Distributed Task Migration for Thermal Hot Spot Reduction in Many-core Microprocessors

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

***Abstract*—In this paper, we propose a new distributed task migration method to reduce the thermal hot spots and on-chip temperature variance, which leads to better thermal reliability and reduced package costs of emerging many-core processors. The novelty of the new algorithm is that the task migration is done in a fully distributed way while we can still maintain some degrees of global view to guide the process. This is enabled by recently proposed distributed state tracking technique to dynamically estimate the average temperature of all the cores, which provides the important global view of the temperature of the whole chip to efficiently guide local task migration among cores. In addition, the local task migration will be carried out based on the power, temperature, and load influence from neighboring cores. Our experimental results on a 36 core microprocessor demonstrate that the proposed method can reduce 30% more thermal hot spots compared with the existing distributed thermal management method, leading to more balanced temperature distribution of many-core microprocessor chips.**

I. Introduction

Thermal constraints are the major driving forces for wide adoption of multi/many core architectures. As the power density continues to increase, the excessively high temperature spots, called *thermal hot spot* would adversely lead to performance degradation, increased cooling costs, reduced reliability and signal integrity issues for multi-core microprocessors [1]–[4]. Thus, thermal management and related cooling techniques has been identified by ITRS [5] as one of the five key challenges during the next decade for achieving the projected performance goals of the industry.

The goal of thermal management is to keep the multi-core microprocessor system working below a safe temperature level while maintaining the efficient task execution performance. Some methods for runtime thermal management are involved with a dynamic power reduction technique, called *Dynamic Voltage and Frequency Scaling* (DVFS) [1], [6], [7]. These types of method could quadratically reduce the dynamic power through reducing the supply voltage and operating frequency, which translates to reduced data rate. Thus, the DVFS based thermal management method achieves temperature reduction at the expense of reduced performance.

In order to maintain the performance of the processor system, task scheduling/migration technique is served as an alternative to control the temperature of the cores [3], [8], [9]. The general idea of these methods is to migrate the heavy loaded task away from an overheated core to a cooler core. The traditional approach like [8], typically migrate the heaviest task (with largest power consumption) to the coolest cores to balance the temperature profile of all the cores. However, such an intuitive decision may lead to sub optimal result, because it does not take into account of the influence from the neighboring cores. To improve the thermal profile, recent work [3] proposed a new thermal management approach that used a temperature incremental factor to count the heat flux from the neighboring cores.

This work is supported in part by NSF grant under No. CCF-1017090, in part by NSF Grant under No. CCF-1255899, in part by Semiconductor Research Corporation (SRC) grant under No. 2013-TJ-2417 and a startup grant from UESTC.

Also, most existing approaches [3], [8]–[10] are centralized control approaches, which requires a Centralized Manager (CM) to monitor the temperature and power of each core in the many-core microprocessor, and globally reallocate the resources and schedule the tasks. The centralized control method, however, will not be scalable in the context of future many core processors with hundreds and even thousands of cores [11], [12]. It will suffer high computational costs inside the CM, single point of failure, large volume of monitoring and communication traffics around CM. To mitigate these problems of the centralized control approach, the distributed control approaches have been proposed [13], [14].

In this work, we aim at developing a fully distributed thus scalable task scheduling/migration method to reduce the thermal hot spots and temperature variation of many-core microprocessors. To better perform the local task migrations, we activate our task migration scheme based on the distributed estimated global temperature, and look at the temperature and load of the current core as well as those of immediate neighborhood cores to make better task migration decisions. Our experimental results on a 36 core microprocessor demonstrate that the proposed method can reduce 30% more thermal hot spots compared with the existing distributed thermal management method, leading to more balanced temperature distribution of many-core microprocessor chips.

II. The proposed distributed thermal management method

The proposed fully distributed thermal management method has the following two features: (1) the task migration is controlled distributedly, and it considers load influence from its adjacent cores. (2) the avarage temperature is tracked in a distributed way, using the recently proposed distributed average signal tracking algorithm.

A. The distributed estimation of average temperature via a nonlinear dynamic tracking algorithm

In distributed control scheme, the average temperature cannot be calculated by taking the mean of the temperature values across all the microprocessor cores because each core is only aware of its own temperature and the temperature of its immediate neighboring cores. Thus, a distributed state based average temperature estimation method is proposed to estimate the average temperature across all the cores without monitoring the global temperature of the chip, and a brief review of this technique is presented below.

