Hybrid Dynamic Thermal Management Method with Model Predictive Control

Jian Ma*, Hai Wang*, Sheldon X.-D. Tan†, Chi Zhang*, and He Tang*
*School of Microelectronics & Solid-State Electronics,
University of Electronic Science & Technology of China, Chengdu, Sichuan, 610054 China
†Department of Electrical Engineering, University of California, Riverside, CA 92521 USA

***Abstract*—Dynamic thermal management methods are important to control the temperature of microprocessor at runtime and improve the reliability performance of the chip. In this paper, a model predictive control based dynamic thermal management method is proposed with hybrid thermal regulation techniques combining task migration and dynamic voltage frequency scaling (DVFS) methods. The new method is able to track the specified safe temperature ceiling, reduce processor performance degradation, and lower temperature variation among different cores.**

I. Introduction

High temperature caused by high power density of the chip has become one of the most important constraints for the evolution of microprocessor: the high temperature not only lowers the reliability of the chip, but also limits the performance of the processor [1].

Dynamic thermal management (DTM) techniques are introduced to mitigate the thermal induced reliability issues as well as enhance the performance of the processor under safe temperatures [2]. Effective DTM methods include Task Migration and Dynamic Voltage and Frequency Scaling (DVFS) techniques. In task migration method [3], [4], heavy load (high power consumption) tasks at high temperature cores migrate to low temperature cores with light load (low power consumption) tasks, and as a result, the temperature variation among cores is minimized. Although it is able to keep the high performance of the processor, it cannot guarantee the core temperatures to be within the safe temperature range. Instead, DVFS method [5], [6] keeps the core temperature in safe range through reducing the power consumption by means of lowering the voltage and frequency of a core. It is able to guarantee the temperature safety, but with the penalty of significant performance drop.

Model predictive control (MPC) [7] has been introduced in control community to track a target output in the future by computing the desired input trajectory. In [8], [9], MPC has been introduced for microprocessor DTM problems and used with DVFS method which directly takes the frequencies of cores as the control input. However, due to the limitation of DVFS introduced before, such algorithms have negative impact on the performance of the processor. Power consumption of a core is not only related to the frequency, but also associated with the application running on the core: with the same frequency, some applications consume much less power than others. As a result, in MPC assisted DVFS, there are cores which run at maximum frequency but still have low power consumptions, and probably have low temperatures, while at the same time, there are also cores whose frequencies have been greatly reduced in order to match the temperature targets on them.

In this paper, we propose a new hybrid dynamic thermal management method with model predictive control. The new method utilizes model predictive control to compute the desired power input distribution for a specified temperature target, then performs task migration and DVFS to adjust current power distribution to match the desired one. With the guidance of the weighted bipartite matching method, frequencies of only limited number of cores are scaled. The hybrid method takes advantages from both task migration and DVFS techniques: it is able to maximize the performance of the processor, minimize the temperature variations over cores, and also track target temperatures.

This research was supported in part by NSFC grant under No. 61404024, and in part by an initial startup grant from UESTC. Please address comments to Hai Wang (e-mail: wanghai@uestc.edu.cn).

II. Basics of Model Predictive Control with Thermal Application

In this section, the basics of thermal modeling techniques are presented first, followed with the introduction of the model predictive control method.

A. Thermal modeling

A thermal model of a microprocessor with m cores can be expressed as an ordinary differential equation, and then written into descretized form by applying Eular's method with fixed time step as

$$\begin{aligned} t_m(k+1) &= A_m t_m(k) + B_m p(k), \\ y(k) &= C_m t_m(k), \end{aligned} \tag{1}$$

where $t_m(k) \in \mathbb{R}^n$ is the thermal vector representing temperatures of n blocks of the processor (including m cores, with $m < n$) at time k; $A_m \in \mathbb{R}^{n\times n}$ and $B_m \in \mathbb{R}^{n\times m}$ contains thermal dynamic information determined by the thermal conductance, thermal capacitance, and the topology of the processor; $p(k) \in \mathbb{R}^m$ is the power vector of m cores at time

978-1-4799-5230-4/14/$31.00 ©2014 IEEE

k; $y(k)$ is the thermal vector of m cores; $C_m \in \mathbb{R}^{m\times n}$ is the output selection matrix which selects the m core temperatures from $t_m(k)$.

