A Discrete Thermal Controller for Chip-Multiprocessors

Yingnan Cui
School of Computer Engineering
Nanyang Technological University
Singapore 679398
Email: yncui@ntu.edu.sg

Wei Zhang
Department of Electronic &
Computer Engineering
Hong Kong Univ. of Sci. and Tech.
Hong Kong
Email: wei.zhang@ust.hk

Bingsheng He
School of Computer Engineering
Nanyang Technological University
Singapore 679398
Email: bshe@ntu.edu.sg

***Abstract*—As the power density of modern processors keeps increasing, thermal management remains a challenging problem for processor designers. Among various solutions, closed-loop automatic thermal controllers have the benefits of fast response speed and high control accuracy. However, as a processor is a discrete system by nature, controllers designed by classic control theories fail to consider the system features related to the discreteness and thus cannot achieve optimal result. In this work, we propose a discrete thermal controller with the form of the digital filter with special concern about the frequency field response affected by the sampling process. We optimize the sampling period and the response time of the controller. Experimental results show up to 50% sampling frequency reduction and up to 25% improvement in the performance of CMP systems with thermal constraints when compared to other state-of-the-art closed-loop thermal controllers.**

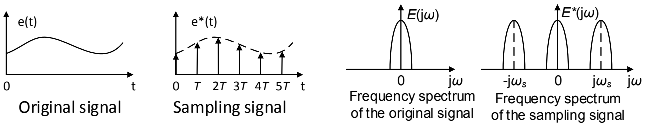
I. INTRODUCTION

Chip-level multiprocessors (CMP) have become the main stream architecture for modern microprocessors [1]. However, as the development in technology node has rapidly increased, thermal constraint has become a major barrier that limits the performance of CMPs. The increased power density due to feature size shrinking causes the temperature of the chips to rise, which may significantly damage the reliability of the chips [2]. In order to avoid thermal failures, it is necessary to maintain the temperature of the CMP below a threshold. In addition, the dark silicon problem [3] puts more stringent constraints on the temperature of processors. A common solution to the problem is to temporarily scale down the voltage and frequency level of the processors when thermal emergencies happen [4]. However, such techniques may significantly throttle the performance of the processor when the processor is under high workload and frequently meets thermal emergencies.

To better exploit the performance of the processors with thermal constraints, people start to design thermal management techniques that allow the processors to work under the possible highest voltage and frequency level without violating the thermal constraint [5], [6]. Among these techniques, the closed-loop automatic thermal controller is a promising solution. According to control theory, the closed-loop control systems have the benefits of high stability, low tracing error and fast response speed. Various closed-loop thermal controllers have been proposed in previous studies [7], [8], [9], [10]. However, these previous studies fail to consider the discreteness of the thermal controller for CMPs. As a part of a digital chip, the on-chip thermal controller is by nature a discrete control system. In previous studies, they first adopt the design methodology for continuous controllers and then covert the controllers into digital forms. When converting a continuous system to a discrete system, the frequency response of the system could be distorted. The distortion can be reduced by decreasing the sampling period, and therefore the controllers designed with the continuous methodology usually have to use smaller sampling period than the theoretical optimal value. This leads to significant decrease in performances of the processor. Indicated by our experiments, applying the theoretical optimal sampling period to these controllers may results in more than 10 times of thermal emergencies. And when compared to the optimal sampling period, the reduced of sampling period may cost up to 98.2% increase in the overhead of frequency switching.

In order to solve the above problem, we propose a digital thermal controller for CMPs using discrete control theory. The objective of the design is to further improve the performance of CMPs under thermal constraints by reducing the sampling period of the thermal controllers while maintaining satisfying control quality. Our design is distinctive when compared to previous studies in the following aspects. Firstly, we use discrete control theory to design the controller which avoids the distortion problem of the continuous design methodology and is able to work under the optimal sampling period. Secondly, to eliminate the influence of the high frequency components of the output signal of the thermal controller, we attached a low pass digital filter at the output of the our controller. In experiments, our discrete thermal controller achieves up to 50% sampling frequency reduction and up to over 25% improvement in the performance of CMP systems with thermal constraints when compared to other state-of-the-art closed-loop thermal controllers. In addition, we also propose an efficient thermal-aware scheduling algorithm regarding the characteristics of the discrete controller to further boost the performance of the CMP systems.

