heat, and consequently the operating temperature of a processor is getting hotter. In recent years, thermal issues have severely hindered the development of highly advanced and reliable chip multiprocessors (CMPs). Excessively high operating temperature is the root of many reliability issues, and can cause temporary timing errors as well as permanent physical damages. The rates of many failure mechanisms will increase exponentially with operating temperature [1]. Also, high operating temperature is known for increasing CMP's power consumption [1], especially leakage power. The increase of leakage power contributes to the increase of total power consumption, which in turn increases the operating temperature. This thermal-leakage positive feedback loop may lead to thermal runaway, which in the worst case may burn the chip.

A. PRIOR ART

Thermal modeling for CMPs has received a lot of attention recently. Accurate thermal modeling is the key to enable both thermal-aware designs and dynamic thermal management (DTM) [2][3][4]. Huang et al. [8] proposed Hotspot – an accurate, simulation-based thermal model for planar ICs – and the corresponding thermal-aware floorplanning. Li et al. [9] developed an efficient numerical method to solve large thermal grids for ICs. Bosch [10] demonstrated a thermal model with special focus on the heat flux distribution over the sides of a component. Sridhar et al. [11] developed 3D-ICE, a compact transient thermal model for fast thermal simulation of 3D ICs with inter-tier micro-channel cooling. Wang et al. [12] proposed a transient thermal simulator based on an alternating direction implicit method. Xu et al. [13] adapted the conventional flow for electrical RC network simulation to calculate the thermal profiles for 3D ICs with complex interconnect structures.

Although thermal RC simulation or finite-difference method (FDM) [14] used by prior arts usually guarantee a good accuracy in thermal modeling, these methods are very expensive in terms of execution time, especially when the required accuracy of transient temperature is high. Furthermore, accurately modeling the temperature-leakage feedback loop will incur extra invocations of costly thermal simulations. Generally, several days may be needed when a large amount of power configurations need to be examined for evaluating the thermal behavior of software applications or to explore the architectural design space in the early design stage. Such a long simulation time can become prohibitively expensive for computer architects or system designers.

Furthermore, dynamic thermal management (DTM) techniques heavily rely on thermal models which can efficiently estimate the temperature online [5]. Coskun et al. [6] adapted the performance counters, such as instruction per cycle (IPC), as a temperature estimator to perform thermal-aware job scheduling. Sharifi et al. [7] used Kalman filtering as an online thermal model for temperature prediction. The common point of these models is fast, and hence DTM techniques can be invoked within a short period to improve the performance or to reduce the peak temperature. From all the aforementioned reasons, an extremely fast thermal modeling that has reasonably good accuracy in capturing the transient thermal behaviors of CMPs is highly needed.

B. PAPER CONTRIBUTIONS

To the best of our knowledge, this paper brings the following novel contributions:

- We develop a learning-based autoregressive (AR) framework to enable fast and accurate transient thermal prediction, specially targeting CMPs. Compared to existing simulation-based models like [8], the proposed framework achieves approximately 113X speed-up, while introducing a root-mean-square-error (RMSE) of only 0.8˚C. The proposed framework can be applied to enable a wide spectrum of thermal optimizations or evaluation schemes, such as thermal characterization of software applications and proactive DTM.
- The proposed framework provides concrete, quantitative statistical inferences for the thermal behaviors of a CMP. Somewhat counter-intuitively, the inferences show that the single most important factor to influence the transient temperature is the temperature temporal correlation, rather than its spatial correlation, dynamic power, leakage power or other factors. The temporal correlation can account for approximately 66% of transient temperature changes.
- To demonstrate the effectiveness of our framework, we perform thermal optimization of a CMP by mapping workloads in a thermal-aware fashion. The experimental results show that, compared to the results from a popular thermal-aware mapping similar to [6], the proposed approach can further reduce the peak temperature by 2.9˚C on average.

C. PAPER ORGANIZATION

The remainder of this paper is organized as follows. Section II introduces the background knowledge. Section III provides the configurations used in this work. Section IV details the proposed AR framework for the transient thermal analyses of CMPs. Section V presents the implementation flow. Section VI demonstrates the experimental results. Section VII concludes this paper.

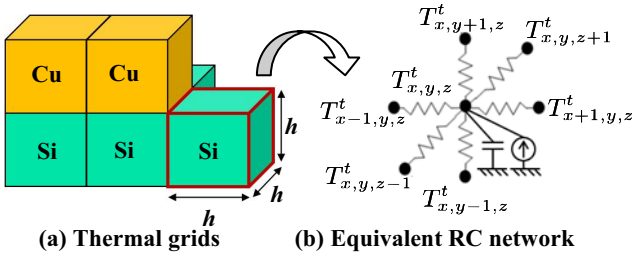
II. BACKGROUND

In this section, we present the detailed thermal modeling, and demonstrate the spatial and temporal correlations of temperature changes, which will be used in the proposed AR framework.

