# Distributed Model Predictive Control for Dynamic Thermal Management of Multi-Core Systems

Lei Xu*, Hai Wang*, Sheldon X.-D. Tan†, Chi Zhang*, Yuan Yuan‡, Keheng Huang§, and Zhenghong Zhang§
*School of Microelectronics & Solid-State Electronics,
University of Electronic Science & Technology of China, Chengdu, 610054 China
†Department of Electrical Engineering, University of California, Riverside, CA 92521 USA
‡School of Automation Engineering, University of Electronic Science & Technology of China, Chengdu, 610054 China
§Southwest China Research Institute of Electronic Equipment, Chengdu, 610036 China

***Abstract*—Model predictive control (MPC) based dynamic thermal management (DTM) is effective in managing temperature and power of multi-core systems at runtime. However, due to its centralized structure, such method has poor scalability. In this work, we propose a distributed MPC based DTM method, by dividing the full chip thermal model into submodels and decoupling the interactions among submodels with novel techniques. Experiments show that the proposed method performs well in thermal management with fully distributed structure.**

## I. Introduction

As technology advances, high power density of the chip leads to serious thermal related problems which harm the reliability and performance of the system. Many dynamic thermal management (DTM) methods have been used to alleviate such problem [1]. DTM methods regulate the temperature by changing the power values and/or power distribution of the chip through methods like Dynamic Voltage Frequency Scaling (DVFS) and task migration methods. Knowing how should power be adjust at current time is the core of a DTM algorithm. Simple DTM methods, which make ad hoc or heuristic power adjustment decisions, lead to poor control accuracy and smoothness in temperature and sub-optimal chip performance [2]. As a result, model predictive control (MPC) is proposed to assist the power adjustment decision in DTM methods. By predicting the future temperature, MPC method is able to calculate the optimal power needed which leads to accurate and smooth temperature control [3], [4], [2], [5].

One major drawback of MPC based DTM is its inability to handle systems with large number of cores. This is because large core number leads to large thermal model, which is used in MPC for predicting the future temperature. In this work, we propose a distributed MPC method for DTM of multi-core systems to solve this scalability problem. The new method divides the thermal model of the chip into multiple sub-models. In order to consider the lateral heat exchange of adjacent cores at the sub-model boundaries, redundant overlapping areas are kept for boundaries of each submodel, and the coupling effects from the package is compensated by introducing the edge package convection resistance model. Then, distributed MPC is performed using each submodel in parallel. Experiments show good thermal management quality of the new method with distributed structure.

This work is supported in part by National Natural Science Foundation of China under grant No. 61404024, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry, in part by the Open Foundation of State Key Laboratory of Electronic Thin Films and Integrated Devices under grant No. KFJJ201409.

## II. Basics of MPC based DTM

In this section, we introduce the MPC based DTM method.

The thermal model of a system with $l$ cores can be expressed as difference equations in discrete time domain as

$$\begin{aligned} T(k+1) &= AT(k) + B_d P(k), \\ Y(k) &= LT(k), \end{aligned} \tag{1}$$

where $T(t) \in \mathbb{R}^n$ is the thermal vector representing temperatures of $n$ blocks of the chip, which includes $l$ cores (with $l < n$) and also boundary condition nodes and nodes for package components; $A \in \mathbb{R}^{n\times n}$ and $B_d \in \mathbb{R}^{n\times l}$ contains thermal dynamic information determined by the thermal conductance, thermal capacitance of the chip; $P(k) \in \mathbb{R}^l$ is the power vector with power dissipations of $l$ cores at time $k$ as *input*; $Y(k)$ is the thermal vector with temperature information of the $l$ cores as *output*; $L \in \mathbb{R}^{l\times n}$ is the output selection matrix, which selects the $l$ core temperatures from $T(k)$.

By using the system model in (1), MPC is able to calculate the power input adjustment needed in order to track a user defined output. In order to maximize the performance of the system under thermal constraint, we use highest temperatures (called *ceiling temperatures*) allowed for the $l$ cores, $Y_{max} \in \mathbb{R}^{l\times 1}$ as the user defined output. Please note that such user defined output can be easily changed according to different goals and applications. Next, we show how to obtain the power input adjustment to track the output $Y_{max}$.

Assume the ceiling temperatures of cores over several time steps into the future are given, and written in a vector form as

$$Y_{ceil} = [Y_{max}^T, Y_{max}^T, ..., Y_{max}^T]^T \in \mathbb{R}^{lN_p\times 1}.$$

$N_p$ stands for a time frame from current to the $N_p$ steps into the future, and is called the *prediction horizon*. Also, the future control trajectory at the power input side (which is actually unknown and needs to be computed) is introduced as

$$\Delta P_k = [\Delta P(k), \Delta P(k+1), \ldots, \Delta P(k+N_c-1)]^T,$$

978-1-4673-9716-6/$31.00 ©2016 IEEE

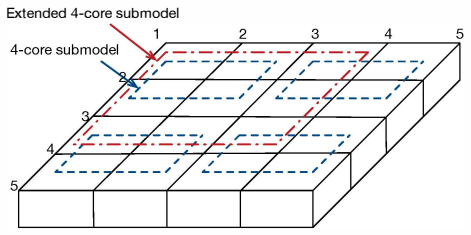



Fig. 1. A 16-core chip example showing 4-core submodel (in blue dashed frame) and the extended 4-core submodel (in red dash-dot frame). Only one extended submodel out of four is shown here for simplicity.