The rest of the paper is organized as follows. Section II



(a) Time domain

(b) Frequency domain

Fig. 1: The influence of sampling process.

introduces the background information for this work. Section III introduces the objectives and the detailed design of the discrete controller. Section IV proposes a novel thermal-aware scheduling algorithm concerning the system features for close-loop thermal controllers. Section V evaluates the performance of the controller and compares it with other state-of-the-art controllers. Section VI concludes this paper.

II. Background

A. Sampling Process

In a CMP, continuously monitoring the temperature change of each core causes unnecessarily heavy overhead. In reality, the temperature is measured by digital thermal sensors at each sampling point. As a result, the thermal controller also works at the sampling point and decides the power consumption of the CMP in the following sampling period. The sampling process and the sampling period have major impact on the property of the system, especially for the frequency characteristics.

Assume that a continuous signal is denoted by $e(t)$ and is sampled by a digital sensor every T seconds. The sampling signal produced by the digital sensor is denoted by $e^*(t)$. The definition of $e^*(t)$ is given by Eq. (1), where $\delta(t)$ is the unit impulse function, and n represents the number of sampling steps. The time domain description of the sampling signal is shown in Fig. 1a.

$$e^*(t) = e(t)\sum_{n=0}^{\infty}\delta(t-nT) = \sum_{n=0}^{\infty}e(nT)\delta(t-nT) \tag{1}$$

To analyze the frequency characteristics of the signals, we assume that $E(\mathrm{j}\omega)$ is the frequency spectrum of $e(t)$ and $E^*(\mathrm{j}\omega)$ is the frequency spectrum of $e^*(t)$, where j is the imaginary unit of complex numbers. Eq. (2) defines the relationship of $E(\mathrm{j}\omega)$ and $E^*(\mathrm{j}\omega)$, where $\omega_s = 2\pi/T$ is the angular sampling frequency. This equation can be obtained by a series of Fourier series expansion and Laplace transform [11]. From Eq. (2) we can see that $E^*(\mathrm{j}\omega)$ is the sum of copies of $E(\mathrm{j}\omega)$ shifted on the frequency axis, as shown in Fig. 1b.

$$E^*(\mathrm{j}\omega) = \frac{1}{T}\sum_{n=-\infty}^{\infty}E[\mathrm{j}(\omega+n\omega_s)] \tag{2}$$

We can see that in the frequency domain, sampling causes two problems. Firstly, as the natural signals usually have infinite bandwidth, the shifted components in the spectrum of the sampling signal may overlap with each other and cause distortion of the measured signals. Secondly, the sampling process significantly increases the amplitude of the spectrum in the high frequency range. The distortion of the spectrum of the signal caused by the sampling process affects the performance of the thermal controllers designed by the continuous methodology. To minimize the influence, such controllers have to reduce the sampling period [12] which leads to the increase of system overhead.

B. Thermal Model for CMP

According to [13], when considering the thermal interactions between different cores of a CMP, the temperature trace of core i in the CMP can be computed by Eq. (3). In the equation, τ_i is the temperature of the core i, C_i is the thermal capacitance of the core, NB is the set of neighboring cores surrounding core i, R_{ij} is the thermal resistance between core i and core j, p_i is the power consumed by core i, τ_a is the ambient temperature of the CMP, and R_{ai} is the thermal resistance between core i and the ambient environment.

$$C_i\frac{\mathrm{d}\tau_i}{\mathrm{d}t} = \sum_{j\in NB}\frac{\tau_j-\tau_i}{R_{ij}} + p_i + \frac{\tau_a}{R_{ai}} \tag{3}$$

Considering Eq. (3) for all the cores in a CMP, we can get Eq. (4), where $\boldsymbol{\tau} = [\tau_1,\ \tau_2,\ ...,\ \tau_n]^{\mathrm{T}}$ is a vector composed by the temperature of each core in the CMP, $\mathbf{p} = [p_1,\ p_2,\ ...,\ p_n]$ is the vector of the power consumption in each core, $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are coefficient matrices. $\mathbf{A}$ and $\mathbf{B}$ are both square matrices of $n\times n$ order , while $\mathbf{C}$ is a n matrix. The coefficient matrices can be acquired by transformation of Eq. 3.

$$\dot{\boldsymbol{\tau}} = \mathbf{A}\times\boldsymbol{\tau} + \mathbf{B}\times\mathbf{p} + \mathbf{C}\cdot\tau_a \tag{4}$$

C. Power Model for CMP

According to the thermal model, it is the power consumption of each core that directly determines the temperature of the CMP. The thermal controller controls the power consumption of the CMP by setting the voltage and frequency levels of each core and therefore a power model of the CMP is required.