A. THERMAL MODELING

From a physical perspective, the temperature T is a function of time t and three spatial directions x, y and z. We use $T^t_{x,y,z}$ to denote the temperature of location (x,y,z) at a certain time point

978-1-4673-0772-7/12/$31.00 ©2012 IEEE



(a) Thermal grids **(b) Equivalent RC network**

Figure 1: Thermal RC model.

t. $T^t_{x,y,z}$ can be expressed by the heat equation that describes the heat flow in a given homogenous region over time:

$$\frac{\partial T^t_{x,y,z}}{\partial t} = \vartheta\left(\frac{\partial^2 T^t_{x,y,z}}{\partial x^2} + \frac{\partial^2 T^t_{x,y,z}}{\partial y^2} + \frac{\partial^2 T^t_{x,y,z}}{\partial z^2}\right) + q^t_{x,y,z} \quad \text{Eq(1)}$$

where ϑ is the material-dependent thermal diffusivity and q is the internally-generated heat [15]. Generally, finite-difference methods (FDM) are used to approximate the partially differentiated terms; for example, the central difference approximation is a popular method to approximate $\partial^2 T/\partial x^2$:

$$\frac{\partial^2 T^t_{x,y,z}}{\partial x^2} = \frac{T^t_{x+1,y,z} - T^t_{x,y,z}}{h^2} - \frac{T^t_{x,y,z} - T^t_{x-1,y,z}}{h^2} + O(h^2) = \frac{T^t_{x+1,y,z}}{h^2} - \frac{2T^t_{x,y,z}}{h^2} + \frac{T^t_{x-1,y,z}}{h^2} + O(h^2) \quad \text{Eq(2)}$$

where h is a sufficiently-small step size used to discretize the continuous variable x. $O(h^2)$ is the big O notation [16] used to represent the bound of accuracy loss due to the approximation. To consider the boundary condition, we assume that the environment temperature (or ambient temperature) is set to a given constant value and does not vary over time [8]. $\partial^2 T/\partial t^2$, $\partial^2 T/\partial y^2$ and $\partial^2 T/\partial z^2$ can be derived in a similar manner as Eq(2).

There is a well-known analogy between the solid heat conduction and the electrical current flow. The heat conduction can be modeled as a heat current flowing through thermal resistance and capacitance network [15], resulting in temperature differences. The values of thermal RCs depend on the material used to fabricate CMPs. For the purpose of thermal analysis, heat conduction is converted into electrical conduction; CMPs are divided into several cuboidal thermal grids as shown in Figure 1(a), and each thermal grid can be converted into an equivalent RC network as shown in Figure 1(b), with the temperature modeled as voltage and heat flow modeled as electrical current. Therefore, $T^t_{x,y,z}$ of Eq(1) is modeled as the voltage of grid (*x*,*y*,*z*) at the time frame *t* and q is modeled as the power consumption of a grid. In Figure 1(b), we can see that each node connects to six of its immediate neighboring nodes. This is because in many prior arts, such as [8], FDM similar to Eq(2), i.e., the central difference, is used to approximate the 2nd-order partial derivatives. The physical meaning behind Eq(2) is that "first-level" neighboring grids are used to capture the spatial correlation of temperature changes.

B. THERMAL CORRELATIONS

Heat conduction is a continuous process happening within a certain region and over a period of time. This continuous phenomenon makes temperature differences have both spatial and temporal correlations. More specifically, let us focus on *x-y* directions and rewrite Eq(1) into:

$$\frac{\partial T^t_{x,y}}{\partial t} = \vartheta\left(\frac{\partial^2 T^t_{x,y}}{\partial x^2} + \frac{\partial^2 T^t_{x,y}}{\partial y^2}\right) + q^t_{x,y} \quad \text{Eq(3)}$$

By using the approximation described in Eq(2) on $\partial^2 T/\partial x^2$, $\partial^2 T/\partial y^2$ and $\partial T/\partial t$, we will obtain [15]:

$$T^{t+1}_{x,y} = \frac{\vartheta u}{h^2}(T^t_{x\pm1,y} + T^t_{x,y\pm1}) + \frac{h^2 - 4\vartheta u}{h^2}(T^t_{x,y}) + u(q^t_{x,y}) + O(h^2) \quad \text{Eq(4)}$$

where u is the step size for time. Here we assume $h^2 > u$, so the accuracy loss is bounded by $O(h^2)$[1].

1. SPATIAL CORRELATION

The first term of Eq(4) represents the spatial correlation of temperature changes, and shows that the first-level neighboring grids are used to approximate $T^{t+1}_{x,y}$. If we further include the second-level neighboring grids, $\partial^2 T/\partial t^2$ can be expressed as Eq(5) by using Taylor's series:

$$T^{t+1}_{x,y} = \sum_{\ell=1}^{2} a_\ell (T^t_{x\pm\ell,y\pm\ell}) + b \cdot (T^t_{x,y}) + c \cdot (q^t_{x,y}) + O(h^4) \quad \text{Eq(5)}$$

where a_ℓ, b and c are constants derived from ϑ, h and u. Since $h << 1$, $O(h^4)$ is smaller than $O(h^2)$ in Eq(4), which means the accuracy loss decreases when higher-level neighboring grids are included in the model. Theoretically, when ℓ^{th}-level neighboring grids are included, the accuracy loss should be reduced and bounded by $O(h^{2\ell})$. In practice, a large ℓ will lead to an extremely-high complexity thermal model. In this paper, ℓ is empirically set to three to balance the accuracy and the model complexity.

2. TEMPORAL CORRELATION

The second term of Eq(4) or Eq(5) shows the temporal correlation between $T^{t+1}_{x,y}$ and $T^t_{x,y}$. In Eq(4), the step size u needs to be smaller than the thermal RC constant, τ, to guarantee the convergence of the numerical integration. According to [17][18], τ is usually in the range of 0.1–0.5ms. In addition, the authors of [18] pointed out that it takes at least 0.1ms to raise the transient temperature of CMPs by 0.1˚C. Hence, in this work we set the step size u to 0.1ms.

