TheSPoT: Thermal Stress-Aware Power and Temperature Management for Multiprocessor Systems-on-Chip

Arman Iranfar, *Student Member, IEEE,* Mehdi Kamal, *Member, IEEE,* Ali Afzali-Kusha, *Senior Member, IEEE,* Massoud Pedram, *Fellow, IEEE*, David Atienza, *Fellow, IEEE*

***Abstract*— Thermal stress including temperature gradients in time and space, as well as thermal cycling, influences lifetime reliability and performance of modern Multiprocessor Systems-on-Chip (MPSoCs). Conventional power and temperature management techniques considering the peak temperature/power consumption do not provide a comprehensive solution to avoid high spatial and temporal thermal variations. This work presents TheSPoT, a novel multi-level thermal stress-aware power and thermal management approach for MPSoCs. At the top level, core consolidation and deconsolidation is performed based on peak temperature, thermal stress, and power consumption constraints. These constraints are also used at the next level, where operating frequencies are determined. At this level we obtain optimal core frequencies by solving a convex optimization problem. However, thereafter, to reduce the runtime overhead in large MPSoCs, we alternatively propose to use a fast heuristic algorithm. The efficacy of the proposed approaches in reducing the thermal cycles and temporal/spatial temperature gradients is evaluated by comparing the results with the state-of-the-art methods. The evaluation performed on 4-core, 8-core, and 16-core MPSoCs, using PARSEC benchmarks, reveals a considerable reduction in thermal stress. For the 8-core MPSoC case study, on average, for the proposed heuristic(optimal) approach, the mean time to failure improved by 47(35) % compared to the state-of-the-art techniques with only 6(4) % performance degradation. Also, our simulations show that TheSPoT is more efficient in thermal stress reduction when more heterogeneous workloads are used.**

I. Introduction

Multiprocessor Systems-on-Chip (MPSoCs) play a major role in modern computational systems due to their higher performance [1]. However, the increase in the speed of these systems is accompanied by higher power consumption, more power density and frequent hot spots, if proper power control measures are not taken. Moreover, the availability of more resources in comparison with uniprocessors leads to more non-uniformity of the temperature profile. These spatial thermal gradients across the chip deteriorate system reliability and degrade its performance [2]. Also, the variety of the workloads, which could be processed at the same time, may cause large temporal temperature variations at a single point on the chip [1]. As a result, temporal temperature gradients and thermal cycles incorporate in degrading the performance and reliability of the modern MPSoCs [2].

Despite the importance of thermal variation in performance and reliability of MPSoCs, most power and thermal management techniques solely aim at power consumption/peak temperature reduction regardless of what adverse impacts their policies could have on the lifetime reliability of the target MPSoC. Several power management techniques including Dynamic Voltage Scaling (DVS) [3] and task allocation and scheduling [4] help reducing the chip average temperature by lowering the average power consumption. Although these approaches reduce hard failures corresponding to Time-Dependent-Dielectric-Breakdown and Electromigration [5], they do not take into account thermal stress as a dominant factor in reliability of the modern MPSoCs [6].

The study performed in [7] reveals well that the increase in the amount of power saving, which is usually followed by peak/average temperature reduction, improves the mean time to failure (MTTF) by reducing the Electromigration and time-dependent dielectric breakdown occurrences, while causes the overall system MTTF to fall down, since the MTTF related to thermal cycling decreases faster. Particularly, DPM (dynamic power management) and DTM (dynamic thermal management) approaches usually utilize DVFS, thread migration, and clock gating [8] to decrease the total power consumption and peak/average temperature. However, such techniques cause temperature variations not only more frequently but also with higher amplitudes, hence, reducing the system reliability. As a result, a comprehensive approach which considers thermal stress, power consumption, peak temperature, and performance objectives altogether, is vital.

In this work, we present TheSPoT, a multi-level thermal stress-aware power and temperature management approach. TheSPoT suits High Performance Computing (HPC) applications on MPSoCs. As a starting point, the variation-aware power/thermal management (VPTM) framework we introduced in [9] and adapted in [10] for thermal cycling-awareness is considered by which we develop our novel algorithms to alleviate thermal stress.

Overall, the contributions of this work compared with our previous work [10] may be briefly stated as follows:

1) considering the spatial thermal gradient (STG) in the DVFS convex optimization formulation,
2) proposing a fast heuristic algorithm for determining the near-optimal frequencies of the cores,
3) validating the scalability when the number of cores increases,
4) validating the efficiency of the proposed methods when confronting large workload variations.

The rest of the paper is organized as follows. Section II, reviews the background concepts of the paper and related works. Our power and thermal management framework, TheSPoT, is introduced in Section III. Section IV presents the proposed consolidation and deconsolidation algorithms. We present the proposed optimal and heuristic approaches, in detail, in Sections V and VI, respectively. The experimental setup and results are explained in Section VII. Finally, the paper is concluded in Section VIII.

II. Background Concepts and Related Work

A. Thermal gradients and thermal cycling

Thermal stress influences system reliability and, in particular, determines the MTTF at moderate temperatures [11]. Thus, reducing the thermal hot spots is not solely enough to achieve comprehensive thermal management for MPSoCs. In this work, any rapid temperature change, in either time or space, is regarded as a kind of thermal stress mechanism.

Temporal Temperature Gradient (TTG) is the rate of temperature changes over time. For a given time, the rate of the temperature changes from one point to another indicates the spatial temperature gradient (STG). Both STG and TTG pose a critical impact on the system lifetime reliability [5][12]. However, when speaking about STG, power and thermal management techniques must be applied regarding current status of more than one core. In contrast, TTG is more affected by the core frequency and its workload.

Thermal cycling phenomenon is another important thermal stress mechanism. By definition, when the temperature rises up (drops down) and goes back to the initial value a thermal cycle occurs [13] and it can be counted by Dowining simple rainflow-counting algorithm proposed in [14]. The expansion coefficient mismatch between the layers results in thermomechanical stresses leading to several failure mechanisms such as dielectric/thin film cracking, fractured bond wire, solder fatigue, and cracked die [15]. Thermal cycling (TC) tends to reduce the whole system MTTF as the number of cycles or amplitudes increases. Large amplitudes are normally induced due to improper task scheduling on a single core. Number of thermal cycles increases especially by the power management techniques which frequently turn cores on and off [5].

The number of cycles that can result in the occurrence of the failure due to the i^{th} thermal cycle is obtained from the modified Coffin-Manson equation as [13]

$$N_{TC}(i) = A_{TC}(\delta T_i - T_{th})^{-b}\exp(E_{aTc}/KT_{max_i}) \qquad (1)$$

where δT_i is the maximum thermal amplitude change of the i^{th} thermal cycle, T_{th} is the threshold temperature at which inelastic deformation begins, b is the Coffin-Manson exponent constant, E_{aTC} is the activation energy, T_{max_i} is the maximum temperature in the i^{th} cycle, and A_{TC} is an empirically determined constant [13]. The MTTF related to thermal cycling can be obtained by:

$$MTTF_{TC} = \frac{N_{TC}\Sigma_{i=0}^{m} t_i}{m} \qquad (2)$$

where m is the total number of cycles. For metallic structures, when δT increases from10˚C to 20˚C, the lifetime reliability may decrease up to 16 times [5].

B. Power and thermal management

Power and thermal management of MPSoCs is quite rich in previous works. When power consumption started to become one of the most significant issues of MPSoCs, researchers simply focused on power management policies through which peak temperature could also be controlled [18]- [22].

Although power management approaches could, to some extent, alleviate the thermal hot spots across the chip, increasing power density of MPSoCs made bare power management insufficient to deal with hot spots and led authors to propose thermal management policies at both design [23]-[25] and run time [26]-[31]. In particular, [23] and [24] propose optimal solutions for task scheduling and processor speed, respectively, [25] maximizes the performance of a periodic application, and [26] presents thermal balancing policy. Speedup of multicore processors under thermal constraints is determined in [27]. An OS-level technique for job scheduling is proposed in [28]. Thermal aging is addressed in [29]. The thermal impacts of the adjacent cores on the thermal profile is considered in [30]. Authors in [31] propose a dynamic thermal and power management using temperature prediction methodology. All these works, however, fail to consider thermal stress as a new dominant factor in modern MPSoCs lifetime reliability [6].

C. Thermal stress-aware power management

Considering thermal stress, as an important factor of MPSoCs reliability, in power and thermal management even increases the complexity of the management due to the contradictory behavior of peak temperature reduction techniques with thermal stress reduction approaches. Even though there are several works considering thermal stress, they rarely provide a comprehensive solution to cope with all thermal stress mechanisms along with power constraints. For instance, although in [32] the tradeoffs between temporal and spatial thermal gradient mitigation schemes were investigated, power management and thermal cycling were not considered. In addition, [33] proposes a new task scheduling method for reducing the temporal temperature gradient. Nonetheless, it does not consider thermal cycling and spatial gradient.

In what follows, we review the main works which address the direct reduction of thermal cycling of MPSoCs. The work in [1] describes an online task assignment and scheduling technique for maximizing the lifetime reliability of MPSoCs based on heterogeneous architectures. In [36], the authors propose a steady state temperature-aware task mapping and scheduling on a heterogeneous multicore architecture by considering the thermal cycle effect. An online learning method, using a multivariate loss function which considers hot spots, thermal cycles, spatial gradients, and average load altogether for the temperature management, is proposed in [5]. In [34], a hierarchical controller based on an aging sensor for improving the performance of homogeneous MPSoC architecture has been proposed. However, these works do not consider power and/or performance either as an objective or a design constraint.