During run-time, the power consumption of each core in a CMP can be estimated using performance counters [14]. Eq. (5) shows the dynamic power model for a core in the CMP, where R_{fetch}, R_{float}, $R_{decoder}$ and R_{alu} stand for the data reading of the performance counter at instruction fetching unit, floating point unit, decoder unit and the ALU respectively, and α_* are coefficients depending on architecture and technology node. The leakage power of can be estimated by Eq. (6) where N is the number of gates in the CMP, τ is the temperature of the core, and α_τ is a coefficient based on architecture and technology node. The techniques for retrieving the above architecture-based coefficients are also given by [14]. Eq. (5) and Eq. (6) describe the relationship between the power consumption and supply voltage V_{dd}. In the controller design, we apply the equations to decide the voltage value for the processors.

$$P_d = (\alpha_f R_f + \alpha_{fp}R_{fp} + \alpha_{dc}R_{dc} + \alpha_{alu}R_{alu})V_{dd}^2/t \tag{5}$$

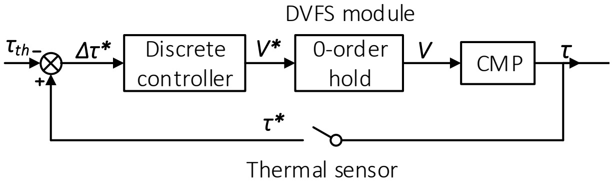


Fig. 2: The block diagram for the thermal controller.

$$P_{sub} = N \cdot \alpha_\tau \cdot \tau \cdot V_{dd} \tag{6}$$

III. Design of the Discrete Controller

A. Overview

Fig. 2 shows the block diagram of general closed-loop thermal control systems for CMPs with explicit concern for the discrete nature of the controllers. In the system, the thermal sensor measures the temperature at sampling points. The sampling signal of the temperature (τ^*) is compared with the threshold temperature ($\tau_{\mathbf{th}}$) and the difference ($\boldsymbol{\Delta}\tau_{\mathbf{th}}$) is sent to the thermal controller. The thermal controller then decides voltage level of the CMP. This voltage level ($\mathbf{V}^*$) is still a sampling signal because it is produced by the thermal controller only at sampling points. In reality, the voltage level of a processor (V) is a continuous signal. The conversion is performed by the DVFS module which set the voltage of processor at the desired level during the next sampling period. In control theory, this process is abstracted as a device called as the zero-order hold, the function of which is defined by Eq. (7).

$$e(nT + \Delta t) = e(nT) \tag{7}$$

For a thermal controller, the fundamental objective is to force the temperature of the processor tracing the threshold temperature with little errors. In this way, the processor always runs at the highest possible frequency level and thus the performance of the processor is maximized. In this study, our goal is to improve the performance of the system by increasing the sampling period of the thermal controller. Increasing the sampling period not only reduces the overhead of the thermal controller itself but also reduces the number of frequency level switching for the DVFS module which further reduces the overhead of the system.

As discussed in Section II-A, increasing the sampling period results in higher possibility of signal distortion caused by the sampling process. In order to maintain satisfying control quality of the thermal controller under large sampling period, we adopt the discrete control theory in our design. Firstly, we use the Shannon sampling theorem [15] to find an upper bound of the sampling period to avoid signal distortion. Secondly, we attach a digital filter to the controller to screen out the high frequency components contained by the output signal of the controller. Finally, by analyzing the discrete time response of the controller, we propose a controller design that is able to re-stabilize the temperature of the processor with minimal response steps after disturbance in temperature occurs.