III. CONFIGURATIONS

Before elaborating on the proposed AR framework, we first introduce the architecture and dataset used herein. We introduce the micro-architecture and CMP architecture in Section III.A, followed by the dataset used to train and test the proposed model in Section III.B.

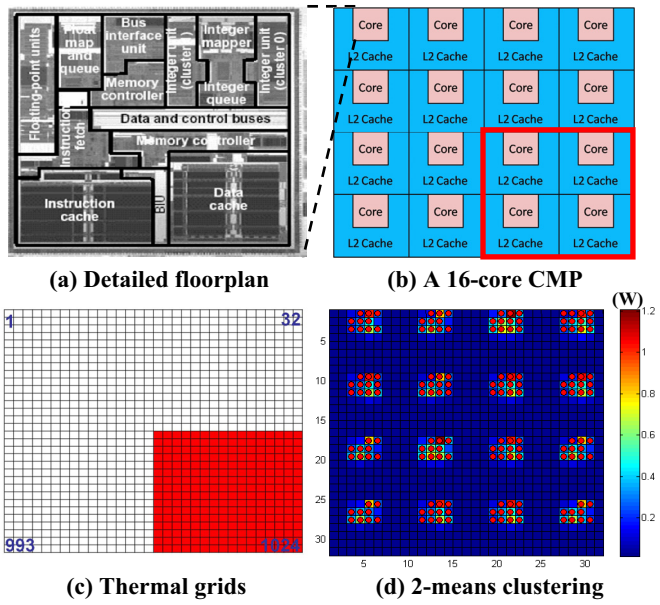
A. TARGET ARCHITECTURE

The architecture used throughout this paper is a symmetric CMP, consisting of 16 out-of-order Alpha 21264 cores [19]. The corresponding micro-architecture parameters are listed in Table 1. Figure 2(a) illustrates the floorplan of Alpha 21264 processing core [19]. This floorplan along with the L2 cache is replicated 16 times in a 4×4 mesh to create a planar 2D CMP. As shown in Figure 2(b), processing cores and caches are placed in a fine-grained, interwoven manner. For simplicity and without losing much accuracy, the target CMP is homogenously partitioned into 32×32 = 1,024 [18] thermal grids for analysis as shown in Figure 2(c). For example, four processors in the bottom-right corner of Figure 2(b) are mapped to the corresponding thermal grids in Figure 2(c). This resolution (32×32) of thermal grids can be changed according to the different requirements of accuracy. Note that thermal grids are distributed in *x-y* direction, instead of *x-y-z* as mentioned in Section II. This is because in the model proposed by [8], each grid implicitly includes all vertical components that generate heat, such

Parameters	Values
Number of cores	16
Frequency	3.0 GHz
Technology	45nm node with V_{dd}=1.0V
L1- I/D caches	64KB, 64B blocks, 2-way SA, LRU
L2 caches	1MB, 64B blocks, 16-way SA, LRU
Pipeline	7 stage deeps, 4 instructions wide

Table 1. Processor parameters

[1] Due to page limit, we do not include all details of the derivation.



(a) Detailed floorplan (b) A 16-core CMP

(c) Thermal grids (d) 2-means clustering

Figure 2: A CMP and the corresponding thermal grids.

as metal, active Si and substrates layers.

B. DATASET

In this paper, we use SPECcpu2000 [34] as workloads and the Hotspot [8] as the thermal simulator to characterize the thermal behavior of a CMP. The detailed implementation will be elaborated in Section V. The generated thermal responses are used as inputs to train and test the proposed AR model. The dataset contains 100 different power configurations and 513×1,024 thermal responses for each power configuration, while 513 is the number of time frames and 1,024 is the number of grids. Each time frame is set to 0.1ms [18]. In this work, we treat each grid (x,y) at a time frame t as a sample, so a total of $N = 100\times513\times1{,}024\approx10^7$ samples are used to train and test the proposed AR framework. The features of the dataset is described in Table 2. Each sample has P features, including five physical features plus ξ_{AR} autoregressive (AR) features. The five physical features of each sample include: its x location (X), y location (Y), radius (R), total power consumption (P_{tot}), and leakage power consumption (P_{leak}). R is calculated by $\sqrt{(X-mid)^2+(Y-mid)^2}$; mid is set to (32+1)/2 (since the resolution of thermal grids is 32×32). Also, P_{leak} is included in P_{tot}; we separate this term out because P_{leak} is more sensitive to temperature changes [20] and may potentially be a good thermal predictive feature [21].

As mentioned in Section II.B, $T^t_{x\pm\ell,y}$, $T^t_{x,y\pm\ell}$ and $T^t_{x,y}$ are highly correlated to $T^{t+1}_{x,y}$, and therefore these features should be included in the dataset to improve the prediction accuracy. These features are called AR features. Unlike the physical features above, AR features will be evaluated at each time frame. Therefore, for each sample, its AR features need to be updated on the fly. ξ_{AR} represents the number of AR features. In this work, ξ_{AR} is 13 because ℓ=3, such that $T^t_{x\pm\ell,y}$, $T^t_{x,y\pm\ell}$, ℓ = 1 to 3 and $T^t_{x,y}$ are included.