Distributed average tracking theory for multi-agent system has proved that the states of all the distributed agents could converge to the average value of the time-varying reference signals [15], [16]. According to [16], the state equation of agent i of the distributed system can be formulated as

$$\begin{aligned} \dot{z}_i(t) &= \alpha \sum_{j\in N_i} sgn[x_j(t) - x_i(t)] \\ x_i(t) &= z_i(t) + r_i(t) \end{aligned} \tag{1}$$

in which $x_i(t)$ is the local state of the distributed agent, $r_i(t)$ is the reference signal with bounded derivatives in a finite time, sgn is the sign function, and $\alpha > 0$ is a constant. N_i denotes the state at the neighborhood of agent i. The distributed system assumes that the value of each reference $r_i(t)$ is local to agent i only (temperature of core i), and the value of the state $x_i(t)$ is known only by the distributed agents in its neighborhood.

Suppose that the changing rate of $r_i(t)$ is bounded by β. For the algorithm (1), our theoretical analysis in [17] shows that if the graph G is undirected and connected, and $\alpha > \beta$, then each estimate $x_i(t)$ approaches the average reference signal $\frac{1}{N}\sum_{j=1}^{N} r_j(t)$ in finite time, and the convergence time is guaranteed to be upper bounded by $\frac{1}{2(\alpha-\beta)}\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}|r_i(0) - r_j(0)|$. We can easily select the parameter α to ensure an arbitrary desired convergence time, and use the converged local states to represent the average temperature of the cores.

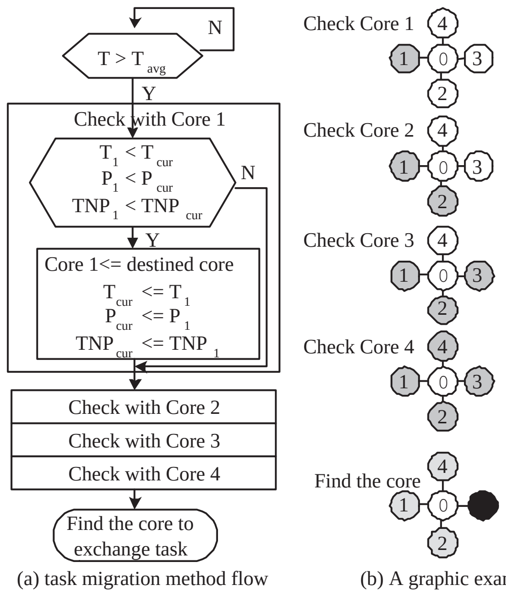
B. The proposed new distributed task migration method

Task migration policy is the most important part of thermal management of multi/many-core microprocessors. In the existing DTB-M [14], the overall goal is to reduce the average temperature across all cores through task migrations. However, lower average temperature across all cores does not necessarily lead to more balanced temperature distribution, and thus the efficiency of reducing the number of thermal hot spots may be compromised. To reach more balanced on-chip temperature distribution and reduce the number of on-chip thermal hot spots more efficiently, we propose a new distributed thermal management method that considers the load influence from neighboring cores. Also, by employing average temperature tracking technique, the proposed thermal balance policy could identify the cores with temperature above the average value and perform only necessary task migration to effectively reduce the temperature, which translates to the reduced amount of data traffic between the cores due to task migrations.

In our proposed approach, the task migration policy will be activated for a processor core if the temperature of the core is above the average temperature (T_{avg}) of all the cores. Task migrations happen only between two immediate adjacent cores as shown in Fig. 1, where the circle nodes represent the cores, and core 1-4 are the immediate neighbors of core 0 in the center. We define the current core as the one that the task migration policy is applied to, the destined core as the core that could potentially exchange task with current core if all of the following criteria are satisfied:

1. $T_{des} < T_{cur}$, where T_{des} is the temperature of the destined core and T_{cur} is that of the current core.
2. $P_{des} < P_{cur}$, where P_{des} is the task load of the destined core in the new execution cycle, and P_{cur} is the counterpart of the current core.
3. $TNP_{des} < TNP_{cur}$, where TNP_{des} is the total task load including the immediate neighboring cores surrounding the destined cores in the new execution cycle, and TNP_{cur} is the counterpart of the current cores.