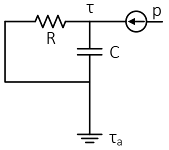


Fig. 3: Simplified thermal model for a CMP

B. Sampling Period Selection

Theorem 1 (Shannon sampling theorem)**.** *Assume signal $e(t)$ has a limited bandwidth ω_h. If $e(t)$ could be perfectly reproduced by the sampling signal $e^*(t)$, then the sampling period T must fulfill the following condition:*

$$T \leq \frac{2\pi}{2\omega_h}$$

. The Shannon sampling theorem [15] gives the theoretical maximal sampling period for a discrete control system. However, in this study, the temperature of CMP is a natural signal, which has infinite bandwidth. In order to use Shannon sampling theorem to decide the sampling period, we have to set a *cut-off frequency* for the temperature signal where the spectrum outside the cut-off frequency could be ignored.

To find a proper cut-off frequency, we need to analyze the frequency response of the system. We consider it is impossible for the temperature of the CMP to fluctuate beyond that frequency level. As shown in Section II-B, the thermal model for CMP system is a RC network. For ease of discussion, we use a simplified thermal model to consider the frequency response of the temperature of the CMP. As shown in Fig. 3, if we consider the power of the processor, denoted by p, as the input of the system, and consider the temperature, denoted by τ, as the output of the system. Then the frequency response of the thermal model is the same as a simple low-pass filter. The cut-off frequency for such a RC network is $1/(2\pi RC)$, which means the angular frequency is $\omega_h = 1/RC$. Then according to the Shannon sampling theorem, we get the maximum sampling period as $T = \pi RC$. Using the cutoff frequency of the thermal RC network is also crucial to guarantee the performance of the controller. It gives the controller enough response time before the temperature of the processor gets above the threshold temperature during next sampling period. In reality, the thermal RC network is more complicated than Fig. 3 and we use the technologies provided in [11] to decide the cutoff frequency of the real RC networks. The resulting sampling period is also rounded down to milliseconds for the ease of use.

C. Controller Design

The structure of the thermal controller is shown in Fig. 4. There are four components in the discrete controller: the decoupling unit, the power planner, the signal restore unit, the power model and the filter. Based on the decoupled input variables, the power planer decides the power consumption

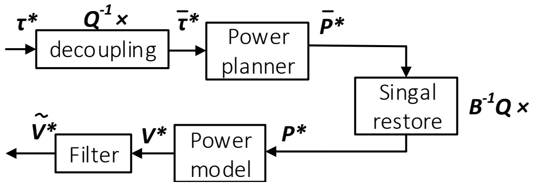


Fig. 4: The structure of the discrete thermal controller.

for each core using a minimal step controller design which minimizes the response steps for the controller to stabilize the system. The power consumption decided by the controller should be converted into voltage and frequency levels. This is performed by the the power model, which computes the voltage and frequency level based on the power model introduced in Section II-C. Finally, the filter after the controller is a finite impulse response (FIR) low pass filter, which screens out most of the high frequency components contained by the discrete signals before the discrete voltage signal is converted to a continuous signal by the zero-order hold in the system.

Decoupling and signal restore. To decouple the thermal model represented by Eq. (4), we should apply linear transform on the equation so that we get Eq. (8), where $\bar{\mathbf{A}}$ is a diagonal matrix, and the diagonal elements are the eigenvalues of matrix $\mathbf{A}$. Assume the linear transform can be presented by a matrix $\mathbf{Q}$, then the relationship between the decoupled model and the original model is defined by Eq. (9). With the decoupling, the original thermal model can be described by a group of independent equations, as shown by Eq. (10). In Eq. (10), $\bar{\tau}_i$ is the new input variable of the thermal model, which is also a linear combination of the temperature of each core in the CMP. Similarly,$\bar{p}_i$ is the linear combination of the power consumption in each core. With the new group of independent equations, we can design the control policy to decide the power consumption for each core individually. And after the power planner generates the power consumption of the decoupled thermal model, $\bar{\mathbf{p}}$, we must restore it to the original power consumption of the system. According to Eq. (9), the signal restore unit perform this function by premultiply $\mathbf{B}^{-1}\mathbf{Q}$ to $\bar{\mathbf{p}}$.