To better explain the proposed methodology, we denote $T^t_{x,y}$ as the thermal response of the i^{th} sample T_i, and both physical and AR features as $\boldsymbol{m^i} = (m_{i1},\cdots,m_{iP})$. The bold font represents a vector instead of a scalar. Here we focus only on the features of the dataset, which will be used to explain the proposed framework. More detailed implementaion about the dataset as well as this work will be presented in Section V.

Thermal responses	Samples×(Features)
100×513×1024 conf.×time×grids	100×513×1024×$(P=5+\xi_{AR})$ conf.×time×grids×(features)

Table 2. Features of the dataset

IV. METHODOLOGY

The proposed framework has two main components: *k*-means clustering and autoregressive (AR) model. *K*-means clustering serves as a pre-processing of the dataset, and based on the clustering results the AR model will learn the fitting coefficients to predict the temperature.

A. K-MEANS CLUSTERING

The goal of *k*-means clustering [22] is to partition the 1,024 thermal grids into k clusters such that each grid belongs to the cluster with the nearest mean power consumption. Therefore, grids in each cluster will have similar values of P_{tot}. Since the temperature of each grid is not known in advance, P_{tot} is used as a proxy criterion to cluster grids. Empirically, thermal profiles of certain functional blocks of a CMP are completely different from others – usually thermal hotspots are located in power-hungry (P_H) blocks such as integer arithmetic logic unit (IALU) and register files (RF) [18], depending on the behaviors of executed applications. These functional units lie within processing cores. In this context, it is necessary to separate thermal grids into two groups that represent power-hungry (P_H) and power-intermediate (P_I) blocks, respectively. Nevertheless, as we will show later, not all blocks in the processing cores are power-hungry. Therefore, the clustering cannot be performed simply based on the functionality of a block. For each cluster, a set of regression coefficients will be learned and plugged into the proposed AR model in order to predict the respective temperatures.

Based on the P_{tot} of each grid, we apply *k*-means clustering to separate thermal grids into P_H and P_I groups, so k is empirically set to two since P_H and P_I blocks have their distinct thermal profiles from the aforementioned observation. Given 1,024 of grids, $\boldsymbol{P_{tot}}$ of each grid g is a 3-dimension vector, where 3=100×513. To reduce complexity and without losing accuracy, we use the average of $\boldsymbol{P_{tot}}$ to perform the 2-means clustering that aims to partition the 1,024 grids into 2 sets, $\boldsymbol{S} = \{S_1,S_2\}$ so as to minimize the within-cluster sum of squares (WCSS):

$$WCSS = \arg\min_{\boldsymbol{S}} \sum_{g=1}^{1024}\sum_{k=1}^{2}\sum_{g\in S_k}\left\|\overline{P^g_{tot}}-\mu_k\right\|^2 \qquad \text{Eq(6)}$$

where $\overline{P^g_{tot}}$ represents the average power of the grid g and μ_k is the mean of $\overline{P^g_{tot}}$ in S_k. Eq(6) can be solved very efficiently by the methods proposed in [23][24] (not included due to space constraints).

Figure 2(d) shows the clustering results. P_H grids are identified by red circles, and the rest of grids are P_I grids. The color bar indicates the average power intensity of each grid in Watts (W). From Figure 2(a) and (d), we can see most of P_I grids lie within the L1 and L2 caches, whereas all P_H grids lie within processing cores. However, several functional blocks within processing cores, such as the floating point units, are assigned to the P_I group instead of the P_H group. A total of 169 grids are categorized as P_H grids. We want to point out that this number is design- and workload-dependent and may vary if these two factors change dramatically. Based on this clustering result, the original dataset is separated into two sub-dataset: one for P_H and one for P_I. The dimensions of each sub-dataset are N_{PH}=100×513×169 and N_{PI} =100×513×855, respectively. Since we apply the same AR framework on both P_H and P_I clusters, we only focus on the P_H cluster in the later sections for the conciseness of explanation.

B. LEARNING-BASED AR FRAMEWORK

The learning-based AR framework uses *Lasso* regression [25] as its kernel to predict T_i. Lasso regression consists of a linear regression model with L1 regularization. It shrinks the fitting coefficients and sets some of them to exact zero, and hence tends to retain only the highly relevant features to predict T_i. According to Eq(4) and Eq(5), T_i can be approximated by a linear function of predictive features $\boldsymbol{m^i}$:

$$T_i = \sum_{j=1}^{P} \alpha_j m_{ij} + \beta \qquad \text{Eq(7)}$$

where $(\boldsymbol{\alpha}, \beta)$ are fitting coefficients. As in the usual regression setup, m_{ij} are standardized so that $\sum_i m_{ij}/N_{PH} = 0$ and $\sum_i m_{ij}^2/N_{PH} = 1$, and T_i are assumed to be conditionally independent given m_{ij} since the potential correlations among T_i are already modeled by AR features in m_{ij}. Again, N_{PH} is the number of samples in the power-hungry cluster as mentioned in IV.A. The physical insight behind Eq(7) is that, in addition to the AR features and P_{tot}, the rest of the features are used to linearly converge to $O(h^{2\ell})$.