Both static and dynamic methods are employed by [12] to reduce the hot spots, spatial gradients and thermal cycles. In the static strategy, an integer linear programming scheduling method optimizes the power and temperature subject to the performance constraint. The optimization is based on balancing the thermal hot spots and suppressing the temperature variation without being concerned about the spatial gradient. In the dynamic method, a heuristic algorithm allocates ready jobs to the coolest processor with idle neighbors. Also, in [12], the Adaptive-Random [6] technique is used to consider the temperature histories of the cores as well as their current temperatures. In this work, the proposed consolidation policy does not consider the adverse effect of thermal cycle.

Machine learning is leveraged by [35] and [38] for thermal cycle reduction. Although in [38] authors consider all thermal

stress mechanisms, the efficiency of their Q-learning-based approach has not been evaluated for rapid workload variations.

Finally, our previous work [10] proposes a convex optimization solution and uses both consolidation/ deconsolidation and DVFS for reducing thermal stress. However, the formulated convex optimization problem for DVFS does not consider spatial thermal gradients and the runtime overhead is a major concern.

D. Where TheSPoT stands

Modern MPSoCs are equipped with several power/thermal management knobs. In particular, Intel is leveraging DVFS, P-states, and C-states to optimize the performance considering thermal/power constraints [16]. In addition, memory throttling has been proposed for Intel's multicore processors [17]. Along with industry, academia seeks for more energy-efficient performance optimization solutions, employing the available control knobs. Nonetheless, a holistic approach to deal with power/thermal management, performance optimization, and lifetime reliability, including all thermal stress mechanisms has not yet been achieved.

In this paper we propose a methodology for a comprehensive thermal stress-aware power management of MPSoCs. Although in this work we implement TheSPoT on software, it could also be implemented on hardware. In this context, the main difference between our approach and those proposed recently by AMD [39] and Intel [40] is that TheSPoT leverages thermal stress-aware algorithms to further improve the MTTF of the system.

III. THESPOT: THERMAL STRESS-AWARE POWER/TEMPERATURE MANAGEMENT FRAMEWORK

In this section, we propose TheSPoT, shown in Fig. 1. TheSPoT is an improved thermal stress-aware power/thermal management framework which employs various controlling knobs, including DVFS, core consolidation/deconsolidation and thread migration. As a starting point, we consider VPTM which is a hierarchical dynamic power/thermal management framework for heterogonous MPSoCs [9], and modify it in order to make it applicable for thermal stress reduction. TheSPoT, similar to VPTM, contains a workload analyzer providing the IPS (instruction per second) of the running application by applying a moving average calculation. In contrast to VPTM, our framework includes Tier1 and Tier2 modules modified for thermal stress-aware power and temperature management.

Tier1 and Tier2 modules are called at the beginning of their corresponding decision epochs. Tier1 performs the core consolidation/deconsolidation and avoids thermal emergencies while it reduces both spatial and temporal thermal gradients. Tier2 is in charge of determining the most appropriate frequencies of the cores in order to satisfy power budget, peak temperature and thermal stress constraints while considering performance as a primary objective.

In particular, Tier1 receives the predicted IPC (instruction per cycle) values provided by the workload analyzer, reads the current per-core power and temperature, and is aware of power budget and peak temperature constraints. Then, Tier1 delivers the IDs of running cores to Tier2 after having performed the consolidation/deconsolidation according to the thermal stress

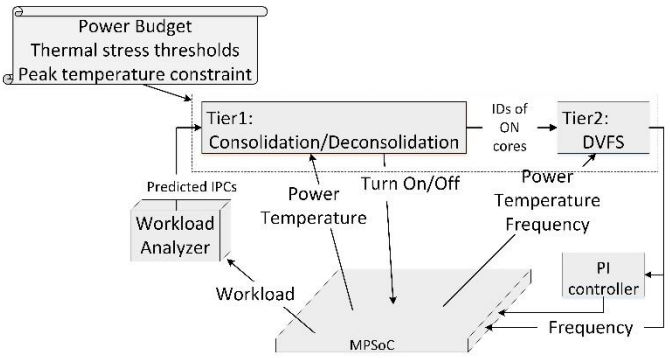


Fig. 1 TheSPoT framework

considerations. Afterwards, Tier2, which is aware of the per-core current operating frequency, power and temperature, recalculates the most appropriate frequencies of the cores to satisfy thermal stress, power, and peak temperature constraints.

While the algorithm used for Tier1 is consistent through this work, we propose two different algorithms for the DVFS of Tier2. First, the optimal frequencies and voltages of the cores are determined by solving a convex optimization problem. Thereafter, in the second algorithm, we employ a heuristic algorithm to avoid high runtime overhead of the convex optimization solution.

On one hand, in the proposed convex optimization approach, the performance objective (IPS, which is directly dependent on the frequency) is followed by power, peak temperature, and thermal gradient ($\nabla\theta$) constraints. In this formulation, the power and peak temperature constraints are fixed constraints, while $\nabla\theta$ is dynamically changing at runtime based on the temperature history to provide more opportunities for performance enhancement. The thermal gradient constraint includes both spatial and temporal thermal gradients in this formulation.

On the other hand, the same objective and constraints are defined in the proposed heuristic approach. By considering a boundary around the thermal stress thresholds more opportunities are provided to increase the performance. This is similar to the approach taken throughout our proposed convex optimization approach. After following the guidelines introduced in Section VI.B, the maximum possible frequency that satisfies the thermal gradient constraints is determined. However, given this frequency the power and temperature constraints must be satisfied. If not, the frequency is reduced until these constraints (power and temperature) are met.

Finally, a closed-loop proportional-integral (PI) controller, based on actual measurements, modifies the decisions taken by Tier2 and fine-tunes the core DVFS settings at runtime [9]. It makes the power and thermal management robust to workload variations and addresses the overestimation/underestimation caused by the DVFS technique, similar to AVFS proposed by AMD [39]. TABLE I presents the notation used in this paper.

IV. TIER1: CONSOLIDATION AND DECONSOLIDATION

A. Consolidation

For consolidation, first, a tuple of (i,j) cores (corresponding to the source and destination cores) are selected. The i^{th} core is selected if its IPS is smaller than a predefined constant

TABLE I OVERVIEW OF THE USED NOTATION

$IPS_i, IPS_{const,i}$	Current IPS on the i[th] core, and its constraint
$Cost_{TempDiff,k}$	Temperature difference between cores of the k[th] tuple
$Cost_{Migration,k}$	Migration cost for the k[th] tuple
f_i	Frequency of the i[th] core
$\theta_i(t)$	Temperature of the i[th] core
$P_i(t)$	Power of the i[th] core
$P_i(f,\theta)$	Power as a function of frequency and temperature
θ_{const}	Temperature constraint
f_{min}, f_{max}	Minimum and maximum frequency of by the i[th] core
$\nabla\theta_{INC_i}, \nabla\theta_{DEC_i}$	Increase and decrease constraints for thermal gradient
θ_v, θ_p	Valley and peak temperatures
$\Delta\theta_{Th}$	Temperature difference threshold
$\nabla\theta_{max}$	Maximum allowable thermal gradient
$V_{min,i}, V_{max,i}$	Minimum and maximum voltage of by the i[th] core
TTG_{th}, STG_{th}	Temporal and spatial thermal gradient thresholds
$TTG_i(t)$	Temporal thermal gradient of the i[th] core
$STG^{i,j}(t)$	Spatial thermal gradient of the i[th] core

value ($IPS_{CONST,i}$), and the cost of its thread migration to the j[th] core is smaller than $Cost_{Migration}$. The j[th] core is selected if the consolidation of its thread and the threads of the i[th] core does not lead to an IPS which is more than the maximum IPS allowed for the j[th] core.

Next, for each tuple, the difference between the maximum and the minimum temperatures of the chip is estimated assuming that the consolidation is performed and the i[th] core is turned off. Therefore, a power of zero for the i[th] core is assumed while the power of the j[th] core is elevated by assuming that the IPS_j^{new} is equal to summation of IPS values before consolidation, *i.e.*, $IPS_j + IPS_i$. In particular, the power of the destination core is estimated from the power model of [9] as:

$$P(f,\theta) = d.f^{\beta} + l.f + k_{\theta}.\theta \quad (3)$$

where f and θ are the core frequency and the temperature, d, l and k_θ are empirical coefficients for dynamic power consumption, temperature-independent and temperature-dependent components of leakage power dissipation, respectively, and β has a value between 2 and 3. In this paper, for power and temperature models we use the same methodology as in [9].

The frequency of the j[th] core is increased such that the core can handle the IPS value required after the consolidation. Therefore, the frequency is obtained from:

$$f_j^{new} = \frac{IPS_i + IPS_j}{IPS_j} \times f_j \quad (4)$$

where f_j^{new} (f_j) is the frequency of the j[th] core after (before) the consolidation. This frequency calculation is used in power and thermal models. On the other hand, the target platform provides some discrete frequency values to which this calculated value should be mapped. Therefore, our methodology can tolerate inaccuracies in frequency calculation. As a consequence, although this rough frequency calculation does not take into account the IPS change when a thread migrates from a core to another in case of heterogeneity, our formulation still is valid.

Algorithm 1. Pseudo-code of proposed optimal DVFS

```
1: if (Tier2 Decision Time)
2:     Calculate STG for all pairs of adjacent cores
3: for each core
4:     Determine thermal constraints based on the importance of STG
5:     Formulate the convex optimization problem
6:     Solve the convex optimization problem
7: for each core
8:     Apply frequencies
```

Now, based on the relation between the temperature and the power, $\theta(t+\Delta t)$ values for all the units of the MPSoC are obtained from [41]:

$$\boldsymbol{\theta}(t+\Delta t) = \mathbf{A}.\boldsymbol{\theta}(t) + \mathbf{B}.P(t) \quad (5)$$

where $\mathbf{A}$ and $\mathbf{B}$ are $n \times n$ (n is equal to the number of units of the MPSoC) coefficient matrices. These matrices are dependent on the floorplan and technology and are extracted using Hotspot [42]. $\boldsymbol{\theta}(t+\Delta t)$ is an $n \times 1$ matrix whose i[th] row contains the temperature of the i[th] unit. We use (3) and (5) in a loop to model the positive feedback between leakage power and temperature.

