Chip-level Thermal Simulation for a Multicore Processor Using a Multi-Block Model Enabled by Proper Orthogonal Decomposition

Lin Jiang, Anthony Dowling, Yu Liu, and Ming-C. Cheng
Department of Electrical & Computer Engineering
Clarkson University
Potsdam, NY, USA 13699
{jiangl2, dowlinah, yuliu, mcheng }@clarkson.edu

***Abstract*—To perform chip-level thermal simulation effectively for large-scale processors with multicores/manycores, a multi-block model enabled by proper orthogonal decomposition (POD) and domain decomposition is applied. This approach partitions a large-scale processor into smaller building blocks, such as cores, caches, I/O units, etc. For each building block, a set of temperature solution data accounting for parametric variations of interest is collected individually from FEniCS, a finite element simulation platform, to extract its basis functions (or POD modes). Using smaller building blocks, the multi-block approach significantly enhances the computational efficiency of POD mode generation to construct a POD model for the entire chip. In this work, a set of POD modes is trained by the solution data from each of two selected building blocks, a core and a level-2 cache, of AMD Athlon II X4 610e, a quad-core chip. A two-block POD thermal model is developed for Core 1 and L2 Cache by projecting these two blocks to a functional space represented by these 2 sets of POD modes. The discontinuous Galerkin method with the penalty number is applied to ensure the boundary continuity at the block interface. An optimal range of the penalty number for the two-block POD thermal model has been observed to provide an accurate prediction of the dynamic thermal distribution in Core 1 and L2 Cache. For the two-block POD model, a least square error below 3% is achieved with only 3 POD modes in each block. This results in a reduction in the numerical degrees of freedom by almost 4 orders in magnitude and thousands of times faster than FEniCS for the thermal simulation.**

Keywords—Multicore CPUs, thermal simulation, proper orthogonal decomposition, hot spots, reduced order model.

I. INTRODUCTION

Artificial intelligence (AI) and machine learning (ML) have been widely used in most of domains of technology [1], [2]. The models used in AI and ML are trained by processing millions of crawled data giving rise to considerable demand for high-performance processors [1] To satisfy the need, more cores are integrated on a semiconductor chip, and the density of transistors and power dissipation have been increasing dramatically in recent years, which has led to high temperature and hot-spot generation due to severe joule heating. High temperature and hot spots contribute to not only degradation of performance but also deterioration of reliability [3],[4]. To reduce temperature and suppress hot spots in high-performance processors, the general practice is to apply effective thermal-aware task scheduling and thermal management, which however requires effective and accurate chip-level thermal-simulation techniques.

Several approaches have been developed for the thermal simulation of semiconductor chips; each of them offers a different level of efficiency and accuracy. Among these approaches, direct numerical simulations (DNSs) based on either the finite element method (FEM) or finite difference method (FDM) provide accurate and detailed thermal analysis at the expense of a large number of degrees of freedom (DoF). Many open-source or commercial DNS tools are available for such applications, for example, FEniCS [5], ANSYS [6], COMSOL [7], etc. These DNSs, although offering accurate thermal solution with fine resolution, demand extensive computational resources and are impractical for chip-level thermal simulations.

To conduct the chip-level thermal simulation efficiently, the lumped RC thermal circuit model has been used to predict the thermal profile in large-scale semiconductor chips; for example, the block model of HotSpot [8]–[10] is one of most popular thermal simulators using the compact RC thermal model for chip-level thermal simulations. Due to the large RC lumped element, the RC thermal circuit model is not able to capture the small-size hot spots in semiconductor chips but only offers average temperatures for the large RC elements. With the approximation associated with large lumped element, heat flux at the element interfaces cannot be estimated accurately. The accuracy of the block model of HotSpot has been challenged due to the inaccurate thermal prediction for some floorplans, compared to DNS [11]. To improve the accuracy of the block model of HotSpot, the grid model of HotSpot [12] was developed, where smaller elements are allowed to provide a more detailed/accurate temperature prediction. However, when using very small elements for better accuracy, the grid model of HotSpot is equivalent to the FDM and becomes prohibitive for chip-level simulation.

To enhance the efficiency of chip-level thermal simulations, another strategy is to develop a spatial impulse response (or the Green's function)[13]–[15] of the selected chip. The Green's function is usually pre-trained by the thermal solution derived

from DNS in response to a unit point heat source at the center of the chip. The spatial temperature solution is then obtained by a convolution of the pre-trained Green's function with the power profile. However, for the Green's function method, it is difficult to apply boundary conditions (BCs) [13], [14] or to perform transient thermal simulation [13], [15]. In addition, the training of the Green's function using DNS of the entire chip is extremely time consuming [13], especially if a high resolution is needed to capture the localized hot spots. As the technology node is further reduced and more cores are integrated on a chip, the computation of the Green's function is becoming more intensive and impractical for developing such a thermal model for the entire chip, especially when a high resolution is needed.

An alternative is to use a reduced-order simulation model enabled by a data-driven approach based on proper orthogonal decomposition (POD) [16], [17]. This approach projects a dynamic thermal problem from the physical domain onto a functional space (POD space) described by a finite set of basis functions (also called POD modes). To derive an optimal set of modes, dynamic thermal data accounting for parametric variations of interest, such as variations of heat excitations and BCs, are obtained from DNSs to train the POD modes. The POD model constructed by these trained robust modes is therefore able to respond accurately to the parametric variations within or near the training conditions with a very small number of DoF. In addition to the high accuracy and efficiency, the POD model also offers the temperature profile as detailed as DNS.