Before we elaborate on how to adapt Lasso regression to predict T_i, let us first introduce the coefficient-learning process for constructing a predictive model. In general, this process can be separated into two phases: the training phase and testing phase. The goal of the training phase is to learn the estimate of fitting coefficients $(\boldsymbol{\alpha}, \beta)$, denoted as $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ and $\hat{\boldsymbol{\alpha}} = (\hat{\alpha_1}, \cdots, \hat{\alpha_P})$. The $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ can be learned by:

$$(\hat{\boldsymbol{\alpha}}, \hat{\beta}) = \underset{(\boldsymbol{\alpha}, \beta)}{arg\ min} \left\{ \sum_{i=1}^{N_c} \left(T_i - \beta - \sum_{j=1}^{P} \alpha_j m_{ij} \right)^2 \right\}, \quad s.t. \sum_{j=1}^{P} |\alpha_j| \le \lambda. \qquad \text{Eq(8)}$$

where λ is the parameter to control the amount of shrinkage that is applied to the estimates. In this work, the solver provided by Friedlander [27] is used to optimize Eq(8) and learn $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$. Here, we use 10-fold cross validation (CV) [22] to select λ which results in the smallest root-mean-square-error (RMSE): the best value of λ is 1, selected from the range of 10^5 to 10^{-5}. CV is the unbiased error estimator and is widely used in statistics and machine learning domains [22].

In the testing phase, $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ are learned and plugged into Eq(7) and to calculate $\hat{T}_i$ as an estimate of T_i:

$$\hat{T}_i = \sum_{j=1}^{P} \hat{\alpha_j} m_{ij} + \hat{\beta} \qquad \text{Eq(9)}$$

By using Eq(9), $\hat{T}_i$ can be calculated instantly if $\boldsymbol{m^i}$ is given. No time-consuming thermal simulation is required in this phase. Please note that Eq(9) is different from Eq(7) because $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ and $\hat{T}_i$ are estimates, while $(\boldsymbol{\alpha}, \beta)$ and T_i of Eq(7) are actual values.

1. PREDICTION ACCURACY

To evaluate the accuracy of the proposed learning model, we again use 10-fold cross-validation to calculate the prediction error: Figure 3 shows the cross-validated prediction results for the P_H cluster; the X axis stands for the actual simulated results obtained with Hotspot, whereas the Y axis represents the predicted temperatures by using the learning-based AR model. Each color represents one instance of cross validation. As it can be seen in the figure, our thermal prediction is very accurate. The RMSE is 0.43˚C and the correlation coefficient (CC) is 0.99. For the P_I cluster, the prediction is even more accurate: the RMSE is 0.20˚C and CC is almost one. Therefore, just relying on the fitting coefficients $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ learned from the proposed framework, one can accurately predict the transient temperature for a CMP, without actually performing time-consuming thermal simulations.

2. COEFFICIENT ANALYSIS

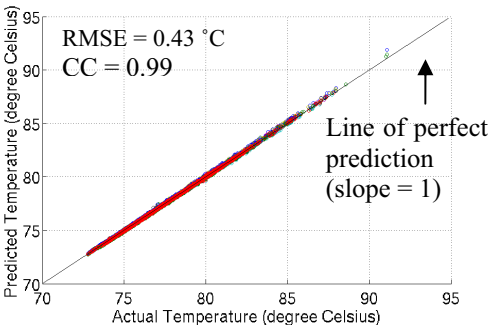


Figure 3: Prediction accuracy.

We also show the distribution of fitting coefficients for each feature, namely $\hat{\boldsymbol{\alpha}}$, for the P_H cluster. The physical meaning of $\hat{\boldsymbol{\alpha}}$ is the sensitivity of temperature changes to each predictive feature. Figure 4 shows the relative percentage of $\hat{\boldsymbol{\alpha}}$ in a pie chart. All notations here are the same as described in Section III. $T^t_{x\pm\ell,y\pm\ell}$ in Figure 4 represents the sum of $\hat{\boldsymbol{\alpha}}$ of $T^t_{x\pm\ell,y\pm\ell}$. We can see that $T^t_{x,y}$ dominates the prediction of $T^{t+1}_{x,y}$ by 66%. This is counter-intuitive because P_{tot} is generally considered as the most important factor to affect temperature. It is also worth mentioning that the $\hat{\boldsymbol{\alpha}}$ of R and P_{leak} are negative values. This is interesting because a grid with a large R actually means that it is located on or close to the rim of a CMP, which has better heat dissipation. Also, a grid with a large P_{leak} means that this grid idles often. Both these two phenomena lead to a lower temperature profile, and hence the corresponding coefficients of R and P_{leak} are negative.

The other interesting observation is that if the step size u increases from 0.1ms to 0.5ms and 1ms, its significance drops from 66% to 51% and 43%, respectively. In contrast, the significance of P_{tot} increases from 15% to 31% and 42%, respectively. As a result, the proposed AR model could automatically capture these physical properties via statistical learning, and reflects these phenomena by setting different values to $\hat{\boldsymbol{\alpha}}$. Finally, we examined the pole-zero plot of the fitted AR model, and found that all poles fall within the unit circle, which means the stability of the model is guaranteed [15].

3. FORWARD PREDICTION

So far, we have demonstrated how to predict T_i with $\boldsymbol{m^i}$. Recall that within $\boldsymbol{m^i}$, there are several AR features, such as $T^t_{x,y}$, which cannot be known in advance before the time frame evolves to *t*. Hence, we need to wait for these AR features to be known, in order to predict $T^{t+1}_{x,y}$. In other words, if we are interested in the transient temperature at the time frame *t+1*, we need to wait until the thermal estimation or measurement, such as the reading from a thermal sensor, at the time frame *t* is available. This restriction greatly reduces the capability of the proposed AR framework.