The distributed task migration method flow can be illustrated in Fig. 1. The task migration policy will be activated if the temperature of the current core is above the average temperature of all the cores (T_{avg}). The current core communicates with its neighboring cores to check if the migration criteria (1), (2) and (3) are satisfied. The flow starts with core 1(the left neighboring core). If the criteria are satisfied, core 1 is selected as the destined core for task migration; at the same time, the thermal and load parameters of the current core ($T_{cur}, P_{cur}, TNP_{cur}$) are updated by the thermal and load parameters of core 1 (T_1, P_1, TNP_1). Then the flow continues to check with core 2 (the core located at the downside of the current core), if the migration criteria are satisfied, it indicates that core 2 is a better migration choice with even lower T_2, P_2, and TNP_2 than the previous migration choice of the destined core, and thus will be selected to replace the previous choice (core 1 in this case) as the destined core for task migration. The same procedure will be repeated for core 3, and core 4. The flow will finally find the best choice for task migration among the 4 immediate neighboring cores, which has the lowest value of temperature (T), load (P), and total load (TNP). An example flow is shown in Fig. 1 (b) in which the filled grey nodes indicate the checked cores, and the filled black node indicates the chosen core for task migration at the end of the task migration method flow. In this example, the new method checks with all of the four immediate neighboring cores (stamped with grey) and finally selected core 3 as the destined core (stamped with black) to migrate the task to.



(a) task migration method flow (Applicable to any arbitrary core)
(b) A graphic example (the right core is chosen)

Fig. 1. The proposed distributed task migration method

In this way, the distributed thermal management method takes account of the temperature, the task load, and the influence of the task load of the neighboring cores to performs task migrations when the temperature of the core is above the average temperature T_{avg}, which is computed in a distributed way as mentioned in subsection II-A. By identifying the cores with temperature above average and apply thermal balance policy to them, elevated temperature could be avoided with only necessary task migrations, which means that minimum number of task migrations are performed.

C. The overall run-time thermal management scheme

Similar to [14], we also assume that each task occupies an equal slice of execution time, and power traces for each task could be obtained from OS or predicted from the history of the power traces using some estimation methods like time series prediction methods. At the beginning of each execution cycle, the proposed distributed task migration policy is applied to each core, and task migration between two adjacent cores is performed if the task migration criteria discussed in Fig. 1 is satisfied. In this way, the tasks are scheduled to balance the temperature profile of the multi/many-core microprocessors in the new execution cycle so that the on-chip temperature gradient and occurrence of thermal hot spots during the task execution should be reduced. The algorithm flow of the proposed thermal management scheme is summarized in Alg. 1.

Algorithm 1 Distributed thermal management framework

```
Require: Task loads, many-core processor configuration
Ensure: Optimized temperature distribution
  Start simulation at room temperature
  for each execution cycle do
    1. Simulate power traces under different task loads.
    2. Obtain temperature responses of the many-core microprocessor, and estimate average temperature using distributed state tracking algorithm.
    if migration criteria is met then
      Perform distributed task migration using the proposed scheme in Fig. 1 core by core.
    end if
  end for
```

III. Experimental results

A. Experiment setups

The proposed thermal management method is implemented using Matlab 7.0, and Hotspot [1] is employed to build the thermal model of a 36-core processor system, using 6×6 tile configuration. Each core of this processor has geometry of 4 mm $\times$ 4 mm, and the thickness of the processor chip is 0.15 mm. More detailed thermal package structure and material properties are summarized in Table I, in which k denotes thermal conductivity, and c denotes specific heat. The convection capacitance of the heat sink of the 36-core processor is set to 660 J/K. The sampling interval of the thermal data of Hotspot is set to be 30 μs to preserve simulation accuracy. The proposed task migration method is technology independent, which is applicable for different processor package and different technology. We used Wattch

TABLE I
The package structure and thermal properties of the 36-core microprocessor.

Components	Chip	Heat Spreader	Heat Sink
Thickness(mm)	0.15	1.00	6.90
k ($W/(mK)$)	100.0	400.0	400.0
c ($J/(m^3K)$)	1.75×10^6	3.55×10^6	3.55×10^6

as the architecture level power analysis tool [18], and simulate the detailed transient power using 'bzip', 'gzip', 'mcf', 'gcc', 'swim', 'mgrid' and 'galgel' from SPEC2000 benchmarks. During the simulation, the 36 cores randomly take the task assignments at the beginning of each execution cycle (2048 time steps). If the temperature of a certain core is above the average temperature of all the cores, the thermal management will be activated to check with its immediate neighboring cores to find a task migration choice. We compare the occurrence of thermal hot spots and the on-chip temperature gradient during the task execution under the proposed method with those under the existing DTB-M method in [14].