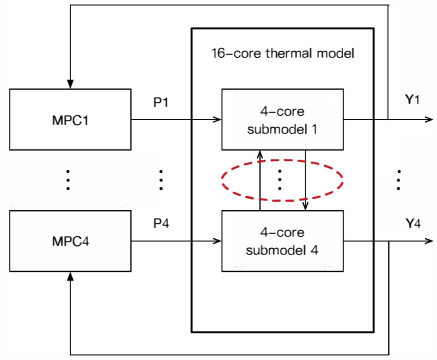



Fig. 2. Structure of the distributed MPC on the 16-core chip with 4-core model as the submodel. The interactions among submodels as in the red dashed circle need to be eliminated for the fully distributed structure.

where $N_c$ is called the *control horizon*. The prediction of core temperatures is defined as

$$Y_k = [Y(k+1|k)^T, Y(k+2|k)^T, \ldots, Y(k+N_p|k)^T]^T,$$

where $Y(k+j|k)$ is the predicted core temperatures at time $k+j$ using information of current time $k$. $Y_k$ can be calculated assuming $\Delta P_k$ is known, using

$$Y_k = V\hat{T}(k) + \Phi \Delta P_k, \tag{2}$$

where $V \in \mathbb{R}^{N_p l \times (n+l)}$ and $\Phi \in \mathbb{R}^{N_p l \times N_c l}$ are known matrices formed by thermal model matrices in (1), and their detailed structures are not given here due to page limitation.

Finally, MPC is able to calculate the optimal $\Delta P_k$ to track the user defined output as

$$\Delta P_k = (\Phi^T \Phi + R)^{-1} \Phi^T (Y_{ceil} - V\hat{T}(k)), \tag{3}$$

where $R$ is just a tuning matrix added.

At each MPC time $k$, we only use the first computed control signal $\Delta P(k)$ from (3) and update the power distribution as

$$P(k) \leftarrow P(k) + \Delta P(k), \tag{4}$$

Then, the resulting temperature $Y(k)$ would track the desired ceiling temperature with the updated power input.

## III. New Distributed MPC based DTM

One major drawback of the previously presented MPC based DTM method is that the computing time grows fast with model size. One evidence is that the sizes of $V$ and $\Phi$ matrices in (2), which are used to computing $\Delta P_k$ in (3), grow with the core number $l$ and thermal node number $n$. In order to enable parallel computing to solve the scalability problem, we propose a distributed MPC method.

The basic idea of distributed MPC is to use multiple controller instead of only one (as in the centralized case), with each controller taking care of a submodel of the original full model. One example with 16-core chip is given by dividing its 16-core thermal model into four 4-core submodel, as shown in Fig. 1 with blue dashed frame boxes. Then, each submodel is controlled by one MPC controller as shown in Fig. 2.

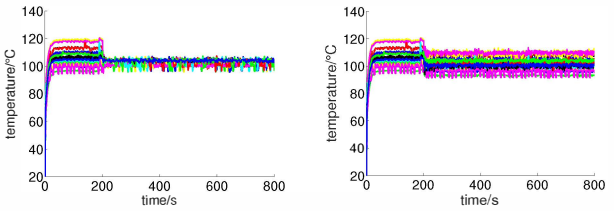
However, such simple scheme above does not work well practically. The problem comes from the interactions among the submodels. It is obvious that the 4-core submodels in the example exchange heat with each other through boundaries and through package. Such heat exchange among submodels is very hard to be considered efficiently in a distributed system. Here we introduce a solution without modelling the sophisticated heat exchange but still keeping reasonable submodel accuracy with two novel techniques working together.

First, we decouple the interaction through boundaries of submodels by introducing the *extended submodel*, which contains the original submodel together with its adjacent cores. The example of the extended 4-core submodel is shown in Fig. 1 as the red dash-dot frame box. It is noted that there are five cores in the extended model which do not belong to the original 4-core submodel, these cores are called *overlapping cores*. The overlapping cores work as the heat exchange buffer at the boundaries, and their temperatures are not directly used in MPC (i.e., they are in $T$ only, but not in $Y$ of (1)). With the overlapping cores, we simply assume the boundaries of the extended submodel do not have any heat exchange.