B. Model predictive control

For the completeness of this paper, model predictive control method is briefly introduced focusing on thermal application. Interested readers are referred to [7] for detailed MPC discussion.

In order to maximize the performance of the processor without jeopardizing the reliability of the chip, we want every core of the chip to be running with the load as heavy as possible resulting in a temperature just below the ceiling of the safe temperature range. In this work, MPC method is used to achieve this goal, with the ceiling temperature provided in a vector as

$$G = [g^T, g^T, ..., g^T]^T \in \mathbb{R}^{mN_p\times 1}.$$

In this vector, $g \in \mathbb{R}^{m\times 1}$ contains the ceiling safe temperatures of each core, N_p stands for a time frame from current to the N_p steps into the future, and is called the prediction horizon. In order to keep the core temperatures tracking the goal in the prediction horizon, at a time k, the future control trajectory (which is actually unknown and needs to be computed in the end) is introduced as

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

where N_c is called the control horizon, and $\Delta p(k) = p(k) - p(k-1)$. The prediction of core temperatures

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

where $y(k+j|k)$ is the predicted core temperatures at time $k+j$ using information of current time k, can be calculated assuming ΔP is known, using

$$Y = Vt(k) + \Phi\Delta P, \tag{2}$$

where V and Φ are shown in (3) on top of the next page, and A, B, C, and t used in (2) and (3) are from the augmented model (not discussed in this paper due to the page limit), with the structure as

$$A = \begin{bmatrix} A_m & 0_m \\ C_m A_m & I \end{bmatrix}, \qquad B = \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix},$$
$$C = \begin{bmatrix} 0_m & I \end{bmatrix}, \qquad t(k) = \begin{bmatrix} \Delta t_m(k) \\ y(k) \end{bmatrix},$$

where 0_m is a matrix with all zero elements with suitable size.

The cost function is constructed as

$$F = (G-Y)^T(G-Y) + \Delta P^T R \Delta P, \tag{4}$$

where $R = rI_{N_c\times N_c}$ is tuning matrix with r as the tuning parameter. Please also note that Y is a function of the unknown variable ΔP.

Optimization is performed to minimize (4) by taking the first derivative of (4) with respect to ΔP and making it equal to zero. The solution of ΔP is

$$\Delta P = (\Phi^T\Phi + R)^{-1}\Phi^T(G - Vx(k)). \tag{5}$$

At each MPC time k, we only use the first computed control signal $\Delta p(k)$ from (5) and update the power distribution as

$$\bar{p}(k) \leftarrow p(k) + \Delta p(k), \tag{6}$$

where $\bar{p}(k)$ is the updated power distribution. The resulting temperature $y(k)$ would track the desired temperature ceiling with the updated power input. In other words, the updated power input is the highest power can be reached without violating the temperature requirements.

III. Hybrid DTM method with MPC

As briefly introduced in Section II, MPC method can be used to maximize the performance of the processor without violating the temperature constraints. However, how to adjust the power input as desired in (6) still needs to be researched. In this section, a hybrid method combining task migration and DVFS is introduced to solve this problem.

A. Task migration with weighted bipartite matching method

We first consider task migration method only. Task migration method redistribute tasks among processor cores in order to reduce the temperature variation among cores. It usually involves swapping heavy load tasks from high temperature cores with light load tasks from low temperature cores.