The POD simulation approach has been shown to be effective in many areas of research [18]–[27]including thermal simulations of integrated circuits and CPUs [20]–[22], [27]. However, similar to the problem encountered in the pre-training of the Green's function, a long simulation time and massive thermal data needed to train the POD modes become prohibitive for larger chips with high resolutions. To overcome the difficulty, the multi-block POD methodology is proposed for large-scale chips, such as multicore/manycore processors. In the multi-block POD model, the domain decomposition technique is implemented to partition a large semiconductor chip into smaller building blocks, such as cores, caches, I/O units, etc. For each small block, a set of POD modes and the model parameters can be generated more efficiently and stored into a technology library. The POD model for the entire chip can then be constructed by gluing these POD blocks with the discontinuous Galerkin (DG) method [28], [29]. This method is applied to stabilize the numerical solution at the interface by enforcing the heat flux continuity but allowing a small temperature discontinuity (i.e., the weak boundary condition) in an average sense at the interface between any 2 neighboring blocks. With the multi-block POD model for a large chip partitioned into a large number of building blocks, parallel computing can also be implemented in POD mode generation and thermal simulation to further enhance the computational efficiency.

Continuing a previous study [27], this work investigates a two-block POD model that projects two building blocks (Core 1 and its adjacent L2 cache) in AMD Athlon II X4 610e [30] to a POD space described by the 2-block POD modes. For each building block, DNSs are performed in FEniCS [5], an open-source FEM platform, to collect temperature data for the extraction of POD modes. The two-block POD model is demonstrated and verified against the DNS, and it has shown that the POD results are in very good agreement with the DNS with almost 4 orders reduction in the DoF.

II. Thermal Simulation Methodology based on POD

A. Single-block model

Using the POD method, the physical domain is projected onto a mathematical space represented by a finite number of POD modes. Temperature in space and time $T(\vec{r},t)$ can then be represented by a linear combination of the selected POD modes φ_i as

$$T(\vec{r},t) = \sum_{i=1}^{M} a_i(t)\,\varphi_i(\vec{r}), \tag{1}$$

where φ_i is the *i*-th POD mode, *M* is the number of selected POD modes which determines the accuracy and efficiency of the POD approach and $a_i(t)$ is the time-dependent coefficient of the *i*-th POD mode.

To obtain an optimal set of the POD modes, each POD mode is obtained by maximizing the mean square inner product of the thermal solution with the modes via the following equation

$$\frac{\langle\left(\int_\Omega T(\vec{r},t)\varphi d\Omega\right)^2\rangle}{\int_\Omega \varphi^2\, d\Omega}, \tag{2}$$

where Ω is the physical domain of the selected structure and the brackets $\langle\,\rangle$ denote the average over the collected thermal solution data. For dynamic thermal simulation, the average is computed over temporal samples (snapshots) obtained from DNSs. The maximization process in (2) gives rise to a Fredholm equation shown below for the POD modes,

$$\int_{\vec{r}'} \mathbf{R}(\vec{r},\vec{r}')\,\vec{\varphi}(\vec{r}')d\vec{r}' = \lambda\vec{\varphi}(\vec{r}), \tag{3}$$

where $\mathbf{R}(\vec{r},\vec{r}')$ is a two-point correlation tensor expressed as

$$\mathbf{R}(\vec{r},\vec{r}') = \langle T(\vec{r},t)\otimes T(\vec{r}',t)\rangle. \tag{4}$$

With the temperature data $T(\vec{r},t)$ of the simulation domain collected from DNSs, the method of snapshots [25], [26] is applied to solve the eigenvalue problem in (3) for the eigenvalues λ_i and POD modes φ_i.

With the generated POD modes, the heat conduction equation can be projected onto a POD space represented by the POD modes using the Galerkin projection,

$$\int_\Omega \left(\varphi_i(\vec{r})\frac{\partial \rho C T}{\partial t} + \nabla\varphi_i(\vec{r})\cdot k\nabla \mathrm{T}\right) d\Omega$$
$$= \int_\Omega \varphi_i(\vec{r})\cdot P_d(\vec{r},t)d\Omega - \int_S \varphi_i(\vec{r})(-k\nabla \mathrm{T}\cdot\vec{n})dS, \tag{5}$$

where k is thermal conductivity, ρ is the density, C is the specific heat, $P_d(\vec{r},t)$ is the power density, S is the boundary

surface of the selected domain and $\vec{n}$ is the outward normal vector of boundary surface. Substituting (1) into (5), it leads to an M-dimensional ordinary differential equation (ODE) for $a_i(t)$,

$$\sum_{j=1}^{M} c_{i,j}\frac{da_j}{dt} + \sum_{j=1}^{M} g_{i,j}\,a_j = P_i\,, \quad i = 1 \text{ to } M, \tag{6}$$

where P_i representing the last 2 terms of (5) for the i-th mode is the power density dissipated in the POD space and can be pre-evaluated since the shape of power density is predefined, and $c_{i,j}$ and $g_{i,j}$ are the elements of thermal capacitance and thermal conductance matrices in the POD space and defined as

$$c_{i,j} = \int_{\Omega} \rho C \vec{\varphi}_i \vec{\varphi}_j d\Omega\,, g_{i,j} = \int_{\Omega} k\nabla\vec{\varphi}_i \nabla\vec{\varphi}_j d\Omega. \tag{7}$$

Once a_j is determined from (6), the temperature solution can be evaluated from (1).