To handle the aforementioned problem, we develop a

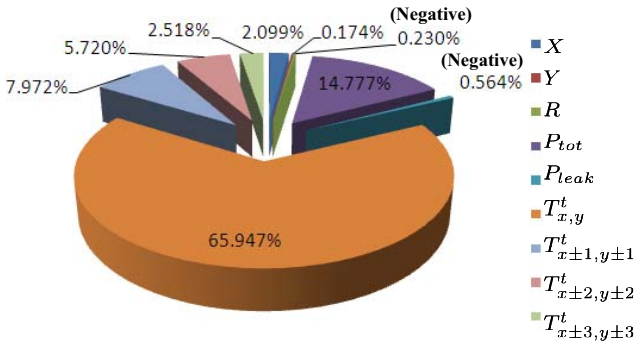


Figure 4: Coefficient distribution.

technique called *forward prediction*. The concept is simple: if $T_{x,y}^{t}$ is not available yet, but we need it to predict $T_{x,y}^{t+1}$ – we predict $T_{x,y}^{t}$ first and then use $\hat{T}_{x,y}^{t}$, i.e., the estimate of $T_{x,y}^{t}$, to predict $T_{x,y}^{t+1}$. The concept can be recursively applied until the time frame equals zero, i.e., all temperature values are the ambient temperature. The computational complexity of this forward prediciton is linear with the number of time frame N_T, denoted as $O(N_T)$, and hence can be efficiently computed. With this forward prediction technique, the proposed AR model could be used to predict the transient temperature at any time frame, without be restricted by AR features. The prediciton accuracy of the forward prediction will be demonstrated in Section VI.A.

4. DYNAMIC THERMAL MANAGEMENT (DTM)

The proposed AR model can be used to enable fine-grained DTM techniques. Once the model is trained offline, i.e., the fitting coefficients $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ are learned, the thermal prediction allowed by Eq(9) can be used online. Furthermore, since the overhead of this thermal prediction is very small (only one instance of matrix multiplication), the trained model can be plugged into the proactive DTM techniques, such as thermal-aware thread migration [3][6], to control the thermal behaviors of a CMP within a short interval of time. Note that in a real setting, the reading from thermal sensors could be used as thermal responses to train the proposed AR framework.

C. LIMITATION

There is a limitation of the proposed AR framework – the training cost. To learn the fitting coefficients $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$, the thermal response T_i of each $\boldsymbol{m^i}$ is required. In this paper, for the workloads considered, T_i is obtained via thermal simulations that take up to eight hours. However, this is a one-time training cost. Once $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ are learned, the transient temperature can be predicted instantly given $\boldsymbol{m^i}$. The model may need to be retrained if the underlying design changes significantly. For example, if the floorplan or cooling device of the target CMP changes, $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ may need to be relearned by the newly generated $\boldsymbol{m^i}$.

We would like to point out that the proposed framework is to serve as a faster alternative of existing thermal characterization frameworks and associated DTM techniques. The proposed framework relies on accurate temperature analysis or simulation to provide high quality training inputs to learn the fitting coefficients. Also, while the proposed methodology is generic, the model trained by this framework is not a general-purpose thermal models – it specifically targets and models the thermal profiles of a given CMP.

V. IMPLEMENTATION

In this section, we describe the experimental setup and the corresponding implementation flow in detail. To obtain the dataset described in Section III, we use modified SimpleScalar [28], Wattch [31], and Hotspot [8] for the performance, power, and thermal simulations, respectively. We modified the leakage power model in Wattch based on [32][33][36] for more accurate leakage values. Leakage currents are characterized by using HSPICE simulation with the 45nm high performance Predictive Technology Model [35]. For the Hotspot configuration, the chip size and spreader size are set to 0.03m×0.03m; sampling rate is set to 3×10^5 clock cycles; the parameters not mentioned here are assumed to be the default values. SPECcpu2000 benchmarks [34] are randomly selected to form 100 different multi-program workloads for a 16-core CMP. With the above settings, we perform a full-system simulation for 500 million instructions, and then collect the power profiles for the temperature simulation.

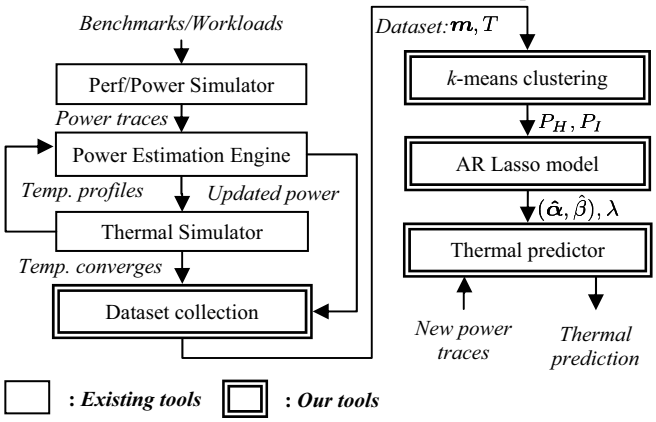


Figure 5: Overall implementation flow.