After estimating the temperatures of the units, the temperature difference between the coolest and the hottest units of the cores is considered as the temperature cost ($Cost_{Temp}$) of the tuple. Finally, by using a merit function, the tuple with the smallest cost is selected:

$$M_k = Cost_{Migration,k} + Cost_{TempDiff,k} \quad (6)$$

Three cost types are defined for thread migration: a fixed cost to transfer a few kilobytes of architectural state to the other core, a cost of draining and refilling the pipeline, and warmup cost for caches [43]. The last two costs are extracted from the sniper simulator [44] itself, while for the first one, in this work, we consider 300 cycles, following the cost model proposed in [45]. In (6), for the first term, we normalize the migration cost to the maximum value obtained in the iteration. Similarly, the latter is normalized to the maximum temperature difference between the cores in that iteration.

B. Deconsolidation

The core deconsolidation may be performed under two cases. In the first case, the temperature of a core reaches a value higher than the temperature constraint (θ_{Const}) while its frequency is equal to its minimum value (f_{min}). In this case, if the core has more than one thread, one thread is chosen to be migrated to another core, instead of turning off the core as has been done in the approach invoked in [9]. This helps decreasing the temporal thermal gradients of the core. In the second case, the frequency of the core is at its maximum value and the core contains more than one thread. Here, the thread with the highest IPS from the core is selected to be migrated to another core. This leads to the performance increase of the source core. In both cases, the destination core for the selected thread is chosen based on the same method used in consolidation.

V. TIER2: PROPOSED OPTIMAL DVFS WITH PERFORMANCE OBJECTIVE

In this work, we modify the convex optimization formulation of [10] in Tier2 to include spatial thermal gradients. The overall approach for applying the core frequencies is shown in Algorithm 1.

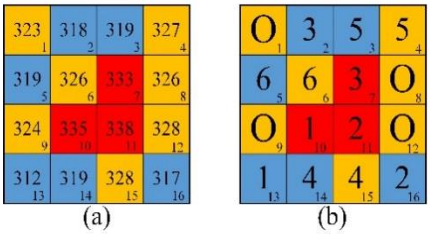


Fig. 2 a) Average core temperature (K), b) Numbering the core pair under spatial stress based on the algorithm

A. *Determining the Spatial Temperature Gradient*

The constraints used for performing the DVFS are determined based on the existence of the spatial gradient in contrast to [10]. Thus, we define the spatial thermal gradient as the absolute value of the temperature difference between the two components divided by their corresponding distance measured from their centers. In this work, by using the tool ArchFP [46], the floorplan of the MPSoC consisting of the cores are determined. We use the center-to-center distance of the cores as the distance between them.

Fig. 2 shows a schematic of a 16-core MPSoC and illustrates the process of determining spatially stressed cores. In this process, after determining the STG of the adjacent cores, only the values above the STG threshold (STG_{th}) are considered.

In Fig. 2(b), the 10th and the 13th cores are numbered by 1 since they appeared to have the largest temperature difference regarding Fig. 2(a). The 11th and the 16th cores are numbered by 2 as they have the second largest temperature difference after excluding the first pair. Each core may be considered as a stressed core only with one another core. If there are more than one candidate, the two adjacent cores with the highest difference are chosen.

B. *Defining Thermal Stress Constraints*

In Tier2, to select the optimal frequency of each core, we have used the formulation proposed in [9] as the base for this tier. In addition to the maximum temperature and maximum power constraints, we suggest adding the temporal and spatial thermal gradient constraints. Hence, the increase and decrease rates of the temperature are limited to $\nabla\theta_{INC}$ and $\nabla\theta_{DEC}$, respectively as:

$$\frac{\theta(t+\Delta t)_i - \theta(t)_i}{\Delta t} < \nabla\theta_{INC_i} \quad (7\text{-}1)$$

$$\frac{\theta(t)_i - \theta(t+\Delta t)_i}{\Delta t} < \nabla\theta_{DEC_i} \quad (7\text{-}2)$$

where $\theta_i(t)$ is the current temperature of the i^{th} unit, Δt is the Tier2 epoch duration, and $\theta_i(t+\Delta t)$ is the temperature of that unit after Δt. Since $\theta(t+\Delta t)$ is a function of the frequency, this constraint sets the upper and lower bounds on the frequency change (through the bounds on the thermal variation) of the i^{th} unit in each Tier2 epoch.

In order to control the amplitude of the thermal cycle along with the temporal thermal gradient in the DVFS process, we propose to adjust the values of $\nabla\theta_{INC}$ and $\nabla\theta_{DEC}$ dynamically. The adjustment is performed based on the current temperature, the peak and valley temperatures of the unit up to this point (denoted by θ_P and θ_V, respectively) and a temperature difference threshold ($\Delta\theta_{Th}$). Moreover, the maximum of the absolute value of the temporal gradient is determined by by $\nabla\theta_{MAX}$.

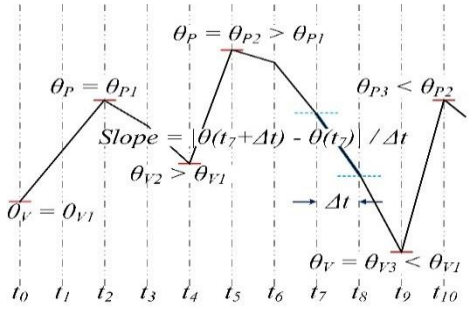


Fig. 3 Determination of the peak and valley temperatures as well as temporal temperature gradients (*Slope*). For the sake of simplicity, the transition at the beginning of each epoch has been neglected

Fig. 3 illustrates the way that the parameters θ_P and θ_V are determined. At the beginning, the valley temperature is equal to the first valley (θ_{V1}). However, the second valley is not considered as a new θ_V since it is not lower than the previous one. Later, θ_{V3} which is lower than the current valley, is considered as a new valley. A similar procedure is used for determining the peak temperature. This approach works based on minimizing the thermal cycle amplitude. In the proposed approach, only if the peak (valley) temperature becomes higher (lower), it should be considered in the algorithm for adjusting the frequency. This situation results in an opportunity to improve the performance by not limiting the temperature increase/decrease rate.

At the beginning of each Tier2 epoch, before solving the convex optimization problem, the temporal thermal gradient constraints are determined. Prior to the constraint formulation, adjacent cores are evaluated to determine whether they are bearing spatial thermal gradients more than a threshold value. If a core does not belong to any pair of the spatially stressed cores, the formulation explained next is used for the thermal gradient constraint determination. In this formulation, if the temperature of the i^{th} unit in the last Tier2 epoch duration has increased (i.e. positive slope), $\nabla\theta_{INC_i}$ and $\nabla\theta_{DEC_i}$ are defined based on the pseudo-code given in Algorithm 2.

In the pseudo-code of Algorithm 2, $\theta_{C,i}$, $\theta_{P,i}$, and $\theta_{V,i}$ represent the current, peak, and the valley temperatures of the i^{th} core and α is a predefined value between 0 and 1. In this formulation, the temperature increase rate is calculated based on the current temperature and the peak temperature up to the previous thermal cycle. This peak temperature is considered as the reference. If the current temperature exceeds θ_P, the increase rate is limited to $\alpha\nabla\theta_{MAX}$, i.e. the lowest allowed temporal thermal gradient constraint (line 1 and 2) in this algorithm. If the difference between the current and the peak temperature is more than $\Delta\theta_{Th}$ (line 3 and 4), the increase rate is set to its maximum value ($\nabla\theta_{MAX}$). Finally, when the current temperature becomes closer to the peak temperature, the temperature increase rate is reduced exponentially (line 6). However, the rate cannot be reduced to a value smaller than $\alpha\nabla\theta_{MAX}$.

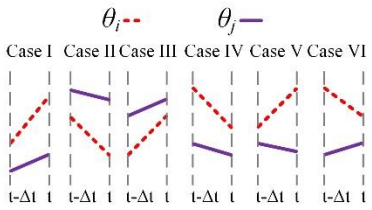


Fig. 4 Six cases for temperature trends of a pair of cores under spatial stress

Algorithm 2. Temporal thermal constraint when temperature is increasing

1: ***if*** $(\theta_{C,i} > \theta_{P,i})$ *//The increasing rate should be suppressed*
2: $\nabla\theta_{INC_i} = \alpha\nabla\theta_{MAX}$ *//0 < α < 1*
3: ***else if*** $(\theta_{P,i} - \theta_{C,i} > \Delta\theta_{Th})$ *//No suppression of temperature*
4: $\nabla\theta_{INC_i} = \nabla\theta_{MAX}$
5: ***else*** *//The increasing rate should be moderated*
6: $\nabla\theta_{INC_i} = \alpha\nabla\theta_{MAX} + ((1-\alpha)\nabla\theta_{MAX})\frac{e^{\frac{\theta_{P,i}-\theta_{C,i}}{\Delta\theta_{Th}}}-1}{e-1}$
7: ***if*** $(\theta_{C,i} < \theta_{V,i})$
8: $\nabla\theta_{DEC_i} = \alpha\nabla\theta_{MAX}$
9: ***else if*** $(\theta_{C,i} > 0.5(\theta_{P,i} - \theta_{V,i}) + \theta_{V,i})$
10: $\nabla\theta_{DEC_i} = \nabla\theta_{MAX}$
11: ***else***
12: $\nabla\theta_{DEC_i} = \alpha\nabla\theta_{MAX} + ((1-\alpha)\nabla\theta_{MAX})\frac{e^{\frac{\theta_{C,i}-\theta_{V,i}}{\Delta\theta_{Th}}}-1}{e^{0.5}-1}$

Moreover, if the slope of the temperature in the last epoch is positive, choosing a lower frequency in the decision time may help reducing the temperature. Hence, in addition to the temperature increase rate constraint, the decrease rate constraint ($\nabla\theta_{DEC}$) should be determined. The decrease rate constraint is calculated based on the current temperature and the valley of the previous thermal cycle (lines 7-12). In the case of increasing temperature in the current epoch, further increase is more probable than the temperature decrease. Hence, in our approach, the temperature increase rate constraint is defined more conservatively than the decrease rate constraint. Based on the study performed in this work, a small value (say, < 0.1) was found appropriate for α.