The purpose of using task migration method in this work is different: we want to simply redistribute tasks (powers) as the "update" action in (6). First, the current power distribution of all m cores is represented as $p(k) = [p_1, p_2, \cdots, p_m]$, the target power distribution after task migration, i.e. the updated power distribution from MPC, is denoted as $\bar{p}(k) = [\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_m]$. To perform the "update" action using task migration, we only need to determine how to re-arrange the elements in $p(k)$ to match the corresponding elements in $\bar{p}$. This is an assignment problem, and can be formulated into a weighted bipartite matching problem. A weighted complete bipartite graph $\mathcal{G} = (U, W, E)$ is first built, with vertex sets $U = \{p_1, p_2, \cdots, p_m\}$, $W = \{\bar{p}_1, \bar{p}_2, \cdots, \bar{p}_m\}$, and edge set E contains connections between U and W with weights. As a complete bipartite graph, every vertex in U is connected with *all* vertices in W. The weight between p_i and $\bar{p}_j$ is defined as the

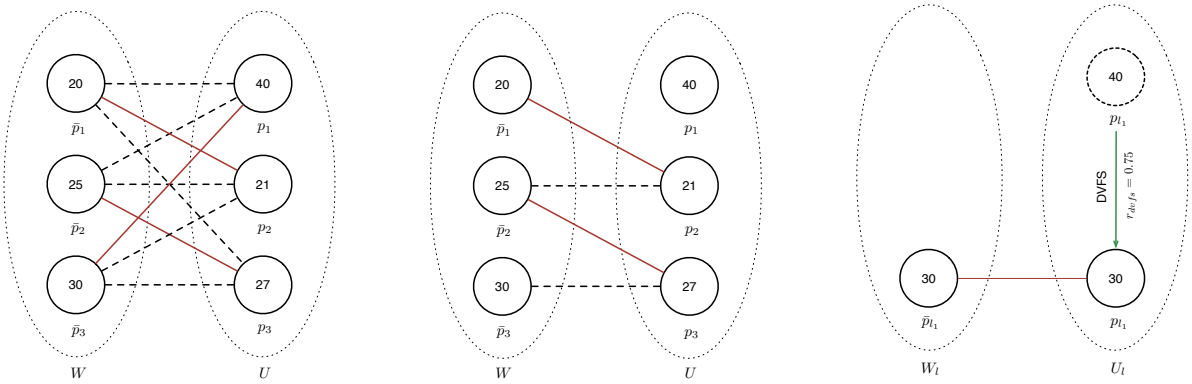
$$w_{ij} = |p_i - \bar{p}_j|. \tag{7}$$

Weighted bipartite matching algorithm is used on the graph, and the output of the algorithm is another bipartite graph with the same vertex sets U and V but every vertex in U (or W) is connected with only one vertex in W (or U) such that the cost (total weight of the output edges) is minimized.

An example of pure task migration method is provided in Fig. 1 (a). All the vertices are paired as expected, with the total cost of 13. It is also obvious that the pair $(\bar{p}_3, p_1)$, which contributes 10 in the total cost, is a worse match than the other two matched pairs.

Sometimes, pure task migration with MPC is able to enhance the performance of the processor. However, two problems are associated with pure task migration: first, if there are much more heavy load tasks than light load ones, the

$$V = \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^{N_p} \end{bmatrix}, \Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix} \tag{3}$$



(a) Pure task migration case.

(b) Hybrid case showing the weighted bipartite matching with the threshold of 5.

(c) Hybrid case showing DVFS process after (b) and then re-perform bipartite matching.

Fig. 1. Example of pure task migration (a) and hybrid method (b) (c) with weighted bipartite matching algorithm. The number in the vertex stands for the power value of the core. The solid red edges are the output edges connecting the matched vertex pairs, the dashed edges mean the corresponding two vertices are not matched. Weights of edges are not shown in the figure for simplicity.

resulting temperature will always be higher than the ceiling temperature, such as caused by the $(\bar{p}_3, p_1)$ pair in Fig. 1 (a); second, even if there are somewhat equal number of heavy and light load tasks, sometimes it is still difficult for the resulting temperature to track the ceiling temperature, because many times a good match (nearly zero cost) in bipartite matching algorithm cannot be found. The two problems are solved next by integrating DVFS and introducing a hybrid method.