As presented above, the POD model development consists of thermal data collection from DNS, calculations of POD modes and eigenvalues from (3) using the snapshot method, and evaluations of model parameters in (7). This *training* process could be computationally intensive for a large simulation domain with a high resolution. To minimize the computational resources in the training, the large domain is partitioned into smaller building blocks, which is presented next.

B. Multi-block model

When placing block together, the last term of (5) needs to be reformulated to account for heat flux across the interface between adjacent blocks. The DG method [28], [29] is applied to properly enforce the interface thermal continuity, and (5) for the multi-block model becomes

$$\int_{\Omega}\left(\varphi_i(\vec{r})\frac{\partial \rho C T}{\partial t} + \nabla\varphi_i(\vec{r})\cdot k\nabla T\right)d\Omega + k\int_{S}\mu[\![T]\!]\cdot[\![\varphi_i]\!]dS - k\int_{S}([\![T]\!]\cdot\{\nabla\varphi_i\} + \{\nabla T\}\cdot[\![\varphi_i]\!])\cdot d\vec{S} = \int_{\Omega}\varphi_i(\vec{r})\cdot P_d(\vec{r},t)d\Omega\,, \tag{8}$$

where $[\![*]\!]$ and $\{*\}$ indicate difference and average across interface, respectively, and μ is the penalty constant defined as N_μ/dr (dr is the size of the local element with N_μ as the penalty number). S is the interface surface between two adjacent blocks. N_μ can be adjusted to balance discontinuities between temperature and heat flux at the interface to minimize the least square (LS) error and to stabilize the numerical solution.

For a two-block POD model including the heat flux exchanges via the interface, the matrix equation for both POD blocks becomes

$$\begin{bmatrix}\boldsymbol{C}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{C}_2\end{bmatrix}\frac{d}{dt}\begin{bmatrix}\vec{a}_1(t)\\ \vec{a}_2(t)\end{bmatrix} + \begin{bmatrix}\boldsymbol{G}_1 + \boldsymbol{G}_{1,B_{1,2}} & \boldsymbol{G}_{1,2}\\ \boldsymbol{G}_{2,1} & \boldsymbol{G}_2 + \boldsymbol{G}_{2,B_{2,1}}\end{bmatrix}\begin{bmatrix}\vec{a}_1(t)\\ \vec{a}_2(t)\end{bmatrix} = \begin{bmatrix}\vec{P}_1\\ \vec{P}_2\end{bmatrix}, \tag{9}$$

where $B_{p,q}$ indicates the interface between p-th and q-th blocks. $\boldsymbol{C}_p$ and $\boldsymbol{G}_p$ are the thermal capacitance and thermal conductance matrices of p-th blocks and their elements are given by (7). Compared to the thermal conductance matrices of single-block model, an extra thermal conductance matrix, $\boldsymbol{G}_{p,B_{p,q}}$, is included for p-th block to consider the effect of q-th block with respect to temperature and heat flux at the interface $B_{p,q}$ and is given as

$$\boldsymbol{G}_{p,B_{p,q}} = \begin{bmatrix} g_{p,B_{p,q},1,1} & g_{p,B_{p,q},1,2} & \cdots & g_{p,B_{p,q},1,M_p}\\ g_{p,B_{p,q},2,1} & g_{p,B_{p,q},2,2} & \cdots & g_{p,B_{p,q},2,M_p}\\ \vdots & \vdots & \ddots & \vdots\\ g_{p,B_{p,q},M_p,1} & g_{p,B_{p,q},M_p,2} & \cdots & g_{p,B_{p,q},M_p,M_p}\end{bmatrix}, \tag{10}$$

where M_p is the number of selected POD mode of p −th block, and the element of $\boldsymbol{G}_{p,B_{p,q}}$ is given by

$$g_{p,B_{p,q},i,j} = -k\int_{S}\left(\frac{1}{2}\varphi_{p,j}\cdot\nabla\varphi_{p,i} + \frac{1}{2}\varphi_{p,i}\cdot\nabla\varphi_{p,j}\right)d\vec{S} + k\int_{S}\mu\varphi_{p,i}\varphi_{p,j}dS. \tag{11}$$