B. Performance evaluation of distributed thermal management

1) Application of the estimated global average temperature: Since the average temperature is an important thermal indicator to activate the task migration policy, an accurate average temperature estimation based on distributed states is the prerequisite for the proposed method. The result in Fig. 2 indicates that the distributed state of each core converges to the value of the average temperature of the whole many-core chip, and thus the thermal management policy could be activated once the temperature exceeds the value of the distributed states of local cores. In this way, we only perform the task migration on the cores whose temperature is above the average temperature, leading to reduced temperature gradients with only necessary task migrations. In contrast, traditional

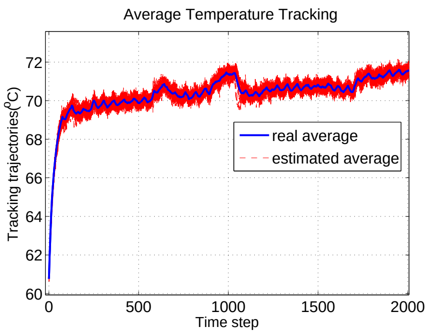


Fig. 2. The average temperature estimation based on distributed state tracking method

thermal management methods usually use fixed temperature as a threshold to activate the task migration policy, which have several problems as the value of the fixed temperature threshold is arbitrarily chosen. When the fixed temperature threshold is set too high, then no task migrations will be performed even though the whole chip can have large thermal gradients, which can still lead to thermal-induced reliability problems like thermal cycling [19]. On the other hand, when the fixed temperature is set too low, then more task migrations will be performed than necessary, leading to increased data traffic and performance overhead. Thus, the estimated global average temperature is more suitable for guiding the local task migrations.

2) Thermal performance comparison with DTB-M: Fig. 3 shows the comparison of transient temperature variance between the proposed method and the DTB-M method in [14]. It clearly shows that the proposed method would lead to reduced (better) temperature variance, which indicates more balanced on-chip temperature distribution across all the cores.

Fig. 4 shows the statistic of thermal hot spot occurrence during the task execution. The figure shows that the number of thermal hot spots (with temperature higher than 80 ^oC) encountered under the proposed thermal management method during task execution is reduced by more than 30% comparing with the same task execution under the existing thermal management method, which indicates that the proposed method

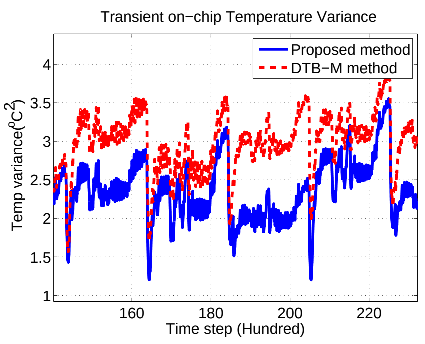


Fig. 3. Comparison of the transient temperature variance under different thermal management policy

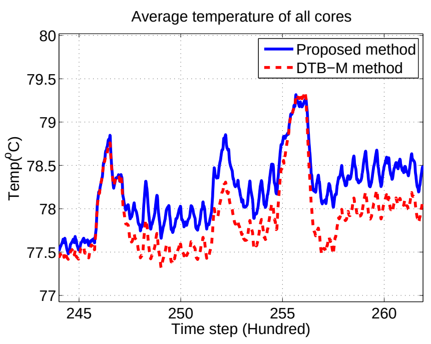


Fig. 5. Comparison of on-chip average temperature during task execution

is more efficient in reducing the number of on-chip thermal hot spots. The figure also shows that, the number of thermal hot spots over higher temperature is reduced quickly, which confirms that the proposed method could effectively reduce the temperature of the over-heated core, and thus prevent thermal emergency. As a result, compared to the DTB-M method [14], which is the existing distributed thermal-aware task migration method, the proposed method can remove more on-chip thermal hot spots and lead to more uniform temperature profile of many/multi-core microprocessors.

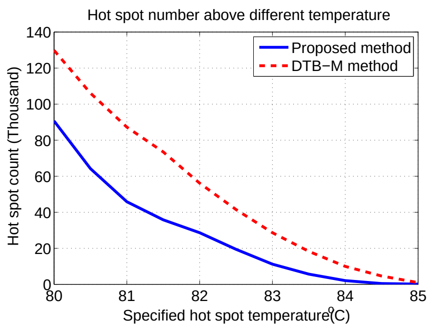


Fig. 4. The occurrence of thermal hot spots above different temperature levels during task execution.