The above discussion was about the case when the temperature is increased in the last Tier2 epoch duration. In this case, the temperature decrease rate constraint ($\nabla\theta_{DEC}$) is almost similar to the increase rate constraint ($\nabla\theta_{INC}$) in the case of increasing temperature, and vice versa.

C. Including the Spatial Gradient in Defining Thermal Stress Constraints

When the spatial thermal gradient for a pair of cores is large enough, the cores may be selected as a pair under spatial stress. In this case, the above thermal gradient constraints are formulated differently for both cores. The goal, here, is to modify the constraints given by Algorithm 2 to make it more probable for the temperatures of the i^{th} and the j^{th} cores (θ_i and θ_j, respectively) to become closer to each other in the next epoch.

Six cases can occur when the STG value of the two cores needs attention as shown in Fig. 4. In Case I, in order to lower the STG, the increase rate constraint of the core with the higher change rate (say, the i^{th} core) should be smaller than that of the other core (say, the j^{th} core). Thus, the increase rate constraint for the i^{th} and the j^{th} cores are modified as shown in Algorithm 3, where $\theta_{L,j}$ is the temperature of the j^{th} core measured in the last decision time. The decrease rate constraints for both cores are obtained from Algorithm 2 due to the STG unimportance.

In Case II, the increase rate constraints of both cores are determined by the algorithms introduced in the previous subsection (due to STG unimportance) while the decrease rates are obtained from Algorithm 4.

Algorithm 3. Increase rate constraint related to Case I

1: $\nabla\theta_{INC_i} = \alpha\nabla\theta_{MAX}$
2: ***if*** $(\theta_{P,j} - \theta_{C,j} > \Delta\theta_{Th})$
3: $\nabla\theta_{INC_j} = \nabla\theta_{MAX}$
4: ***else***
5: $\nabla\theta_{INC_j} = \alpha\nabla\theta_{MAX} + ((1-\alpha)\nabla\theta_{MAX})\frac{e^{\frac{\theta_{C,j}-\theta_{L,j}}{\Delta\theta_{Th}}}-1}{e-1}$

Algorithm 4. Decrease rate constraint related to Case II

1: $\nabla\theta_{DEC_i} = \alpha\nabla\theta_{MAX}$
2: ***if*** $(\theta_{C,j} - \theta_{V,j} > \Delta\theta_{Th})$
3: $\nabla\theta_{DEC_j} = \nabla\theta_{MAX}$
4: ***else***
5: $\nabla\theta_{DEC,Const_j} = \alpha\nabla\theta_{MAX} + ((1-\alpha)\nabla\theta_{MAX})\frac{e^{\frac{\theta_{L,j}-\theta_{C,j}}{\Delta\theta_{Th}}}-1}{e-1}$

For Case III and Case IV, shown in Fig. 4, the temperature change behaviors are such that the STG problem is lessened as time passes. Hence, we can use the constraints given for the cores with no spatial gradient.

In Case V, θ_i and θ_j are diverging. To achieve a smaller spatial thermal gradient in the next epoch, both $\nabla\theta_{INC_i}$ and $\nabla\theta_{DEC_j}$ should be limited to the lowest temporal gradient constraint to lower the temperature difference between the two cores. This case is the worst one among the others considered here. Thus, $\nabla\theta_{INC_i}$ and $\nabla\theta_{DEC_j}$ are given by:

$$\nabla\theta_{INC_i} = \alpha\nabla\theta_{MAX}$$
$$\nabla\theta_{DEC_j} = \alpha\nabla\theta_{MAX} \quad (8)$$

while $\nabla\theta_{DEC_i}$ and $\nabla\theta_{INC_j}$ are unchanged.

In Case VI, where the STG is decreasing, both $\nabla\theta_{DEC_i}$ and $\nabla\theta_{INC_j}$ need to be modified moderately. Consequently, $\nabla\theta_{DEC_i}$ and $\nabla\theta_{INC_j}$ are expressed as:

$$\nabla\theta_{DEC_i} = \alpha\nabla\theta_{MAX} + ((1-\alpha)\nabla\theta_{MAX})\frac{e^{\frac{\theta_{L,i}-\theta_{C,i}}{\Delta\theta_{Th}}}-1}{e-1}$$
$$\nabla\theta_{INC_j} = \alpha\nabla\theta_{MAX} + ((1-\alpha)\nabla\theta_{MAX})\frac{e^{\frac{\theta_{C,j}-\theta_{L,j}}{\Delta\theta_{Th}}}-1}{e-1} \quad (9)$$

D. Convex Optimization Problem

Having obtained the thermal stress constraints, we form a convex optimization problem including power and thermal constraints, and the frequency domain by:

$$Maximize \sum_{i=1}^{|Cores|} IPS_i X_i \quad (10)$$

Subject to:

$$A.\theta + B.P < \theta_{Const}$$

$$P < P_{budget}$$
$$f_{MIN} < f < f_{MAX}$$
$$P = D.f^{\beta} + L.f + K_{\theta}.\theta$$
$$\frac{1}{\Delta t}\left((A.\theta(t) + B.P) - \theta(t)\right) < \nabla\theta_{INC_i}$$
$$\frac{1}{\Delta t}\left(\theta(t) - (A.\theta(t) + B.P)\right) < \nabla\theta_{DEC_i}$$

where X_i is a binary variable which is 1 if the i^{th} core is active (*i.e.*, ON) and 0 if the core is OFF. The proposed formulation leads to optimal solution where all active cores operate at the maximum possible frequency under all thermal and power constraints.

After determining the frequency of the i^{th} core, its corresponding voltage ($V_{supply,i}$) is also calculated using:

$$V_{supply,i} = V_{min,i} + \left(V_{max,i} - V_{min,i}\right) \times \frac{f_i - f_{min,i}}{f_{max,i} - f_{min,i}} \qquad (11)$$

where $V_{min,i}$, $V_{max,i}$, $f_{min,i}$, and $f_{max,i}$ show the minimum supply voltage, maximum supply voltage, minimum frequency and maximum frequency of the i^{th} core, respectively.

VI. TIER2: PROPOSED LOW-COMPLEXITY HEURISTIC DVFS

Although the proposed DVFS approach brings about the optimal frequencies for the power and thermal management problem constrained by thermal stress, it may fail to deal with real-time application due to a large runtime overhead. Fig. 5 shows runtime overhead for *facesim* benchmark of the proposed optimal solution.

As the number of cores increases, the runtime overhead rises super-linearly and makes this solution infeasible for MPSoCs. Hence, due to the large computational overhead of the optimal solutions, we should focus on algorithms which are fast and find the near-optimal solution.

To reduce the runtime overhead of the optimal solution in real applications, we propose a new heuristic DVFS algorithm in Tier2. The flowchart of the algorithm is shown in Fig. 6. The proposed heuristic DVFS considers the thermal stress constraint and available power and temperature budgets and has the objective of increasing the frequency (and the performance) as much as possible.

In this algorithm, first, the temporal and spatial thermal gradient (TTG and STG) of each core are calculated. Then, the cores are classified based on the values of $STG(t)$, $STG(t-\Delta t)$, $TTG(t)$, and $TTG(t-\Delta t)$ of each core to decide on the existence of a kind of thermal stress for the core. In this notation, t and $t-\Delta t$ correspond to the current and last time epochs, respectively.

Here, predefined threshold values, TTG_{th} and STG_{th}, are used for the core classification. The classification includes *Temporally Stressed*, *Spatially Stressed*, *Temporally & Spatially Stressed*, and *Relaxed* which are discussed in detail in subsection A. Based on the assigned class and the trend of the core temperature variation, the frequency of each core is determined. Before applying the calculated frequencies, the temperature and power consumption of each core in the next epoch are predicted to check whether the power and/or temperature constraints are not violated.

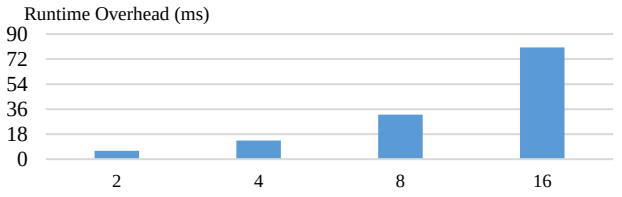


Fig. 5 Runtime overhead of the optimization solution for different number of cores

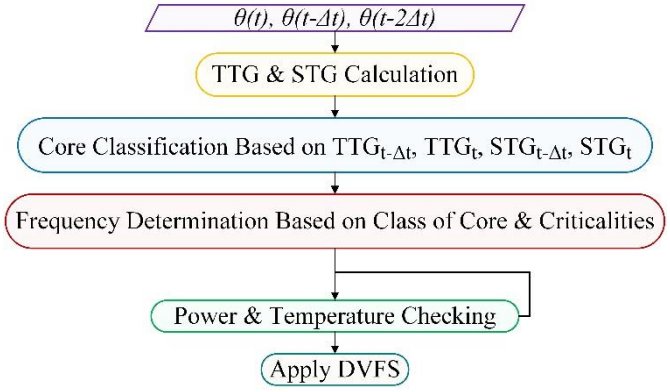


Fig. 6 The proposed flowchart of the heuristic DVFS algorithm in Tier2

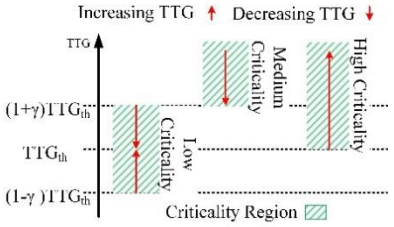


Fig. 7 Regions for different TTG criticality level

A. Core Classification

First, the class of each core based on the temporal and spatial gradients measured in the current and previous time epochs is determined.

a) Temporally Stressed Cores. We classify the stressed cores based on the following criteria:

- The lowest criticality ($\boldsymbol{C_{L,TTG}}$): $TTG_{th} > TTG(t) > (1-\gamma)TTG_{th}$ and $TTG(t) > TTG(t-\Delta t)$ (increasing gradient) or $TTG_{th} < TTG(t) < (1+\gamma)TTG_{th}$ and $TTG(t) < TTG(t-\Delta t)$ (decreasing gradient),
- The medium criticality ($\boldsymbol{C_{M,TTG}}$): $(1+\gamma)TTG_{th} < TTG(t) < TTG_{th}(t-\Delta t)$, and
- The highest criticality ($\boldsymbol{C_{H,TTG}}$): $TTG(t) > TTG_{th}$ and $TTG(t) > TTG(t-\Delta t)$.