In the matrix equation (9), the p-th block is coupled with its adjacent q-th block via $\boldsymbol{G}_{p,q}$. If p-th block is not directly adjacent with q-th block, $\boldsymbol{G}_{p,q} = \boldsymbol{0}$. Otherwise, it is given as

$$\boldsymbol{G}_{p,q} = \begin{bmatrix} g_{p,q,1,1} & g_{p,q,1,2} & \cdots & g_{p,q,1,M_p}\\ g_{p,q,2,1} & g_{p,q,2,2} & \cdots & g_{p,q,2,M_p}\\ \vdots & \vdots & \ddots & \vdots\\ g_{p,q,M_q,1} & g_{p,q,M_q,2} & \cdots & g_{p,q,M_q,M_p}\end{bmatrix}, \tag{12}$$

where M_q is the number of selected modes of q-th block and $g_{p,q,i,j}$ is given by

$$g_{p,q,i,j} = -k\int_{S}\left(-\frac{1}{2}\varphi_{q,j}\cdot\nabla\varphi_{p,i} + \frac{1}{2}\varphi_{p,i}\cdot\nabla\varphi_{q,j}\right)d\vec{S} - k\int_{S}\mu\varphi_{p,i}\varphi_{q,j}dS. \tag{13}$$

III. CHIP-LEVEL THERMAL SIMULATION USING MULTI-BLOCK POD MODEL

The AMD ATHLON II X4 610e processor is selected in this investigation, which consists of four cores, two L2 caches, one northbridge, three I/O units and one DDR3 module, as shown in Fig. 1. The dimension of the quad-core chip is 14mm × 12mm × 242μm (length×width×thickness) and the material property is listed in Table I. In the single-block model, the thermal simulation is performed via DNSs over the entire chip to collect temperature data and it is, as discussed above, computationally intensive for a large chip with a high resolution. In the multi-

block model, the temperature data is, however, independently collected for each building block in the entire domain. The dynamic power map is applied to the top layer (named the device layer hereafter) of the chip with the device layer thickness of 55.8 μm. For data collection of each building block, the thermal simulation is performed over the simulation domain that consists of the green blocks and the building block shown in Fig. 1. The simulation domain for data collection of each embedded building block is shown in Fig. 2. In such a setting, solution data collected from each building block is able to account for the variation of the block BCs induced by the power excitations outside the block.

TABLE I. TEMPERATURE INDEPENDENT MATERIAL PROPERTY.

Specific heat, C	Density, ρ	Thermal conductivity, k
751.1 (J/(kg·K))	2330 (kg/m^3)	100 (W/(m·K))

All outer surfaces of the simulation domain for data collection are assumed adiabatic except for the bottom where the convection BC is implemented with a constant heat transfer coefficient and an ambient temperature T_{amb} of 45°C. The outer boundary surface of the simulation domain is 3 mm from the building block. The dynamic power density in each building block is randomly generated in time; in each time step, it represents an averaged power density over 48k CPU cycles at 3.5 GHz with a total power approximately equal to 8.9 W for L2 cache and 16 W for Core 1. In the surrounding of each building block, the dynamic power density is generated with a different random sequence, which offers the variation of the interface flux on each side of the building and allows the POD modes to adapt the realistic BC variation to construct a more effective POD model.

In this work, collection of thermal data in the CPU domain for training POD modes and calculations of POD model coefficients in (10)-(13) are carried out in FEniCS-FEM. The solution of the ODE matrix equation in (9) and post-processing calculations for the predicted temperature in (1) are performed in C++ using solvers in the PETSc library.

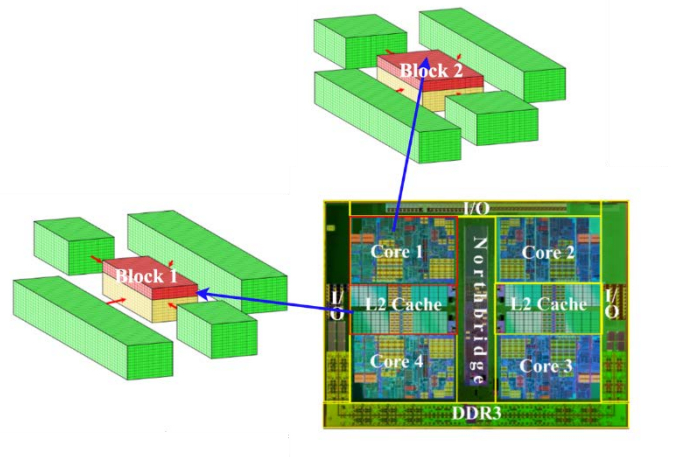

Fig. 1. Floorplan of the AMD ATHLON II X4 610e CPU and schematic of the simulation domains for the building blocks.

A. *Temperature data collection*

In this work, L2 Cache and Core 1 of the quad-core chip are selected as Block 1 and Block 2, respectively, as shown in Fig. 1 and 2. The dynamic thermal simulation is performed for each of Block 1 and Block 2 in FEniCS-FEM independently to collect dynamic temperature data of each building block to generate its eigenvalues and POD modes. The eigenvalue represents the mean squared temperature variation captured by the corresponding POD mode, and therefore its spectrum reveals the information on the number of POD modes needed to offer accurate temperature solution. The eigenvalue spectrums of two building blocks are shown in Fig. 3. For both Block 1 and Block 2, a reduction in the eigenvalue by two orders of magnitude is observed from the first to the second mode and a decrease by four orders from the first to the third mode. Based on the rapid reduction in the eigenvalue for the first few modes, it is expected that the two-block POD model with a small number of modes is able to offer an accurate prediction of dynamic temperature solution. However, the expectation can be achieved only if the quality of the data collected from the DNSs is reasonably good.