In term of average temperature, DTB-M indeed delivers slightly lower average temperature during most of the transient simulation time as shown in Fig. 5, because it tries to reduce the average temperature of the chip. In comparison, the goal of the proposed method is to reduce the occurrence of thermal hot spots and maintain more balance temperature profile, which is more suitable for task migrations as temperature reductions should be more effectively achieved by power reduction techniques such as DVFS, power and clock gating schemes for multi/many-core microprocessors.

IV. Conclusion

The paper presented a new distributed thermal management to reduce the on-chip temperature variance and thermal hot spots during the task execution. Our experimental results on a 36-core microprocessor showed that the proposed method works more effectively to reduce the number of thermal hot spots (30% more thermal hot spot reduction comparing with the existing distributed thermal management methods). Also, the proposed method leads to smaller temperature variations across the many-core microprocessor, which has the potential to improve the chip reliability.

References

[1] 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)*, pp. 2–13, 2003.

[2] M. Hertl, D. Weidmann, and A. Ngai, "An advanced Reliability Improvement and Failure Analysis Approach to Thermal Stress Issues in IC Packages," in *35th International Symposium for Testing and Failure Analysis, ISTFA*, pp. 28–32, 2009.

[3] G. Liu, M. Fan, and G. Quan, "Neigbor-Aware Dynamic Thermal Management for Multi-core Platform," in *Proc. European Design and Test Conf. (DATE)*, pp. 187–192, 2012.

[4] A. Chakraborty, K. Duraisami, A. Sathanur, P. Sithambaram, A. Macii, E. Macii, and M. Poncino, "Implementation of a thermal management unit for canceling temperature-dependent clock skew variations," *Integration, the VLSI Journal*, vol. 41, no. 1, pp. 2–8, 2008.

[5] "International technology roadmap for semiconductors (ITRS), 2011edition," 2011. http://public.itrs.net.

[6] R. Mukherjee and O. S. Memik, "Physical aware frequency selection for dynamic thermal management in multi-core systems," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, pp. 547–552, 2006.

[7] S. Herbert and D. Marculescu, "Analysis of dynaic voltage/frequency scaling in chip-multiprocessors," in *Proc. Int. Symp. on Low Power Electronics and Design (ISLPED)*, pp. 38–43, 2007.

[8] M. Powell, M. Gomaa, and T. N. Vijaykumar, "Heat-and-run: leveraging smt and cmp to manage power density through the operating systems," in *ACM SIGPLAN NOTICES*, vol. 39, pp. 260–270, 2004.

[9] I. Yeo, C. C. Liu, and E. J. Kim, "Predictive dynamic thermal management for multicore systems," in *Proc. Design Automation Conf. (DAC)*, DAC '08, (New York, NY, USA), pp. 734–739, ACM, 2008.

[10] A. K. Coskun, T. S. Rosing, and K. C. Gross, "Utilizing predictors for efficient thermal management in multiprocessor SoCs," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, pp. 1503–1516, October 2009.

[11] S. Borkar, "Thousand core chips: a technology perspective," in *Proc. Design Automation Conf. (DAC)*, pp. 746–749, 2007.

[12] "Intel's 48-core Single-Chip Cloud Computer." http://www.intel.com/content/www/us/en/research/intel-labs-single-chip-cloud-computer.html.

[13] T. Ebi, M. A. Al Faruque, and J. Henkel, "TAPE: thermal-aware agent-based power economy for multi/many-core architectures," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, ICCAD '09, (New York, NY, USA), pp. 302–309, ACM, 2009.

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

[15] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, pp. 1520–1533, 2004.

[16] F. Chen, Y. Cao, and W. Ren, "Distributed average tracking of multiple time-varying reference signals with bounded derivatives," *IEEE Transactions on Automatic Control*, vol. PP, p. 1, 2012.

[17] F. Chen, Y. Cao, and W. Ren, "Distributed computation of the average of multiple time-varying reference signals," in *Proceedings of the American Control Conference*, (San Francisco, CA), pp. 1650–1655, July 2011.

[18] 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)*, pp. 83–94, 2000.

[19] "Failure Mechanisms and Models for Semiconductor Devices." In JEDEC Publication JEP122-A, Jedec Solid State Technolgy Association, 2002.