Here, γ is a coefficient between 0 and 1 which is determined through 10 simulations with small inputs. Fig. 7 illustrates the regions corresponding to each criticality level considering the trends of temperature change.

b) Spatially Stressed Cores. The spatial gradient is defined based on the temperature variation of two neighbor

cores and, hence, the spatial stress is considered only for pairs of the cores. Here, again, we define the coefficient λ between 0 and 1 obtained from simulations. Also, similar to the previous case, we consider three levels of criticality:

- The lowest criticality ($\boldsymbol{C_{L,STG}}$): $(1-\lambda)STG_{th} < STG^{i,j}(t) \leq STG_{th}$ provided that $STG^{i,j}(t) > STG^{i,j}(t-\Delta t)$ or $STG_{th} < STG^{i,j}(t) < (1+\lambda)STG_{th}$ provided that $STG^{i,j}(t) < STG^{i,j}(t-\Delta t)$,
- The medium criticality ($\boldsymbol{C_{M,STG}}$): $(1+\lambda)STG_{th} < STG^{i,j}(t) < STG^{i,j}(t-\Delta t)$, and
- The highest criticality level ($\boldsymbol{C_{H,STG}}$): $STG^{i,j}(t) > STG_{th}$ and $STG^{i,j}(t) > STG^{i,j}(t-\Delta t)$.

Here, $STG^{i,j}(t)$ is the current spatial gradient for the pair of the i^{th} and j^{th} cores. The regions for the STG criticality level may be demonstrated by replacing TTG by STG and γ by λ in Fig. 7.

c) Temporally and Spatially Stressed cores. If there is a pair of spatially stressed cores, which contains at least one core under temporal stress, the whole pair is classified as *Temporally & Spatially Stressed* class.

d) Relaxed cores. When a core is not under any kind of stress, it is classified as a *Relaxed* core.

B. Frequency Determination

To perform an appropriate DVFS scheme providing alleviation of the thermal stress and higher performance, the following guidelines are considered:

G1: To reduce temporal gradient, the frequency needs to be changed in a way to oppose the direction of current temperature trend.

G2: The amount of decrease or increase in the frequency of a core must be a strong function of its stress type and criticality.

G3: Since obtaining a higher performance is the main goal, reducing the thermal gradient is preferred to be solved by increasing the frequency rather than decreasing it.

G4: Since in Case V (Fig. 4) the STG worsens more quickly than the other cases, the frequency should change more.

G5: When a core is both spatially and temporally stressed, alleviating STG and TTG can be achieved through exploitation of tradeoffs between spatial and temporal gradients.

G6: Excessive change of the frequency may either turn a relaxing core into a stressed one, or adversely affect the other stress type, or cause thermal stress in the opposite direction.

Algorithm 5 describes the frequency change applied when a core is under only temporal thermal gradients. In this pseudo code, $CF_{level,TTG}$ presents the change in frequency based on the criticality level of the TTG, and $CF_{performance}$ is the change in frequency to obtain higher performance (G3).

Algorithm 6 shows the pseudo code used for spatially stressed cores where $CF_{level,STG}$ represents the change in frequency based on the criticality level of the STG.

Based on G5, the application of the DVFS scheme by considering $C_{STG}^{i,j}$, C_{TTG}^{i}, and C_{TTG}^{j}, may require some compromise. First, we note that when the TTG value of a core is more than that of the other one, it does not necessarily mean that its TTG-related criticality level is also higher (see Fig. 7). Also, for a pair of cores classified as *Temporally & Spatially Stressed*, there may be only one temporally stressed core and the TTG criticality for the other is considered to be zero.

Algorithm 5.

```
for each core under TTG
    if frequency decrease required
        use CF_{level,TTG}
    else
        use (CF_{level,TTG} + CF_{performance})
```

Algorithm 6.

```
for each pair of cores under STG
    for each core in the pair
        if frequency decrease required
            if temperature ascending
                use (CF_{level,STG} − CF_{performance})
            else
                do not change the frequency
        else
            if temperature ascending
                CF_{performance}
            else
                use (CF_{level,STG} + CF_{performance})
```

Algorithm 7.

```
if Case I or II of Fig. 4
    if C^i_TTG > C^j_TTG:
        f_i changes max(CF^i_{level,TTG}, CF^i_{level,STG})
        f_j changes CF^j_{level,TTG}
    else if C^i_TTG == C^j_TTG:
        f_i changes (max(CF^i_{level,TTG}, CF^i_{level,STG}) + 1)
        f_j changes CF^j_{level,TTG}
    else
        f_i changes (CF^i_{level,TTG} + CF^{i,j}_{level,STG})
        f_j changes CF^j_{level,TTG}
if Case III or IV of Fig. 4
    if C^i_TTG ≥ C^j_TTG:
        f_i changes (CF^i_{level,TTG} − 1)
        f_j changes CF^i_{level,TTG}
    else
        f_i changes CF^i_{level,TTG}
        f_j changes max(CF^j_{level,TTG}, CF^{i,j}_{level,STG})
if Case V of Fig. 4
    f_i changes CF^i_TTG
    f_j changes (max(CF^j_{level,TTG}, CF^{i,j}_{level,STG}))
if Case VI of Fig. 4
    f_i changes (CF^i_{level,TTG} − 1)
    f_j changes (CF^j_{level,TTG} − 1)
```

Algorithm 7 shows the proposed pseudo code for DVFS of the cores under both spatial and temporal thermal gradients. The most appropriate DVFS settings are those that consider the pseudo codes of Algorithm 5 and Algorithm 6, as the basic rules, at the same time, and provide tradeoffs wherever the frequency changes suggested by these two pseudo codes do not agree with each other. Algorithm 7 determines proper frequency changes to simultaneously consider cores under TTG and STG. The term "changes" is replaced by "decreases" or "increases" based on the appropriate change suggested by Algorithm 5. Also, C_{TTG}^{i} denotes the criticality level of TTG for the i^{th} core.

When a core is *Relaxed*, its frequency could be increased to achieve a higher performance. Since an excessive increase in the frequency leads to a thermal stress (G6), the process should

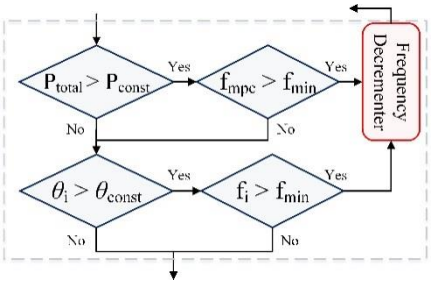


Fig. 8 Flowchart of *Power and Temperature Checking*

be performed carefully. For this reason, when the TTG is (is not) positive, the frequency of the relaxed core is increased by two (three) steps. Note that these numbers were obtained for our simulations where the frequency range was divided by 15 to determine the frequency steps for the MPSoC.

Finally, it should be mentioned that in this work 4 (4), 3 (3), and 2(2) were considered for $CF_{H,TTG}(CF_{H,STG})$, $CF_{M,TTG}(CF_{M,STG})$ and $CF_{L,TTG}(CF_{L,STG})$, respectively.

C. Power and Temperature Checking

Before applying the frequencies obtained from Section VI.B, the temperature and power consumption of the cores in the next epoch are predicted. First, based on the model given by (5) which depends only on the current temperature and power consumption, the next temperature of each core is calculated. Then, using the new temperature and frequency, the total power consumption is obtained based on (3). Afterwards, the total power consumption (P_{total}) is compared with the power constraint (P_{const}) (see Fig. 8). If P_{total} is larger than P_{const}, the frequency of the most power consuming core (f_{mpc}) is lowered one step. This procedure continues until P_{total} becomes lower than P_{const} or f_{mpc} equals to the minimum frequency.

After considering the total power consumption, the temperature of each core is predicted. If the temperature of any core exceeds the θ_{const}, its pre-assigned frequency (f_i) is decreased one step. This procedure lasts till the predicted temperatures of all the cores become lower than θ_{const}, or f_i is no longer greater than $f_{min,i}$.

In the situation where the temperature and power constraints could not be met by the frequency reduction, the control of the algorithm is transferred to Tier1 which can invoke consolidation/deconsolidation procedure. It is preferred to satisfy θ_{const} and P_{const} in Tier2 rather than in Tier1 since the consolidation procedure may lead to turning a core off which reduces the performance compared with the case when the core is running even (with the minimum frequency).