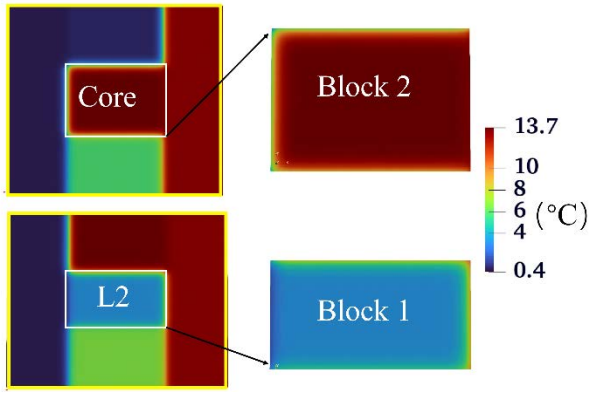

Fig. 2. Temperature contours (relative to the ambient temperature 45 °C) in the simulation domain.

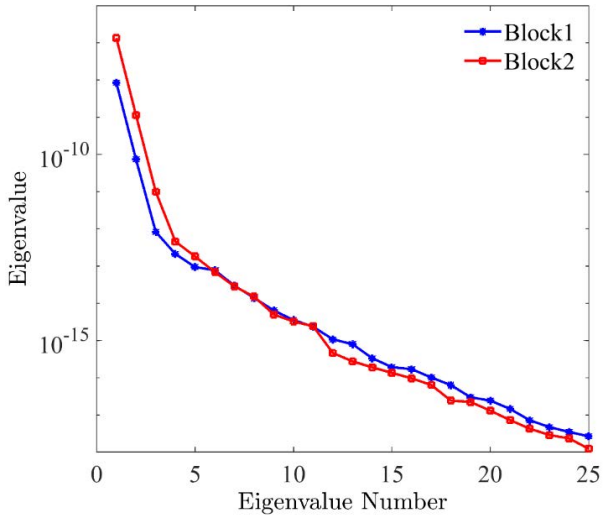

Fig. 3. Eigenvalue spectrum of the thermal data collected from the selected building blocks.

B. *Verification of the multi-block model*

To demonstrate the validity of the two-block POD model, thermal simulation based on (9) for the domain consisting of Core 1 (Block 2) and its adjacent L2 Cache (Block 1), shown in Fig. 1, is performed. The dynamic power density applied to each of the 2 blocks is generated using a random sequence different from those used in the POD mode training. The adiabatic BCs are applied to all surfaces except for the bottom of the chip where a constant heat transfer coefficient is implemented with an ambient of 45ºC. Thermal simulation is also performed via FEniCS-FEM with identical settings including heat sources and BCs to validate the accuracy of the two-block POD model. In this demonstration, a same number of POD modes is used in both blocks.

The LS error estimated from the equation below for the two-block POD model is a function of the number of POD modes.

$$err_{LS} = \sqrt{\frac{\sum_{i=1}^{N_t} \int_{\Omega} e_i^2(\vec{r}) d\Omega}{\sum_{i=1}^{N_t} \int_{\Omega} (T_i(\vec{r}) - T_{amb})^2 d\Omega}}, \qquad (14)$$

where the index i denotes the time step (snapshot), and $T_i(\vec{r})$ and $e_i(\vec{r})$ are the temperature solution from FEniCS-FEM and the temperature difference between FEniCS-FEM and the POD model, respectively. For the two-block POD model, the DG method with an adjustable penalty number N_μ [28], [29] is used to enforce the thermal continuity across the interface between Core 1 and L2 Cache. The effect of penalty number on the LS error is shown in Fig. 4 for the two-block POD model. It is observed that, when using 2 or more modes in the POD model, the LS error reaches a minimum value with N_μ near 7. The LS error vs. the number of modes with $N_\mu = 7$ is thus plotted in Fig. 5. It is interesting to observe in Figs. 4 and 5 that the 2-block POD model with 3 modes actually offers a better accuracy than that with 4 – 7 modes. With $N_\mu = 7$, an LS error as small as 2.9% can be reached and it fluctuates around 3% - 3.1% beyond 3 modes. Based on Fig. 4 when using the 3-mode POD model, the penalty number should be within $4 \leq N_\mu \leq 10$ to reach an LS error below 3%.

The POD simulation demonstrated above reveals a 4-order reduction in the numerical DoF, compared to FEniCS-FEM, which results in a significant saving in computing time. The POD simulation includes solving the ODE in (6) and the post processing calculation using (1) to recover the temperature solution. The computational time of thermal simulation for the selected two blocks using FEniCS-FEM and the two-block POD model is shown in Table II, where Post1 and Post2 denote the post-processing calculations of temperature in the entire domain and device layer, respectively. As shown in Table II, thermal simulation based on the two-block POD model with 3 modes is 1959 times faster than FEniCS-FEM. Practically, only the temperature in the device layer is required, which would offer a speedup of 3918 times, compared to FEniCS-FEM.