Figure 5 presents the overall flow of the proposed methodology. First, the multi-programmed workloads are fed in as inputs to the performance and power simulators, hereby providing both active and leakage power profiles. Second, the Power Estimation Engine collects temperature profiles and then updates the power values based on the current temperature value. The updated power values are fed into the temperature simulator to estimate the new temperature value. This temperature-power iteration will continue updating until the temperature value converges; the converged temperature and power profiles are collected and used as the dataset described in Section III.B.

After the dataset is obtained, *2*-means clustering is applied to separate grids into P_H and P_I groups as mentioned in Section IV.A. Finally, for each group, $(\hat{\boldsymbol{\alpha}}, \hat{\beta})$ and λ are learned by using AR Lasso model as mentioned in Section IV.B, and then plugged into Eq(7) to predict the transient temperature of a CMP under a new power configuration.

VI. EXPERIMENTAL RESULTS

This section presents the experiment results, including (1) transient thermal prediction by using forward prediction, and (2) thermal optimization by using workload mapping.

A. RESULTS OF FORWARD PREDICTION

Here, we demonstrate the accuracy of the forward prediction by using the proposed AR model. Figure 6 shows the RMSE (in Z axis) of each grid (in X axis) over each time frame (in Y axis). For better visualization, we pick 20 of the hottest grids in the P_H group as grids of interest. These grids of interest are often the location of themal hotspots, so our prediction needs to be accurate here. Note that the RMSE is calculated by using 10-fold cross validation. Generally, the RMSE of each grid stays around 0.7°C, and the highest error in Figure 6 is less than 1.1°C. The overall RMSE of every single grid over each time frame is 0.8°C.

Also, we are interested in the peak temperature prediction. Figure 7 illustrates the peak temperature (in Y axis) of a whole CMP at each time frame *t* (in X axis) under a "clean" power configuration (not involved in the training process of the model). The blue line is the actual temperature obtained via thermal simulation, whereas the red line is the predicted temperature. Although the prediction indeed introduces some errors up to 1.2˚C, it is clear that the general trend of peak temperature changes is

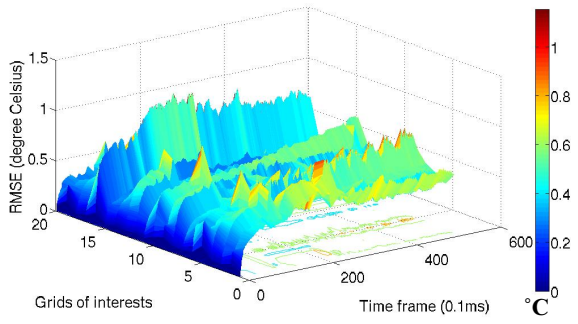


Figure 6: Accuracy of forward prediction.

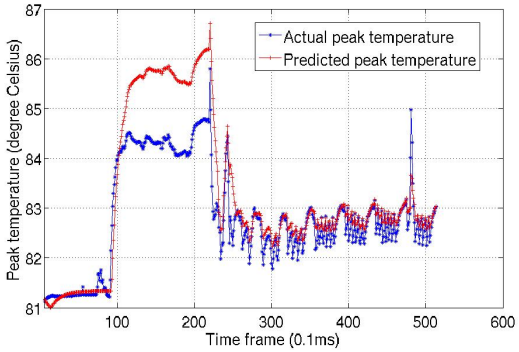


Figure 7: Peak temperature prediction.

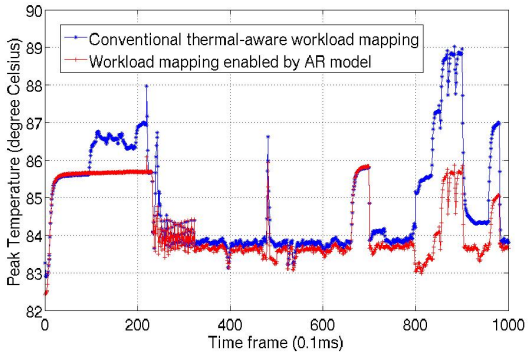


Figure 8: Thermal optimization by workload mapping.

captured very well by the proposed framework.

For execution time, Hotspot [8] needs approximately 291 seconds of CPU time to finish the transient analysis for one power configuration with other settings described in Section V. Once the AR model is trained, only 2.57 seconds are needed by using the forward prediction, and therefore a 113X speed-up is achieved. All these results demonstrate that the proposed forward prediction is accurate and stable.

B. CASE STUDY: THERMAL-AWARE WORKLOAD MAPPING

To demonstrate the effectiveness of thermal prediction, we present an application of thermal optimization based on the proposed framework. The experimental setup is described as follows. According to [18], we separate SPECcpu2000 benchmarks into two categories: intermediate and intensive thermal demands, and then randomly select eight benchmarks from each category to form a representative multi-program workload for a 16-core CMP. The remaining parameters are the same as in Section V. The goal is to find a static workload mapping that leads to the lowest peak temperature.

Similar to the thermal-aware mapping proposed by [6], we map eight thermal-intensive applications to the corners and edges of a CMP and intermediate ones to the center. Next, we exhaustively swap the four applications in the corners and the four in the centers to search for the "coolest" mapping. As a result, a total of 4!×4! = 576 swapping and thermal evaluations are required. Note that using conventional thermal simulations will take 90.1 hours; with our model, this can be done within 49 minutes. Figure 8 shows the results of workload mapping. The X axis represents the time while the Y axis stands for the peak temperature of a whole CMP. The blue line is the conventional thermal-aware mapping, whereas the red line is the mapping enabled by the proposed AR model. Compared to the conventional mapping, we further reduce the peak temperature by 3.1˚C with a different mapping order. The key of achieving this reduction is that, while the conventional strategy separates thermal-intensive applications spatially (to different corners or edges), our approach further ensures that the applications assigned to processors close to each other do not have similar timing of hotspot occurrence. To evaluate our approach with a more general scenario, the whole process described above is repeated for ten times, and on average peak temperature is reduced by 2.9˚C compared to the conventional mapping similar to [6].