VII. Experimental Setup and Results

We have studied the efficiency of TheSPoT in tackling thermal cycling and thermal gradient issues of MPSoCs using the PARSEC [47] benchmarks package. For comparison, we implemented the dynamic power/thermal management approach proposed by [12] which employs DVFS (including $f_{min,i}$ and $f_{max,i}$) and thread migration. In addition, to demonstrate the scalability of the proposed power and temperature management techniques, 4-core, 8-core, and 16-core MPSoCs were considered.

TABLE II. Thermal Values

TTG_{th}	STG_{th}	$\theta_{ambient}$	$\theta_{constraint}$
0.80 K/ms	0.25 K/mm	310K	340K

A. Simulation framework and MPSoC architecture

The simulation framework was implemented in the Sniper multicore simulator [44]. The power consumption and the temperature of the MPSoC were estimated using McPAT [48] and Hotspot [42] tools, respectively. To extract the floorplan of the MPSoC, ArchFP tool [46] was exploited where the areas of different parts were extracted using McPAT based on a 45nm technology. TheSPoT and the power and thermal management algorithm proposed by [12] were implemented using Python programming language. Also, for the case of our optimal approach, the convex problem of Tier2 was solved using NLOPT tool [49]. This toolchain carefully takes into account any change in workload on any core and provides the corresponding performance, power and temperature values such that thermal gradients can be considered accurately.

Moreover, we relied on McPAT support for modeling the wake-up power and delay overheads. Finally, in order to consider DVFS overhead, we used a micro-architectural parameter provided by Sniper simulator and set it to $10\mu s$ [50].

In this work, for all simulations, Tier1 (Tier2) epoch duration was 10ms (5ms). For all simulation scenarios large inputs were considered. For a fair comparison, the approach of [12] is also used every 10ms. TABLE II shows the ambient temperature, temperature constraint, and the threshold values for TTG and STG. The temperature constraint is defined by the user and considered as the core critical temperature. The methodology presented in this work is valid for any threshold values, although improvements in thermal stress reduction and performance overhead may change. In addition, we have used the same threshold values for the three algorithms for all the studies. Also, it is clear that lower values of the thresholds provide less thermal stress at the cost of more performance reduction (mainly due to frequency reduction and the migration overheads). Hence, based on our simulations, we found the values considered in this work as the better values for having a trade-off between the stress reduction and the performance.

We considered 15 degrees as the minimum amplitude for counting the thermal cycles [12]. Using this value, the total number of thermal cycles for all the epochs was counted. In addition, the amplitude of thermal cycles for each simulation scenario was attained by accumulating thermal cycle amplitudes. For the performance (time required to finish processing a job by a benchmark for a given input), we have invoked the number of Tier2 epochs used for finishing the job.

In this paper, we consider 4-, 8-, and 16-core x86 multiprocessors. Each processor is based on Nehalem Intel microarchitecture and derived from Gainestown model codename. Each core comes with one L1 (32 KB) and one L2 (256 KB) private caches while one L3 cache whose size depends on the number of cores is shared among the cores. All cores are out-of-order and can carry out up to two threads simultaneously. Each core consists of five separate functional units including instruction fetch (IF), renaming (RE), execution (EX), load/store (LS), and memory management

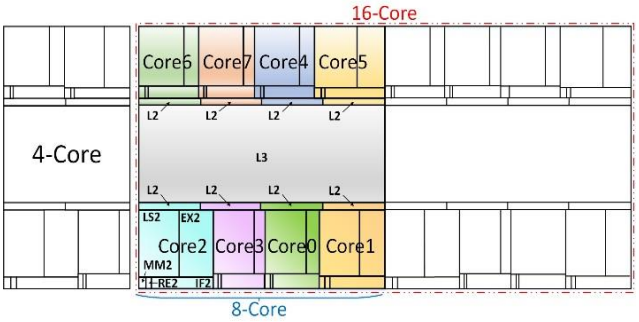


Fig. 9 Floorplan of the 8-core MPSoC

TABLE III. DESIGN PARAMETERS OF THE TARGET MPSOC ARCHITECTURE

N_{core}	P_{const} (Watt)	Dispatch Width	Frequency boundaries (GHz)	L3
4	70	4, 6, 8, 2	$\{f_{b1}, f_{b2}, f_{b3}, f_{b4}\}$	8
8	120	4, 6, 8, 2, 4, 6, 4, 2	$\{f_{b1}, f_{b2}, f_{b3}, f_{b4}, f_{b1}, f_{b2}, f_{b3}, f_{b4}\}$	32
16	200	4, 6, 8, 2, 4, 6, 4, 2, 4, 6, 8, 2, 4, 6, 4, 2	$\{f_{b1}, f_{b2}, f_{b3}, f_{b4}, f_{b1}, f_{b2}, f_{b3}, f_{b4}, f_{b1}, f_{b2}, f_{b3}, f_{b4}, f_{b1}, f_{b2}, f_{b3}, f_{b4}\}$	64

TABLE IV. AVERAGE REDUCTION IN SPATIAL TEMPERATURE GRADIENT, TEMPORAL TEMPERATURE GRADIENT, THERMAL CYCLE NUMBER, AND THERMAL CYCLE AMPLITUDE, AND PERFORMANCE OVERHEAD

	Optimal TheSPoT (%)					Heuristic TheSPoT (%)				
	STG	TTG	TCN	TCA	Perf. Ovh.	STG	TTG	TCN	TCA	Perf. Ovh.
blackscholes	18	10	32	18	3.5	24	11	35	21	4.8
bodytrack	15	11	40	23	4.2	25	11	41	23	5.0
canneal	10	12	26	25	4.0	16	10	19	19	6.1
dedup	21	25	34	29	5.1	35	24	38	34	8.3
facesim	18	14	17	23	5.7	27	16	20	26	6.0
freqmine	5	11	27	19	3.8	22	14	23	31	5.5
vips	5	14	17	19	4.1	14	13	19	12	4.3
x264	16	10	22	26	4.5	26	17	25	23	4.8
ferret	22	18	35	36	5.6	32	21	34	41	7.9

(MM). The floorplan of the MPSoCs studied in this work is shown in Fig. 9. Units are only labeled for the 2nd core.

TABLE III shows the dispatch width, frequency boundaries, power constraints, and L3 cache size (megabytes). We consider 4 different frequency (GHz) boundaries, $f_{b1} = [1.2,2.5]$, $f_{b2} = [1.3,2.66]$, $f_{b3} = [1.2,2.5]$, and $f_{b4} = [1,3]$.

B. Experimental results and discussion

1) Thermal stress reduction

TABLE IV presents the achieved reduction in STG, TTG, TCN (thermal cycle number), and TCA (thermal cycle amplitude) along with the performance overhead (Perf. Ovh.) of the proposed heuristic and optimal approaches of TheSPoT normalized to those obtained from [12], for the 8-core MPSoC.

The achieved reduction in thermal stress is strongly a function of the benchmark nature. For the benchmarks where the workload variations do not cause high temporal or spatial thermal gradients, the proposed approaches do not provide considerable TTG/STG reductions. This is due to the fact that only a few thermal stress violations occur and our thermal

Fig. 10 Number of thermal violations occurred in one run of *blackscholes* benchmark for different number of cores when no thermal stress-aware approach applied

stress constraints and thresholds are not of much. Our approaches specially outperform [12] for benchmarks such as *ferret* and *dedup* featuring different functions with different characteristics at the same time [43]. This improvement occurs because TheSPoT makes decisions based on thermal variations and not only the peak temperature. Conversely, the work proposed in [12] triggers decisions mainly based on peak temperature.

To better understand how our proposed approaches are effective in increasing lifetime reliability in terms of MTTF, we exploited the same methodology and formulation used in [51]. In addition, we modified the TDDB and EM MTTF formulation with respect to [52] in order to include spatial and temporal thermal gradients impact on lifetime reliability. Overall, the MTTF of the proposed optimal and heuristic approaches increased on average, by 35% and 47%, respectively, compared with that obtained by [12]. We considered stress migration (SM), Electromigration (EM), time dependent dielectric breakdown (TDDB), and Thermal cycling (TC) as the most significant failure mechanisms.

Due to lack of access to some technological parameters we were able to report the relative improvement achieved compared to that of [12] as the reference work. However, considering a typical Intel server operating at the ambient temperature of 35oC, the estimated MTTF would be approximately 200000 hours [53]. Thus, assuming no thermal stress-aware power management the MTTF of the system is 200000 hours, whereas the heuristic TheSPoT, optimal TheSPoT and [12] will result in 455700, 418500, and 310000 hours, respectively.

In order to show that TheSPoT provides statistically significant improvement compared to [12] for MTTF, and not by only the mean of the achieved MTTF, we used the Wilcoxon test [54]. Thus, two separate statistical comparisons for "the heuristic TheSPoT and [12]" and "the optimal TheSPoT and [12]" under different benchmarks were considered. Therefore, we formulated the corresponding Null hypotheses as: "the median of the MTTF obtained from the optimal TheSPoT is not higher than that of [12]" and "the median of the MTTF obtained from the heuristic TheSPoT is not higher than that of [12]."

However, since we are evaluating them under different benchmarks, in order to deal with this *multiple comparisons problem*, we conducted the false discovery rate (FDR) test as well. In particular, we used the Benjamini-Hochberg (BH) procedure [55] to control FDR at significance level of $\sigma = 0.05$. The maximum BH-adjusted p-values for the Null

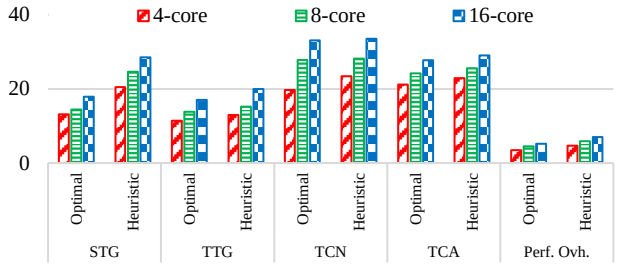


Fig. 11 Average reductions (%) in STG, TTG, TCF, TCA, and performance overhead for TheSPoT compared to [12] for 4-, 8- and 16-core MPSoCs

Hypotheses were obtained 0.033 and 0.038, respectively, for the comparison of [12] with the heuristic TheSPoT and the optimal TheSPoT. Such small BH-adjusted p-values (< 0.05) denote that the alternative hypotheses are valid with sufficient confidence. Hence, both heuristic and optimal TheSPoT approaches are outperforming [12] in MTTF enhancement.