$$\dot{\bar{\boldsymbol{\tau}}} = \bar{\mathbf{A}} \times \bar{\boldsymbol{\tau}} + \bar{\mathbf{p}} + \bar{\mathbf{C}} \cdot \tau_a \tag{8}$$

$$\bar{\boldsymbol{\tau}} = \mathbf{Q}^{-1} \times \boldsymbol{\tau},\ \bar{\mathbf{A}} = \mathbf{Q}^{-1}\mathbf{A}\mathbf{Q},\ \bar{\mathbf{p}} = \mathbf{Q}^{-1}\mathbf{B} \times \mathbf{p}. \tag{9}$$

$$\dot{\bar{\tau_i}} = \bar{a}_i\bar{\tau}_i + \bar{p}_i + \bar{c}_i\tau_a \tag{10}$$

Power planner. According to discrete control theory, the characteristic of a discrete system can be described by the impulse transfer function, which could be acquired by applying *z-transform* to the system model. When applying z-transform to the decoupled system model shown by Eq. (10), we have the impulse transfer function as shown by Eq. (11). Assume that the impulse transfer function for controller power consumption p_i is $H_i(z)$. In the thermal control system, the threshold temperature keeps a constant during run time, which can be viewed as a step function. In this condition, the impulse transfer function of the minimal-step controller for the system is shown be Eq. (12). When the impulse transfer function of the system has the form of Eq. (12), the impluse tranfer function for the error between the desired temperature level and the actual temperature, denoted by $\Phi_i(z)$, is defined by Eq. (13). When the input temperature $\bar{\tau}_i$ is a unit step function, where $\bar{\tau}_i(z) = z^{-1}/(1-z^{-1})$, then the error signal $\bar{\tau}_i^*(z) = \Phi_i(z)\bar{\tau}_i(z) = 1$. This means the error signal $\bar{\tau}_i$ has the value 1 at the first step and in all the following steps, the value of $\bar{\tau}_i$ is 0. Obviously, the system could trace the step functions after only one step. Now we substitute Eq. (11) into Eq. (12), and then apply the reverse z-transform, we can get the difference equation description of the controller, which is defined by Eq. (14). We can see that the controller as the same form as a infinite impulse response (IIR) filter.

$$G_i(z) = \frac{\bar{\tau}_i(z)}{\bar{P}_i(z)} = \frac{z^{-1}}{1 - \mathrm{e}^{\bar{a}_i T} z^{-1}} \tag{11}$$

$$H_i(z) = \frac{\bar{P}_i(z)}{\bar{\tau}_i(z)} = \frac{z^{-1}}{(1 - z^{-1})G_i(z)} \tag{12}$$

$$\Phi_i(z) = \frac{\bar{\tau}_i^*(z)}{\bar{\tau}_i(z)} = \frac{1}{1 + H_i(z)G_i(z)} \tag{13}$$

$$\bar{p}_i(n) = \bar{\tau}_i(n) - \mathrm{e}^{\bar{a}_i T}\bar{\tau}_i(n-1) + \bar{p}_i(n-1) \tag{14}$$

FIR low pass filter. The filter after the controller has to block the high frequency components of the sampling signal from passing through. According to the Fourier transform in Section II-A, the cut-off angular frequency of the filter should be $\omega_h = \omega_s/2 = \pi/T = 1/RC$. Among the various kinds of FIR filter design, we select a 21-coefficient Hamming window FIR filter. The FIR filter has the form shown in Eq. (15).

$$\tilde{V}_i(n) = \sum_{k=0}^{20} a_k V_i^*(n-k) \tag{15}$$

IV. Power Stabilizing Scheduling Algorithm

In previous studies, DVFS-based thermal management methods are usually integrated with efficient thermal-aware scheduling algorithms to further boost the performance of the CMP under thermal constraints [16], [17], [8]. Previous thermal-aware scheduling algorithm usually aims at minimizing the peak temperature of the system by mapping the task with highest power consumption to the core with coolest temperature [18]. However, with our discrete thermal controller presented, such methodology is no longer effective. This is because our discrete thermal controller relies heavily on accurate power consumption number of each core. If the power profile of the mapped tasks on each mismatches the desired power consumption used in the discrete controller, the error could be amplified through the decoupling step in the controller could cost a large overhead for the controller to reestablish a stable state for the temperature. Based on this observation, we propose an efficient thermal-aware scheduling