Based on the results presented in Fig. 4, the optimal penalty number is $N\mu = 7$, and thus the detailed comparison of the dynamic thermal distributions obtained from FEniCS-FEM and the two-block POD model is given below with $N\mu = 7$. As expected according to the eigenvalue spectrum shown in Fig. 3 and the LS error in Fig. 5, the temperature evolution predicted by the two-block POD model with just 3 POD modes is in very good agreement with that obtained from FEniCS-FEM over the entire simulation time. The temperature evolution in time at the center of L2 Cache is given in Fig. 6. Similarly, the dynamic temperature at the center of Core 1 is illustrated in Fig. 7, where the temperature solutions obtained from the two-block POD model with 3 or 5 POD modes and FEniCS-FEM almost overlap each other.

TABLE II. CONSUMPTIONAL TIME OF THERMAL SIMULATION FOR THE MULTI-BLOCK POD AND FENICS-FEM METHODS.

Two-block POD model (s)	**Number of modes**	**1**	**2**	**3**	**4**	**5**
	ODE	0.023	0.019	0.017	0.019	0.019
	Post1	0.015	0.025	0.041	0.053	0.071
	Post2	0.004	0.007	0.012	0.015	0.019
FEniCS (s)	113.623					

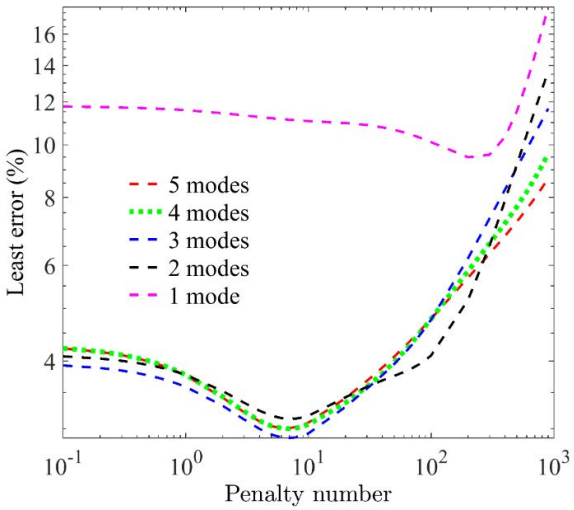

Fig. 4. LS error influenced by the penalty number.

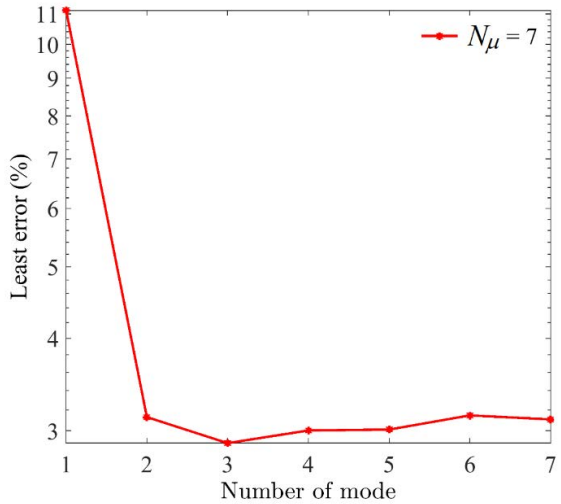

Fig. 5. LS error of the two-block POD model over the entire simulation time and domain.

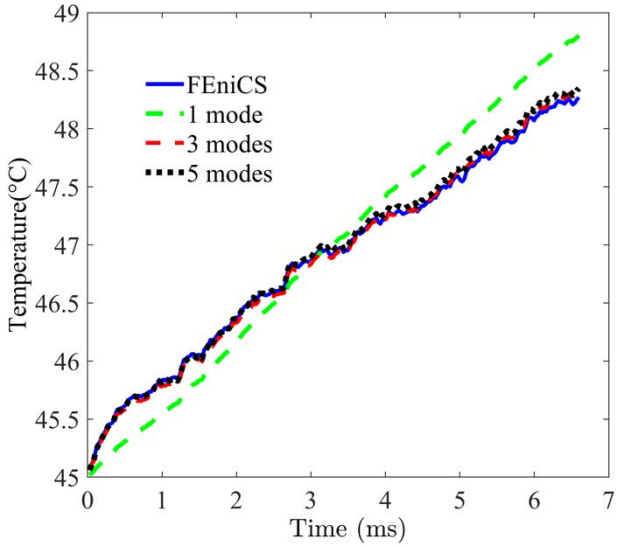

Fig. 6. Temperature evolution in time at the center of L2 Cache. In the POD model, $N_\mu = 7$.

The temperature distribution at $t = 6.6$ ms from L2 Cache to Core 1 along the centers of these 2 blocks across the interface is illustrated in Fig. 8. The temperature profile provided by the two-block POD model with 3 POD modes agrees very well with the temperature profile from FEniCS-FEM, which is consistent with the information indicated by the eigenvalue spectrum in Fig. 3 and the LS error in Fig. 5. Compared with the FEniCS-FEM results in the centers of L2 Cache and Core 1, approximately 2.6% and 1.0% (or 0.09 °C and 0.14 °C) differences, respectively, are achieved when using the two-block POD model with 3 POD modes.