B. Hybrid method with DVFS and task migration

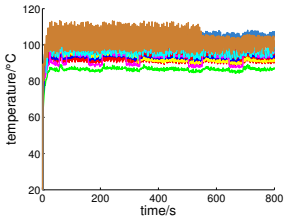
In this section, DVFS method is introduced to assist the bipartite matching algorithm to find the nearly perfect match, even when there are mostly heavy load tasks. In pure task migration case, all the match pairs are found directly, which include some bad match pairs due to the limitation of pure task migration method stated before. Instead, we first find the good pairs *only*, by removing the edges with weights larger than a threshold w_{th} from $\mathcal{G}$, and run bipartite matching algorithm on the modified graph $\tilde{\mathcal{G}} = (U, W, \tilde{E})$. After the first match, assume there are q leftover vertices from U (same number from W). The leftovers from U and W are collected in $U_l = \{p_{l_1}, p_{l_2}, \cdots, p_{l_q}\}$ and $W_l = \{\bar{p}_{l_1}, \bar{p}_{l_2}, \cdots, \bar{p}_{l_q}\}$, respectively. Assume average power from U_l and W_l are $avg(p_l)$ and $avg(\bar{p}_l)$, in case of the first problem (many heavy load tasks), there should have $avg(p_l) > avg(\bar{p}_l)$. This problem is solved by performing DVFS on the corresponding cores of U_l with the power scaling ratio of

$$r_{dvfs} = avg(\bar{p}_l)/avg(p_l), \tag{8}$$

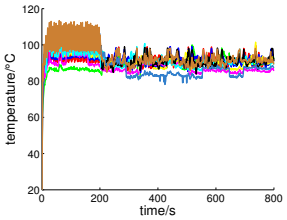
and perform bipartite matching again on the scaled leftovers.

It is also possible that there is $avg(p_l) < avg(\bar{p}_l)$ which means the first round leftovers are not caused by the first problem, or there are still leftovers after DVFS and the second biparte matching. The two cases both mean that we have encountered the second problem. To solve this problem, the threshold w_{th} is first relaxed to a higher value and run bipartite matching algorithm on the leftovers to seek for a sub-optimal solution. After this round, threshold is further raised to infinity (the vertex pair without edge are assumed to have the edge with infinity weight) in order to pair all vertices left (if any), and DVFS is performed on the poorly matched pairs which have a higher power value on the current power side compared to the MPC updated power side.

A simplified example of the hybrid method is shown in Fig. 1 (b) and (c). The modified weighted bipartite graph with the weight threshold 5 is shown in Fig. 1 (b). The edges in Fig. 1 (a) which have weights larger than 5 are removed in this new figure. The resulting matched pairs are only $(\bar{p}_1, p_2)$, and $(\bar{p}_2, p_3)$, with total cost of 3. $\bar{p}_3$ and p_1, which cannot be matched well with any other vertices, are leftovers of the first round. They are collected into the leftover group U_l and W_l and re-numbered as p_{l_1} and $\bar{p}_{l_1}$ as shown in Fig. 1 (c). Then, DVFS is performed on p_{l_1} with the scaling ratio

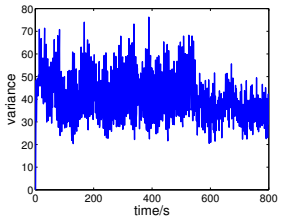


(a) Temperature traces in "free run".

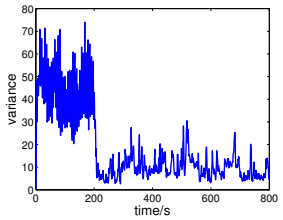


(b) Temperature traces with the new DTM method activated at the 200 second.