All Null hypotheses were rejected for all benchmarks with $p < 0.05$ showing that for each benchmark TheSPoT provides statistically significant improvement in MTTF over [12].

Fig. 10 shows the average number of thermal stress violations (STG, TTG, and TC), counted regarding our predetermined threshold values as the number of cores on the platforms changes for a basic power and temperature management approach which only considers peak power/temperature values under *blackscholes* benchmark. As noticed, when no thermal stress-aware power and thermal management technique is evoked for the MPSoC, more thermal variation occurs both spatially and temporally when the number of cores increases. When the available resources scales, the scheduler faces more choices to run the jobs at each decision time. However, it is unaware of the decision impact on workload variations and, hence, temperature variations across the chip result in more thermal stress violations.

Fig. 11 provides the average reduction percentages of STG, TTG, TCN, and TCA along with the performance overhead obtained from the proposed methods compared with those of [12] for 4-core, 8-core and 16-core MPSoCs. Our thermal stress-aware approaches outperform [12] with respect to the thermal stress reduction with only a negligible performance overhead as the number of cores increases. In TheSPoT, as the number of cores increases, Tier1 is able to find better source and destination cores for consolidation/deconsolidation, which leads to a higher reduction in thermal stress occurrences. In contrast, the approach in [12] assigns the ready jobs to the coolest core with idle neighbors, which increases the risk of high amplitude thermal cycles.

When we scale the platform, both proposed approaches efficiently reduce thermal stress. Nevertheless, the optimal approach fails to be applicable for many-core processors due to the large runtime overhead, while our heuristic algorithm comes with only 5ms runtime overhead even for larger number of cores. As aforementioned, the larger thermal variation is in time or space, the more efficient our thermal stress-aware approaches are. To demonstrate this hypothesis, we compare the above simulation scenario, running all benchmarks separately and then averaging the results, called normal workload variation, with a new scenario where all benchmarks

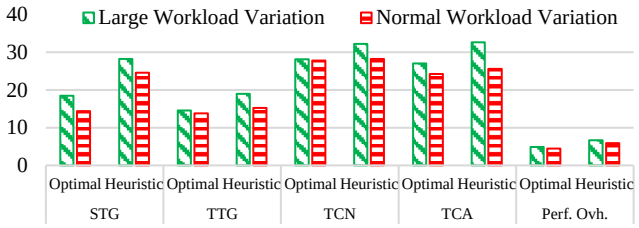


Fig. 12 Average reductions (%) in STG, TTG, TCF, TCA, and performance overhead for TheSPoT compared with [12] for different workload variations

TABLE V. TOTAL NUMBER OF THREAD MIGRATIONS, AND AVERAGE OPERATING FREQUENCIES OF ON CORES

	Number of Thread Migrations			**Average Frequency of ON cores (GHz)**		
	Optimal	Heuristic	[12]	Optimal	Heuristic	[12]
blackscholes	9	11	9	2.11	2.05	2.20
bodytrack	15	15	5	2.21	2.16	2.24
canneal	7	8	4	1.98	1.85	2.10
dedup	64	60	57	1.90	1.74	2.21
facesim	292	312	286	1.77	1.75	2.23
freqmine	23	18	10	1.85	1.75	2.04
vips	14	13	11	1.83	1.80	2.04
x264	77	59	32	1.92	1.88	2.13
ferret	22	20	10	1.88	1.63	2.15

are released and run simultaneously (large workload variation). Fig. 12 reveals much more reduction in thermal stress parameters when the thermal variations (workload variations) are larger. However, this achievement comes with approximately 1% more performance overhead. On the contrary, although the policy of [12] considers temperature variations, it uses peak temperature as the trigger. Thus, it cannot control thermal stress well.

2) Comparison of performance and runtime overhead

On average, for the proposed heuristic(optimal) approach, STG, TTG, TCN, and TCA were, respectively, decreased by 25(14)%, 15(14)%, 28(28)%, and 26(24)% compared with those of [12] with only 6(4.5)% performance degradation. The performance overhead of the proposed approaches in comparison to [12] originates from, first, the reduced average of the operating frequency, and second, more frequent thread migrations as shown in TABLE V. The proposed technique in [12] operates with the maximum available frequency unless a thermal emergency occurs; then, it works with the minimum frequency. Nevertheless, if the number of peak temperature violations increases for a specific benchmark, the overall performance overhead of TheSPoT would decrease compared with that of [12]. Both optimal and heuristic approaches reveal almost the same number of thread migrations, since they employ the same approach for consolidation and deconsolidation. Therefore, the difference in the performance overhead is mainly due to the operating frequency as the optimal approach looks for the optimal frequencies while the heuristic one provides near-optimal values.

Our proposed approaches are implemented in software, and do not require extra hardware. In particular, TheSPoT is able to take advantage of available hardware and knobs dedicated for power and thermal management of modern MPSoCs [16].

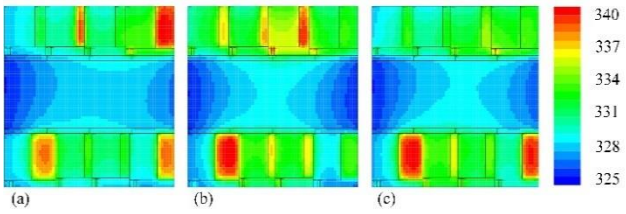


Fig. 13 Thermal map (K) obtained from a) [12], b) optimal, and c) heuristic approaches under *facesim* benchmark for 8-core MPSoC

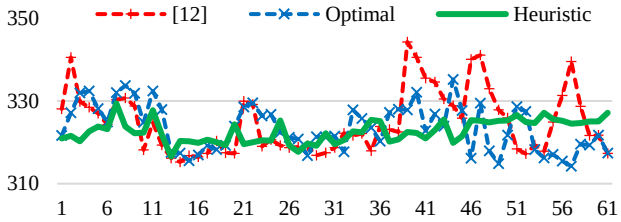


Fig. 14 Average temperature (K) of the first core under *facesim* benchmark

However, any software implementation is accompanied by runtime overhead.

Our proposed approaches are implemented in software, and do not require extra hardware. In particular, TheSPoT is able to take advantage of available hardware and knobs dedicated for power and thermal management of modern MPSoCs [16]. However, any software implementation is accompanied by runtime overhead.

In contrast to the optimal solution, the proposed heuristic algorithm comes with only 5ms computational runtime overhead for 8-core MPSoC. This 5ms overhead is almost constant when using larger number of cores. On the other hand, the computational overhead of [12] is the same as that of our heuristic approach. All in all, the efficacy of TheSPoT is not limited to choosing a 10ms decision epoch (the same interval has been used in several simulation-based works, such as [56]). Although there is tradeoff between the thermal stress reduction and runtime overhead of any thermal aware approach, such as [12], when larger decision epochs are used, TheSPoT still considerably outperform [12] with respect to the achieved MTTF enhancement. However, both approaches encounter slight degradation in the thermal stress reduction. In particular, the MTTF obtained (we performed experiments with *facesim*, and *x264* benchmarks on the 8-core MPSoC) from TheSPoT and [12] decreases by 9% and 6%, respectively when using 100ms decision epoch instead of 10ms epochs.

In this work, as a tradeoff between runtime overhead and thermal stress reduction, we chose 10ms to focus more on the thermal stress reduction. We recall that, by using the same experimental setup for both TheSPoT and [12], we conducted a fair comparison, showing the same runtime overhead but 47% MTTF enhancement for our proposed approach. Reporting the algorithm performance overhead (degradation/improvement) and its runtime overhead separately provides a better insight into comparing different approaches since the runtime overhead, regardless of the decision epoch time, is constant for each scenario.

The heuristic approach ends up with the near-optimal frequency, on average 2% less performance when compared to the proposed optimal solution. However, this performance reduction comes with MTTF improvement. This MTTF enhancement comes from detailed guidelines based on a longer thermal profile history. Specifically, the difference is more obvious for STG reduction, since the proposed heuristic approach considers STG more explicitly when determining the frequencies of cores.

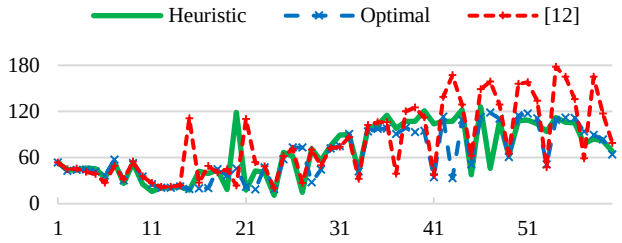


Fig. 15 Total power consumption (Watts) of the 8-core MPSoC under *facesim*

3) Evaluation of the Thermal Stress-Aware Power Management

In this subsection, we show how our thermal stress-aware techniques are able to manage the power/ temperature while maintaining fewer thermal stress violations compared with [12]. For this purpose, we choose *facesim* whose simulation results almost conform to the average values.

Fig. 13 shows the thermal profiles of the 8-core MPSoC obtained by our approaches and [12]. As shown, the spatial temperature gradients obtained by TheSPoT are lower than those of [12], even though in the selected timeslot of *facesim* simulation the maximum temperatures across the chip in all three cases are similar.