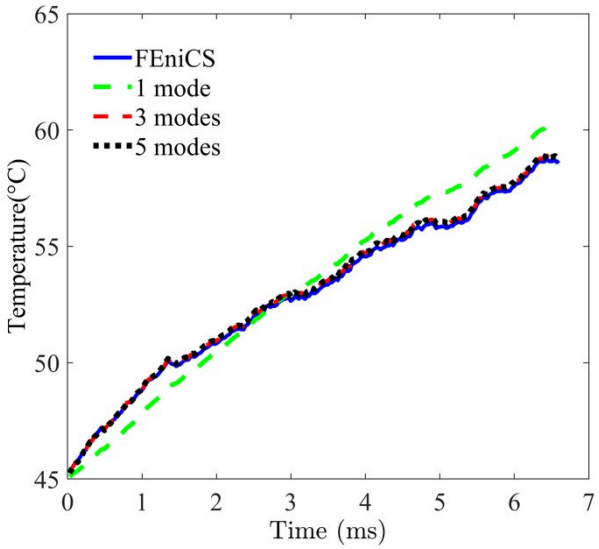

Fig. 7. Temperature evolution in time at the center of Core 1. In the POD model, $N_\mu = 7$.

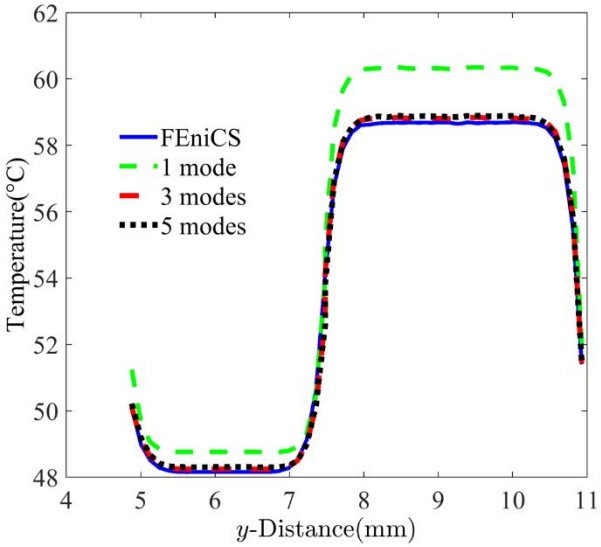

Fig. 8. Temperature distribution at 6.6ms from L2 Cache to Core 1 along the centers of these 2 blocks. In the POD model, $N_\mu = 7$.

IV. Conclusion

A multi-block thermal simulation methodology enabled by the data-driven POD model and domain decomposition has been investigated. The approach has been applied to develop a two-block POD model for thermal simulation of 2 selected blocks, including Core 1 and L2 Cache, from a quad-core chip, AMD ATHLON II X4 610e. This study has shown that an appropriate penalty number N_μ is needed in the 2-block POD model to minimize the interface discontinuity for an optimal prediction of the dynamic thermal distribution in the 2-block domain. It is found that an LS error below 3% can be achieved for the 2-block POD model with 3 modes in each block if $4 \leq N_\mu \leq 10$. This results in a reduction in the numerical DoF by nearly 4 orders of magnitude and leads to nearly 2000 times or 4000 times of speedup for a thermal prediction of the entire 2-block domain or the device layer, respectively, compared to FEniCS-FEM.

This work initiates the development of a multi-block POD thermal simulation methodology at the chip level for an entire CPU or GPU. Such a multi-block concept offers a very efficient approach to generation of the POD modes and calculations the POD model parameters to construct a POD model for very large chips like multicore CPUs or many-core GPUs, especially when a higher resolution is needed. All CPUs and GPUs are designed and constructed based on building blocks, such as cores, caches, I/O units, memory modules, etc., in the selected AMD ATHLON processor. When developing a multi-block POD model, a useful practice would be using these standard building blocks to partition the entire processor into a multi-block domain. For a semiconductor chip consisting of a large number of POD blocks, parallel computing can also be applied to further improve the POD simulation efficiency.

Acknowledgment: This work is supported by National Science Foundation under Grant No. ECCS-2003307

References

[1] J. Lee, S. Kang, J. Lee, D. Shin, D. Han, and H. J. Yoo, "The Hardware and Algorithm Co-Design for Energy-Efficient DNN Processor on Edge/Mobile Devices," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 67, no. 10, pp. 3458–3470, Oct. 2020.

[2] S. Sadiqbatcha, Y. Zhao, J. Zhang, H. Amrouch, J. Henkel, and S. X. D. Tan, "Machine learning based online full-chip heatmap estimation," in *Proc. Asia/South Pacific Design Autom. Conf.*, 2020, pp. 229–234.

[3] X. Huang, A. Kteyan, S. X. D. Tan, and V. Sukharev, "Physics-Based Electromigration Models and Full-Chip Assessment for Power Grid Networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 11, pp. 1848–1861, Nov. 2016.

[4] Y. Liu, M. Li, M. Jiang, D. W. Kim, S. Gu, and K. N. Tu, "Joule Heating Enhanced Electromigration Failure in Redistribution Layer in 2.5D IC," in *Proc. Electronic Comp. Technol. Conf.*, 2016, vol. 2016-August, pp. 1359–1363.

[5] "Fenics project," *https://fenicsproject.org/*, Dec. 13, 2021.

[6] G. Xu, "Thermal modeling of multi-core processors," in *Proc. ITHERM*, 2006, pp. 96–100.