Fig. 2. Comparison of transient temperature traces of 9 cores.



(a) The temperature variance of the "free run".



(b) The temperature variance with the new DTM method activated at the 200 second.

Fig. 3. Comparison of the variance among 9 cores.

$r_{dvfs} = 0.75$ calculated using (8). Finally, $\bar{p}_{l_1}$ and the scaled p_{l_1} are matched with the second round weighted bipartite matching. In this example, there is no more leftovers and this ends the new algorithm. However, if there are still leftovers after the second round matching, the threshold will be raised to a higher value and then to infinity, and DVFS will be performed on the suitable poorly matched pairs as presented in the new algorithm.

IV. Experimental Results

The experiments are performed on a Linux server with 3.10GHz quad-core CPU and 4GB memory. The new method is implemented in Matlab R2010a, and HotSpot [10] is used to build the thermal model based on the configuration of a microprocessor with 9 cores distributed in a 3×3 fashion. Wattch [11] is used to generate the power information with SPEC benchmarks [12]. We use 9 power traces on 9 different SPEC benchmarks as the power traces of the 9-core microprocessor, and the simulation time step is set to be 1 second, with the ambient temperature set to be $20°C$.

First, we let all 9 cores run at maximum speed without any DTM methods involved. This "free run" transient temperature traces are shown in Fig. 2 (a). It can be seem that the cores have temperatures as high as up to $110°C$, which puts the chip in great danger. Moreover, temperature variance among cores is shown in Fig. 3 (a), which reveals the large temperature differences in different cores.

Next, we set the safe temperature ceiling as $90°C$ for all cores, and activate our hybrid DTM with MPC at the time of 200 second. The corresponding transient temperature traces are given in Fig. 2 (b), which shows that temperatures from all cores track the given ceiling of $90°C$ as soon as the new DTM method takes effect at 200 second. In order to study the temperature variation among cores with the new DTM method, the corresponding temperature variance is demonstrated in Fig. 3 (b). It is obvious that the temperature variance among cores has been greatly reduced using the new method.

V. Conclusion

In this paper, an MPC based hybrid DTM method combining task migration and DVFS techniques is proposed. The new method utilizes MPC to calculate the suitable power consumption to track a specified safe temperature ceiling. An algorithm, with both task migration and DVFS, is developed to adjust current core power consumptions according to the computed ones from MPC. The new method is able to track the given temperature with limited frequency scaling actions to avoid significant performance degradation of processor.

References

[1] D. Brooks, R. Dick, R. Joseph, and L. Shang, "Power, thermal, and reliability modeling in nanometer-scale microprocessors," *IEEE Micro*, vol. 27, no. 3, pp. 49–62, May-June 2007.

[2] J. Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proceedings of the 33rd annual international symposium on Computer Architecture*, ser. ISCA '06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 78–88. [Online]. Available: http://dx.doi.org/10.1109/ISCA.2006.39

[3] M. D. Powell, M. A. Gomaa, and T. N. Vijaykumar, "Heat-and-Run: Leveraging SMT and CMP to manage power density through the operating system," in *Proc. of International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2004, pp. 260–270.

[4] Y. Ge, P. Malani, and Q. Qiu, "Distributed task migration for thermal management in many-core systems," in *Proc. Design Automation Conf. (DAC)*, 2010, pp. 579–584.

[5] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *Proc. of Intl. Symp. on High-Performance Comp. Architecture*, 2001, pp. 171–182.

[6] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, 2003, pp. 2–13.

[7] L. Wang, *Model Predictive Control System Design and Implementation Using MATLAB*. Springer, 2009.

[8] F. Zanini, D. Atienza, L. Benini, and G. De Micheli, "Multicore thermal management with model predictive control," in *Proc. 19th European Conference on Cicuit Theory and Design*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 90–95.

[9] 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.

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

[11] 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.

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