VII. CONCLUSION

In this paper, we present a systematic learning framework that accurately predicts the transient temperature of a CMP by using an AR Lasso model. The proposed model achieves 113X speed-up while introducing a RMSE of only 0.8˚C. An interesting line future work is to further develop this framework to predict the transient temperature of a three-dimensional (3D) CMP.

REFERENCES

[1] D. Brooks et al., "Power, thermal, and reliability modeling in nanometer-scale microprocessors," *MICRO*, 2007.

[2] D. Brooks et al., "Dynamic thermal management for high-performance microprocessors," *HPCA*, 2001.

[3] T. Ebi et al., "TAPE: thermal-aware agent-based power economy for multi/many-core architectures," *ICCAD*, 2009.

[4] J. S. Lee et al., "Predictive Temperature-Aware DVFS," *IEEE Trans. Computers*, 2010.

[5] A. K. Coskun et al., "Proactive temperature balancing for low cost thermal management in MPSoCs," *ICCAD*, 2008.

[6] A. K. Coskun et al., "Evaluating the Impact of Job Scheduling and Power Management on Processor Lifetime for Chip Multiprocessors," *Performance*, 2009.

[7] S. Sharifi et al., "Accurate Direct and Indirect On-Chip Temperature Sensing for Efficient Dynamic Thermal Management, " *TCAD*, 2010.

[8] W. Huang et al., "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *TVLSI*, 2006.

[9] P. Li et al., "IC thermal simulation and modeling via efficient multigrid-based approaches," *TCAD*, 2006.

[10] E. Bosch, "Thermal Compact Models: An Alternative Approach", *IEEE Trans. on Components and Packaging Technologies*, 2003.

[11] A. Sridhar et al., "3D-ICE: Fast compact transient thermal modeling for 3D ICs with inter-tier liquid cooling", *ICCAD*, 2010.

[12] T. Wang et al, "3-D thermal-ADI: a linear-time chip level transient thermal simulator", *TCAD*, 2002.

[13] C. Xu et al., "Fast 3D Thermal Analysis of Complex Interconnect Structures Using Electrical Modeling and Simulation Methodologies", *ICCAD* , 2009.

[14] W. Ames, "Numerical Methods for Partial Differential Equations," *Academic Press*, 1977.

[15] M. Reed et al., "Methods of modern mathematical physics," *Academic press*, 1980.

[16] T. Cormen et al., "Introduction to Algorithms," *The MIT Press*, 2002.

[17] K. Skadron et al., "Control-Theoretic Techniques and Thermal-RC Modeling for Accurate and Localized Dynamic Thermal Management," *HPCA*, 2002.

[18] K. Skadron et al., "Temperature-aware microarchitecture: Modeling and Implementation," *TACO*, 2004.

[19] R. E. Kessler, "The alpha 21264 microprocessor," *MICRO*, pp. 24-36, 1999.

[20] Y. Liu et al., "Accurate temperature-dependent integrated circuit leakage power estimation is easy," *DATE*, pp. 1526-1531, 2007.

[21] D.-C. Juan et al., "Statistical thermal evaluation and mitigation techniques for 3D chip-multiprocessors in the presence of process variations," *DATE*, 2011.

[22] T. Hastie et al., "The Elements of Statistical Learning: Data Mining, Inference, and Prediction," *Springer*, 2009.

[23] G. Seber, "Multivariate Observations. Hoboken," *NJ: John Wiley & Sons, Inc.*, 1984.

[24] Matlab®, http://www.mathworks.com/products/matlab/.

[25] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. R. Statist*, 1996.

[26] A. Hoerl et al., "Ridge regression: Biased estimation for nonorthogonal problems, " *JSTOR*, 1970.

[27] M.Friedlander, http://www.cs.ubc.ca/~schmidtm/Software/lasso.html

[28] D. Burger D. Brooks et al., "Dynamic thermal management for high-performance microprocessors," *HPCA*, 2001.

[29] T. Ebi et al., "TAPE: thermal-aware agent-based power economy for multi/many-core architectures," *ICCAD*, 2009.

[30] D. Burger et al., "The SimpleScalar tool set, version 2.0," *ACM SIGARCH Computer Architecture News*, pp.13-25, 1997.

[31] D. Brooks et al., "Wattch: a framework for architectural-level power analysis and optimizations", *ISCA*, 2000.

[32] J. A. Butt et al., "Static power model for architects," *MICRO*, 2000.

[33] S. Rusu et al., "A 65-nm dual-core multithreaded Xeon processor with 16-MB L3 Cache," *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 2007.

[34] SPEC CPU2000, http://www.spec.org/cpu2000/

[35] W. Zhao et al., "New generation of predictive technology model for sub-45nm early design exploration," *IEEE Transactions on Electron Devices*, 2006.

[36] L. Cheng et al., "Physically justifiable die-level modeling of spatial variation in view of systematic across wafer variability," *DAC*, 2009.