Algorithm 1 Power Stabilizing Algorithm

```
   Input: $\mathbf{P(n)}$, $\mathbf{P(n-1)}$.
1: Scale power consumption to normal $V_{dd}$.
2: Sort $\mathbf{P(n)}$, $\mathbf{P(n-1)}$ in ascending order.
3: while $\mathbf{P(n)} \neq \emptyset$ do
4:     for all $p_i(n) \in \mathbf{P(n)}$, $p_j(n-1) \in \mathbf{P(n-1)}$ do
5:         Find minimal $|p_i(n) - p_j(n-1)|$
6:         Transfer task in core $i$ to core $j$
7:         remove $p_i(n)$, $p_j(n-1)$ from $\mathbf{P(n)}$, $\mathbf{P(n-1)}$
8:     end for
9: end while
```

algorithm for our discrete thermal controller. The algorithm is called as the power stabilizing algorithm (PSA).

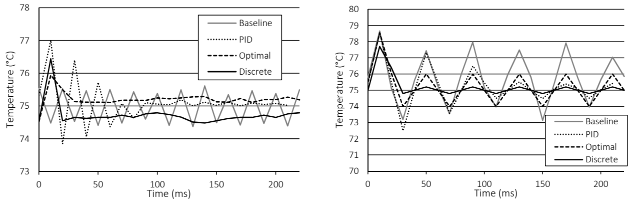
The pseudo code of the PSA is shown in Algorithm 1. The input of the algorithm is the list of power consumption estimated for the current sampling period, denoted as $\mathbf{P(n)}$, and the power for last sampling period $\mathbf{P(n-1)}$. We first scale the power consumption to the highest V_{dd} level of the CMP for fair comparison (line 1). The scaling is according to the power model shown in Section II-C. Then from the two lists, we find the two power consumption with the minimum difference $p_i(n)$ and $p_j(n-1)$ (line 5). This means that the power consumption of core j in last sampling period is most similar to the power consumption of core i in the current period. To keep the power consumption stable, we transfer task running in core i to core j (line 6). We continue this procedure until all tasks are mapped. We note that the scheduling must happen before the thermal controller makes the decision. Because the thermal controller decides the voltage level of next period according to the power consumption of the current period, because it is hard to know the exact power consumption of each task in the future. After the scheduling, the controller decides the voltage level for the next period according to the newly assigned power figure, the frequency level is more likely to keep stable. Such assumption is also adopted by previous studies like [19].

V. Evaluation

A. Experiment Setup

In the experiments, we use HotSpot 5.02 as our thermal simulation platform for the microprocessor [13]. The target CMP system is assume to be a 45nm ARM Cortex-A7 processor with four cores. The DVFS modulation is assumed to be able to adjust the frequency of the CMP for 16 uniformly distributted levels from 0.9GHz to 1.5GHz. We use two benchmarks in the experiments. We first use a microbenchmark, which is a piece of program containing a endless loop of floating point computation, to test the step response of the closed-loop control system. The second benchmark is the SPEC CPU2006 suite, which is used to test the performance of system under real-world applications. We use the GEM5 + MCPAT simulation platform to collect the power trace of the benchmarks [20], [21] .

In the experiments, three different kinds of thermal controllers are adopted for comparison. First, we use a simple threshold-trigger DVFS controller as the baseline of the experiments. Second, we adopt a PID controller which is proposed in [9]. The PID controller is the most commonly adopted controllers in control systems. Finally, we adopt a state-of-the-art thermal controller design from [6]. The controller is designed using the optimal control theory, therefore it is referred to as the optimal controller.



(a) $T = 10\text{ms}$

(b) $T = 20\text{ms}$

Fig. 5: The step responses of the thermal controllers.

B. Step Response

In the first experiment, we set the sampling period as 10ms, which is a relatively small value. Fig. 5a shows the temperature responses to the disturbance of one same core in the CMP. Except for the baseline controller, all the other controllers are able to re-stabilize the temperature of the CMP to the threshold temperature. The stable state error of the temperature tracing is less than 0.6% for all the controllers. When compared to the PID controller and the optimal controller, our discrete controller stabilize the temperature with shortest time. This is because the minimal-step controller design guarantees the system can be re-stabilized within minimal number of steps.