Second, the iteration between submodel and package should also be considered. For IC chip, package parts are much larger in area than the chip itself, and we call the package just vertical of the chip with the same area as the *center package*, and the other parts as *edge package*. In order to consider both center package and edge package effects for the extended submodel, we assume each submodel also has a package with larger area. The heat dissipation of the package is modeled by convection resistors from package thermal nodes to the ground. Then, we need to determine the convection resistances for center package and edge package of the extended submodel. For the center package, we can simply use the original convection resistance value since center structure is not changed at all. For the four edge package convection resistances $R_e$ at four directions, we propose a simple model to calculate their values:

$$R_e = \frac{R_o}{m(1-2x)}, 0 < x < 1, 0 < m < 1 \tag{5}$$

where $R_o$ is the full chip edge package convection resistance at the corresponding direction, $m$ is the normalized length of the extended submodel boundary, $x$ is the normalized distance of the boundary to the full chip edge. This model has good accuracy as will be shown in the experiment, and it can also be simply verified with two extreme boundary position cases by




(a) Temperature traces with centralized MPC based DTM.

(b) Temperature traces with distributed MPC based DTM.

Fig. 3. Transient temperature traces of 16-core CPU. Both methods are activated at 200 second.

TABLE I
ACCURACY COMPARISON OF THREE KINDS OF SUBMODELS.

| Convec. Res. | Overlap. Cores | Temp Error (°C) | |
|---|---|---|---|
| | | max | avg |
| Yes | No | 7.17 | 4.48 |
| No | Yes | 31.46 | 29.68 |
| Yes | Yes | 1.31 | 0.67 |

varying south and east boundaries of the extended submodel in Fig. 1. One extreme case assumes two boundaries of extended submodel is at the center of the chip (line 3 in Fig. 1), then there will be zero package heat dissipation at the south and east edge packages because of the symmetry geometry. Another extreme case assumes boundary of extended model is just the edge of the chip (line 5 in Fig. 1), then the extended model is the full chip itself, which means south and east edge package convection resistances are the same as the full chip.

Now, each MPC in Fig. 2 is able to control the submodel just like in the traditional centralized case since the interactions in red dashed circle are eliminated by introducing the extended submodel and its edge package convection resistance model.

## IV. Experimental Results

In the experiment, the new method is implemented using MATLAB, and HotSpot [6] is used to build the thermal model of a 16-core chip. The extended submodel is just as shown in Fig. 1. The power and instruction information is obtained using Wattch [7] running SPEC benchmarks [8]. The ambient temperature and ceiling temperature are set as 20°C and 105°C, respectively.

First, we verify the accuracy of the new extended submodel with edge package convection resistance model. In order to show the effects of introducing the overlapping cores and edge package convection resistance model, we compare thermal simulation errors of three scenarios: Case 1, with convection resistor but without overlapping cores; Case 2, without convection resistors but with overlapping cores; Case 3, with both. The results are collected in Table I, showing the best accuracy is achieved with both overlapping cores and convection resistors. This result means both convection resistance model and overlapping cores play key roles in decoupling the extended submodels from each other to achieve a fully distributed system.

TABLE II
MANAGEMENT QUALITY COMPARISON OF THE NEW DISTRIBUTED METHOD (DIST.) WITH TRADITIONAL CENTRALIZED METHOD (CENT.).

| Method | Variance | | MIPS |
|---|---|---|---|
| | max | avg | |
| No DTM | 66.18 | 43.64 | 526 |
| Cent. | 8.56 | 3.60 | 518 |
| Dist. | 40.13 | 26.49 | 513 |

Then we compare the performance of the distributed MPC based DTM with the centralized one. For MPC assisted DTM method, there are two quality indeces. The first one is the variance of temperatures of cores. A good MPC based DTM method should make temperatures of all cores to track the provided temperature ceiling, leading to a small variance. The second index is the Million Instructions executed Per Second (MIPS). If the DTM performs well, the MIPS of the chip should be high with few temperature violations. The transient temperature simulation results are shown in Fig. 3, and the variance and MIPS results are collected in Table II. It is clear that the new method successfully managed temperature of the multi-core system, although the management quality is not as good as the centralized method (larger average variance and lower MIPS). But this is well expected as the trade-off of the decoupling error and MPC performance.

## V. Conclusion

In this paper, we have presented a novel distributed MPC based DTM method for multi-core systems. The new method divides the full chip thermal model into submodels, and decouples the interactions among submodels by introducing overlapping cores and edge package convection resistance model. Experiments show that the new method performs well in thermal management with fully distributed structure.

## References

[1] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, June 2006, pp. 78–88.

[2] A. Bartolini *et al.*, "Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 170–183, January 2013.

[3] F. Zanini *et al.*, "Multicore thermal management with model predictive control," in *Proc. European Conference on Cicuit Theory and Design*, 2009, pp. 90–95.

[4] Y. Wang, K. Ma, and X. Wang, "Temperature-constrained power control for chip multiprocessors with online model estimation," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, 2009, pp. 314–324.

[5] J. Ma *et al.*, "Hybrid dynamic thermal management method with model predictive control," in *IEEE Asia Pacific Conference on Circuits and Systems*, November 2014.

[6] W. Huang *et al.*, "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.

[7] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, 2000, pp. 83–94.

[8] J. L. Henning, "SPEC CPU 2000: Measuring CPU performance in the new millennium," *IEEE computer*, vol. 1, no. 7, pp. 28–35, July 2000.