[7] K. R. Vaddina, A. M. Rahmani, K. Latif, P. Liljeberg, and J. Plosila, "Thermal modeling and analysis of advanced 3D stacked structures," in *Procedia Engineering*, 2012, vol. 30, pp. 248–257.

[8] "HotSpot 6.0 Temperature Modeling Tool," *http://lava.cs.virginia.edu/HotSpot/*, Dec. 13, 2021.

[9] K. Sankaranarayanan, S. Velusamy, M. Stan and K. Skadron, "A case for thermal-aware floorplanning at the microarchitectural level", *J. Instruction-Level Parallelism*, vol. 7, Oct. 2005.

[10] K. Skadrony, M. Stanz, M. Barcellaz, A. Dwarkaz, W. Huangz, Y. Liy, et al., "HotSpot: Techniques for modeling thermal effects at the processor-architecture level", in *Proc. Int. Workshop THERMal Investigations ICs Syst*., 2002.

[11] D. Fetis and P. Michaud, "An Evaluation of HotSpot-3.0 Block-Based Temperature Model", in *Proc. WDDD*, 2006-June.

[12] W. Huang, K. Sankaranarayanan, R. J. Ribando, M. R. Stan and K. Skadron, "An improved block-based thermal model in hotspot 4.0 with granularity considerations", in *Proc. WDDD*, 2007.

[13] H. Sultan, A. Chauhan, and S. R. Sarangi, "A survey of chip-level thermal simulators," *ACM Comput. Surv.*, vol. 52, no. 2, pp. 1–35, May 2019.

[14] Y. Zhan and S. S. Sapatnekar, "High-efficiency green function-based thermal simulation algorithms," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661–1675, Sep. 2007.

[15] S. Varshney, H. Sultan, P. J. Qualcomm, and S. R. Sarangi, "NanoTherm: An Analytical Fourier-Boltzmann Framework for Full Chip Thermal Simulations," in *Proc.ICCAD*, 2019, pp. 1–8.

[16] J. L. Lumley, "The structure of inhomogeneous turbulent flows," *Atmospheric turbulence and radio wave propagation*, 1967.

[17] G. Berkooz, P. Holmes, and J. L. Lumley, "The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics*, vol. 25, no. 1, pp. 539–575, 1993.

[18] M.-C. Cheng, "A quantum element reduced order model," in *Proc. SISPAD*, 2019, pp. 1–4.

[19] M.-C. Cheng, "Quantum element method for quantum eigenvalue problems derived from projection-based model order reduction," *AIP Advances*, vol. 10, no. 11, p. 115305, 2020.

[20] R. Venters, Brian T. Helenbrook, Kun Zhang, and Ming-C. Cheng, "Proper-orthogonal-decomposition based thermal modeling of semiconductor structures," *IEEE Trans. Electron Devices*, vol. 59, no. 11, pp. 2924–2931, 2012.

[21] W. Jia, Brian T. Helenbrook, and Ming-Cheng Cheng, "Thermal modeling of multi-fin field effect transistor structure using proper orthogonal decomposition," *IEEE Trans. Electron Devices*, vol. 61, no. 8, pp. 2752–2759, 2014.

[22] W. Jia, Brian T. Helenbrook, and Ming-C. Cheng., "Fast thermal simulation of FinFET circuits based on a multiblock reduced-order model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 7, pp. 1114–1124, 2016.

[23] P. Jacquier, A. Abdedou, V. Delmas, and A. Soulaïmani., "Non-intrusive reduced-order modeling using uncertainty-aware Deep Neural Networks and Proper Orthogonal Decomposition: Application to flood modeling," *Journal of Computational Physics*, vol. 424, p. 109854, 2021.

[24] S. Fresca and A. Manzoni, "POD-DL-ROM: enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition.," *Computer Methods in Applied Mechanics and Engineering*, vol. 388, p. 114181, 2022.

[25] T. G. Ritto, F. S. Buezas, and R. Sampaio, "Proper Orthogonal Decomposition for Model Reduction of a Vibroimpact System," *J. Brazil. Soc. Mech. Sci. Eng.*, vol. 34, no. 3, pp. 330–340, 2012.

[26] Z. Wang, B. McBee, and T. Iliescu, "Approximate partitioned method of snapshots for POD," *J. Comput. Appl. Math.,* vol. 307, pp. 374–384, Dec. 2016.

[27] L. Jiang, Y. Liu, and M.-C. Cheng, "An Effective and Accurate Data-Driven Approach for Thermal Simulation of CPUs," in *Proc. ITHERM*, 2021, pp. 1008–1014.

[28] D. N. Arnold, F. Brezzi, B. Cockburn, L. Donatella Marini, and S. J. Numer Anal, "Unified analysis of discontinuous Galerkin methods for elliptic problems," *SIAM J. Numer. Anal.*, vol. 39, no. 5, pp. 1749–1779, 2002.

[29] D. N. Arnold, F. Brezzi, B. Cockburn, and D. Marini, *Discontinuous Galerkin Methods for Elliptic Problems*, vol. 11. Berlin, Germany: Springer, 2000.

[30] K. Dev, A. N. Nowroz, and S. Reda, "Power mapping and modeling of multi-core processors," in *Proc. ISLPED*, 2013, pp. 39–44.