In the second experiment, we set the sampling period of the controller as 20ms, which is a little less than the maximal sampling period computed by the method introduced in Section III-B. Fig. 5b shows the temperature traces of the CMP with different controllers after temperature becomes relatively unstable. It is obvious that with larger sampling period, the temperature maintained by the baseline, the PID and the optimal controller all become less stable compared to Fig. 5a. However, with special concerns regarding the sampling period and with the low pass filter, our discrete controller outperforms the others. It stabilized the temperature within three sampling period and the steady state error is also the lowest.

C. Real Applications

To achieve satisfiable control quality, the sampling period of the baseline, the PID and the optimal controller are set to 10ms. However, for our discrete controller adopts the 20ms sampling period. In the experiments, each of the four cores is assigned with a different benchmark from the suite. The 12 benchmarks are divided into three groups to be tested together in one experiment.

Table I shows the execution time of the benchmarks in the processors using different kinds of thermal controllers. When compared to the maximal performance, the average increases of the execution time of the benchmarks are 31.23% (baseline), 22.9% (PID), 11.64% (optimal) and 7.02% (Discrete) respectively. The baseline controller results in the worst performance

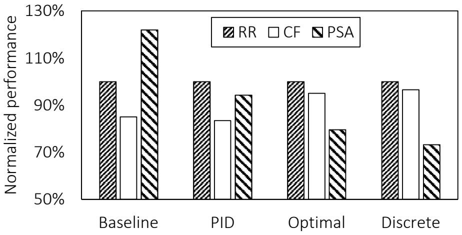


Fig. 6: Normalized performance of the CMPs with different scheduling algorithms

because it switches between the two voltage levels. The discrete controller achieves best performance among the thermal controllers for two reasons. Firstly, the minimal-step controller design adjusts the temperature level quickly to the threshold and thus reduces the unnecessary slow down of the execution speed. Secondly, the discrete controller used a much larger sampling period which results in less number of frequency level switches and therefore significantly reduces overhead.

TABLE I: The execution time of the benchmarks on the CMP.

Bench. #	Execution time (s)				
	core #	Baseline	PID	Optimal	Discrete
400	1	1325.09	1268.35	1103.87	1081.08
401	2	1746.96	1573.25	1436.19	1393.00
403	3	1163.60	1080.28	979.75	944.91
429	4	4223.52	4051.63	3847.04	3425.52
445	1	331.55	329.66	293.02	299.26
456	2	1304.54	1177.27	1133.52	1037.06
458	3	12967.80	11636.82	10012.54	9446.59
462	4	154.19	150.48	140.16	149.94
464	1	84086.11	81657.78	78418.86	67466.61
471	2	1612.66	1408.80	1283.19	1193.24
473	3	21398.60	20333.68	17897.88	16474.53
483	4	377.45	355.36	307.68	298.04

D. Integration with Scheduling Algorithm

Finally, we test the performance of the system by integrating thermal-aware scheduling algorithm with the thermal controllers. Fig. 6 shows the normalized performance of the CMP running the SPEC CPU2006 benchmark with different combination of thermal controllers and scheduling algorithms. The sampling period of the controllers are set as $20ms$. In the experiments, we select three types of scheduling algorithms, the round-robin (RR), coolest first (CF) and our power stabilizing algorithm (PSA). The results are normalized to the RR algorithm which is the baseline algorithm. From the results we can see that the CF algorithm performs best for the baseline controller and the PID controller. This is because for these two controllers, the temperature changes significantly and the difference between coolest and hottest cores are quite obvious. As for the optimal and discrete controller, our PSA shows the best performance because these two controllers controls the temperature with higher quality and temperature difference between cores are not obvious. In such cases, stabilizing the power consumption of the tasks in each core successfully reduces the times of frequency level switching for each core and thus results in optimal performance.

VI. Conclusion

We design the thermal controller with the form of the digital filter with special concern about the frequency field response affected by the sampling process. We optimize the sampling period and the response time of the controller. Experimental results show up to 50% sampling frequency reduction and up to 25% improvement in the performance of CMP systems with thermal constraints when compared to other state-of-the-art closed-loop thermal controllers.

References

[1] D. D. Awschalom and M. E. Flatté, “Challenges for semiconductor spintronics,” *Nature Physics*, vol. 3, no. 3, pp. 153–159, 2007.

[2] A. B. Kahng, “The itrs design technology and system drivers roadmap: Process and status,” in *DAC ‘13*.

[3] J. Henkel and *et al.*, “New trends in dark silicon,” in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015, pp. 1–6.

[4] H. F. Sheikh, I. Ahmad, Z. Wang, and S. Ranka, “An overview and classification of thermal-aware scheduling techniques for multi-core processing systems,” *Sustainable Computing: Informatics and Systems*, vol. 2, no. 3, pp. 151–169, 2012.

[5] T. Ebi, M. Faruque, and J. Henkel, “Tape: Thermal-aware agent-based power econom multi/many-core architectures,” in *ICCAD’09*. IEEE, 2009, pp. 302–309.

[6] Y. Wang, K. Ma, and X. Wang, “Temperature-constrained power control for chip multiprocessors with online model estimation.” in *ISCA*. ACM, 2009, pp. 314–324.

[7] K. Skadron, T. F. Abdelzaher, and M. R. Stan, “Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management.” in *HPCA*, 2002, pp. 17–28.

[8] Y. Wang, K. Ma, and X. Wang, “Temperature-constrained power control for chip multiprocessors with online model estimation,” in *ACM SIGARCH computer architecture news*, vol. 37, no. 3. ACM, 2009, pp. 314–324.

[9] Y. Fu, N. Kottenstette, C. Lu, and X. D. Koutsoukos, “Feedback thermal control of real-time systems on multicore processors.” in *EMSOFT*. ACM, 2012, pp. 113–122.

[10] M. M. e. Sabry, “Energy-efficient multiobjective thermal control for liquid-cooled 3-d stacked architectures,” *Computer-Aided Design of Integrated Circuits and Systems, IEEE Tran. on*, vol. 30, no. 12, pp. 1883–1896, 2011.

[11] A. V. Oppenheim, A. S. Willsky, and S. H. Nawab, *Signals &Amp; Systems (2Nd Ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1996.

[12] B. C. Kuo, *Automatic control systems*. Prentice Hall PTR, 1981.

[13] K. Skadron and et al., “Temperature-aware microarchitecture,” in *ISCA*, 2003, pp. 2–13.

[14] W. L. Bircher and L. K. John, “Complete system power estimation using processor performance events,” *Computers, IEEE Transactions on*, vol. 61, no. 4, pp. 563–577, 2012.

[15] A. J. Jerri, “The shannon sampling theoremits various extensions and applications: A tutorial review,” *Proceedings of the IEEE*, vol. 65, no. 11, pp. 1565–1596, 1977.

[16] X. Zhou, J. Yang, Y. Xu, Y. Zhang, and J. Zhao, “Thermal-aware task scheduling for 3d multicore processors,” *Parallel and Distributed Systems, IEEE Transactions on*, vol. 21, no. 1, pp. 60–71, 2010.

[17] H. Khdr and *et al.*, “mdtm: multi-objective dynamic thermal management for on-chip systems,” in *Proc. of the conference on Design, Automation & Test in Europe*. European Design and Automation Association, 2014, p. 330.

[18] K. Stavrou and P. Trancoso, “Thermal-aware scheduling: A solution for future chip multiprocessors thermal problems,” in *Digital System Design: Architectures, Methods and Tools, 2006. DSD 2006. 9th EUROMICRO Conf. on*. IEEE, 2006, pp. 123–126.

[19] X. Zhou, J. Yang, Y. Xu, Y. Zhang, and J. Zhao;, “Thermal-aware task scheduling for 3D multicore processors,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 1, pp. 60–71, Feb. 2010.

[20] N. Binkert and *et al.*, “The gem5 simulator,” *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[21] S. Li and *et al.*, “Mcpat: an integrated power, area, and timing modeling framework for multicore and manycore architectures.” in *MICRO*. ACM, 2009, pp. 469–480.