The average temperature of the 1st core depicted in Fig. 14 for the first 61 intervals (Tier2 epoch) reveals more temperature variations for the method proposed in [12]. As several threads are launched at the same time, thread migration and core consolidation as well as DVFS add to the thermal variation observed on single core. Hence, large thermal cycles can be noticed not only for the start and end of a simulation.

Also, more peak temperature variation are observed for this core when the thermal management of [12] is applied. In particular, [12] fails to prevent large thermal variation, since it is not the main trigger of its management policy. The total power consumption (Watts) of the MPSoC over time is shown in Fig. 15. The average power consumption attained by [12] is higher than those resulted from TheSPoT. The power consumption exceeds the power constraint (120 watts for 8-core MPSoC) at a few points since [12] does not provide any mechanism to control it.

VIII. Conclusion

In this paper, we have proposed a multi-level thermal stress-aware heuristic power and thermal management approach for MPSoCs. The approach had the objective of increasing the performance while considering the thermal stress constraints including the spatial temperature gradient, temporal temperature gradient, and thermal cycles. The efficacy of the approach was evaluated by simulating MPSoCs with different

number of cores to validate the scalability of the proposed approach. The results of applying the thermal stress-aware approach showed that, compared with a state-of-the-art power and temperature approach [12] and our modified previous work [10], the proposed heuristic approach method reduced the thermal cycle amplitude and frequency as well as temporal/spatial thermal gradients considerably at the price of a minimal performance degradation. While TheSPoT utilized the same algorithm for core consolidation/ deconsolidation, the heuristic DVFS achieved more thermal stress reduction due to considering the spatial and temporal thermal behavior of each core, in detail, in Tier2. In addition, the runtime overhead of the heuristic approach was one sixth of the optimal one in the case of 8-core MPSOC, and more importantly, did not scale with the number of cores. Finally, we showed that our thermal stress-aware approaches behave more efficiently if more workload variations exist in future MPSoC architectures.

ACKNOWLEDGEMENT

This work has been partially supported by the EC H2020 MANGO FETHPC project (Agreement No. 671668), and the ERC Consolidator Grant COMPUSAPIEN (Agreement No. 725657).

REFERENCES

[1] T. Chantem, Y. Xiang, X.Sharon Hu, and R.P. Dick, "Enhancing multicore reliability through wear compensation in online assignment and scheduling," in Design Automation and Test in Europe (*DATE)*, pp.1373-1378, 18-22 March 2013.

[2] A.K. Coskun, D. Atienza, T.S. Rosing, T. Brunschwiler, and B. Michel, "Energy-efficient variable-flow liquid cooling in 3D stacked architectures," in *DATE 2010*, pp. 111-116, March 2010.

[3] T. Ishihara, and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in International Symposium on Low Power Electronics and Design (*ISLPED)*, pp.197-202, 10-12 Aug. 1998.

[4] N.K. Jha, "Low power system scheduling and synthesis," in International Conference On Computer Aided Design (*ICCAD)*, pp.259-263, 4-8 Nov. 2001.

[5] A.K. Coskun, T. Rosing, and K.C. Gross, "Temperature management in multiprocessor SoCs using online learning," in Design Automation Conference (*DAC)*, pp.890-893, 2008

[6] A.K. Coskun, T.S. Rosing, and K.Whisnant, "Temperature Aware Task Scheduling in MPSoCs," in *DATE*, pp.1-6, 16-20 April 2007.

[7] A. Kivilcim Coskun, T. Simunic Rosing, K. Mihic, G. De Micheli, and Y. Leblebici, "Analysis and Optimization of MPSoC Reliability," Journal of Low Power Electronics, vol. 2, no. 1, pp. 56–69, 2006.

[8] A. Kumar, L. Shang, L.-S. Peh, and N. K. Jha, "HybDTM: A coordinated hardware-software approach for dynamic thermal management," in Proc. DAC, 2006, pp. 548–553.

[9] M. Ghasemazar, H. Goudarzi, and M. Pedram, "Robust optimization of a Chip Multiprocessor's performance under power and thermal constraints," in International Conference on Computer Design (*ICCD)*, pp.108-114, Sept. 30 2012-Oct. 3 2012.

[10] M. Kamal, A. Iranfar, A. Afzali-Kusha, and M. Pedram, "A thermal stress-aware algorithm for power and temperature management of MPSoCs," in *DATE*, pp.954-959, 9-13 March 2015.

[11] C. J. Lasance, "Thermally driven reliability issues in microelectronic systems: Status-quo and challenges," Microelectron. Reliab., vol. 43, pp. 1969–1974, 2003.

[12] A.K. Coskun, T.S. Rosing, K.A.Whisnant, and K.C. Gross, "Static and Dynamic Temperature-Aware Scheduling for Multiprocessor SoCs," *IEEE Transaction on Very Large Scale Integration (TVLSI)*, vol.16, no.9, pp.1127-1140, Sept. 2008.

[13] X. Yun, T. Chantem, R.P. Dick, X.S. Hu, and S. Li, "System-level reliability modeling for MPSoCs," in *CODES+ISSS*, pp.297-306, 2010.

[14] S.D. Dowining, and D.F. FSocie, "Simple rainflow counting algorithm," *International Journal of Fatigue*, 4(1), pp. 31-40, 1982.

[15] J. W. McPherson, *Reliability Physics and Engineering*, New York, NY, USA: Springer, 2010.

[16] Processor, Duo. "Power and thermal management in the Intel® core tm." *Intel® Centrino® Duo Mobile Technology* 10.2 (2006): 109.

[17] Iyer, Jayesh, et al. "System Memory Power and Thermal Management in Platforms Build on Intel Centrino Duo Technology." Intel Technology Journal 10.2, 2006.

[18] K.K. Rangan et al., "Thread Motion: Fine-Grained Power Management for Multi-Core Systems," SIGARCH Comput. Archit. News, vol. 37, Issue 3, pp. 302-313, June 2009.

[19] J. A. Winter et al., "Scalable Thread Scheduling and Global Power Management for Heterogeneous Many-Core Architectures," 19th intl. conf. on PACT '10, pp. 29 – 40, September 2010.

[20] H. Jung and M. Pedram, "Supervised Learning Based Power Management for Multicore Processors," IEEE Trans. Comp.-Aided Des. Integ. Cir. Sys., vol. 29, no. 9, pp. 1395-1408, September 2010.

[21] Z. Baoxian, and H. Aydin, "Minimizing expected energy consumption through optimal integration of DVS and DPM," in International Conference on Computer-Aided Design - Digest of Technical Papers,. ICCAD. IEEE/ACM, pp.449-456, 2-5 Nov. 2009.

[22] V. Devadas, and H. Aydin, "On the Interplay of Voltage/Frequency Scaling and Device Power Management for Frame-Based Real-Time Embedded Applications," in *IEEE TC*, vol.61, no.1, pp.31-44, Jan. 2012.

[23] T. Chantem, X. Hu, and R. Dick, "Temperature-Aware Scheduling and Assignment for Hard Real-Time Applications on MPSoCs," IEEE TVLSI, vol. 19, no. 10, pp. 1884–1897, 2011.

[24] A. Mutapcic, S. Boyd, S. Murali, D. Atienza, et al., "Processor Speed Control With Thermal Constraints," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 9, pp. 1994–2008, 2009.

[25] S. Zhang and K. S. Chatha, "Thermal Aware Task Sequencing on Embedded Processors," in Proc. of the Annual DAC, 2010, pp. 585–590.

[26] F. Mulas, D. Atienza, et al., "Thermal Balancing Policy for Multiprocessor Stream Computing Platforms," IEEE Transaction on Computer-Aided Design (*TCAD)*, vol. 28, no. 12, pp. 1870–1882, 2009.

[27] V. Hanumaiah and S. Vrudhula, "Temperature-Aware DVFS for Hard Real-Time Applications on Multicore Processors," IEEE Transactions on Computers (TC), vol. 61, no. 10, pp. 1484–1494, 2012.

[28] X. Zhou, J. Yang, M. Chrobak, and Y. Zhang, "Performanceaware Thermal Management via Task Scheduling," ACM TACO, vol. 7, no. 1, pp. 5:1–5:31, 2010.

[29] M. A. Faruque, J. Jahn, and J. Henkel, "Runtime Thermal Management Using Software Agents for Multi- and Many-Core Architectures," IEEE Design & Test of Computers, vol. 27, no. 6, pp. 58–68, 2010.

[30] G. Liu, M. Fan, and G. Quan, "Neighbor-aware Dynamic Thermal Management for Multi-core Platform," in Proc. of *DATE*. EDA Consortium, 2012, pp. 187–192.

[31] G. Singla, G. Kaur, A. Unver, and U. Ogras, "Predictive Dynamic Thermal and Power Management for Heterogenous Mobile Platforms," In Proc. of *DATE*, pages 1-6, IEEE, 2015.

[32] C. Jeonghwan, et al., "Thermal-aware task scheduling at the system software level," In Proc. *ISLPED*, pp. 213-218. ACM, 2007.

[33] Y. Jun, et al., "Dynamic thermal management through task scheduling," In International Symposium on Performance Analysis of Systems and software, pp. 191-201. 2008.

[34] P. Mercati, et al., "Workload and user experience-aware Dynamic Reliability Management in multicore processors," in *DAC*, pp.1-6, May 29 -June 7 2013.

[35] A. Das, et al., "Reinforcement learning-based inter- and intra-application thermal optimization for lifetime improvement of multicore systems," in *DAC*, pp.1-6, 1-5 June 2014.

[36] I. Ukhov, B. Min, P. Eles, and P. Zebo, "Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems," in *DAC*, pp.197-204, 3-7 June 2012.

[37] M. Gomaa, , D.P. Michael, and T.N. Vijaykumar, "Heat-and-run: leveraging SMT and CMP to manage power density through the operating system." In *ACM SIGARCH Computer Architecture News*, vol. 32, no. 5, pp. 260-270. ACM, 2004.

[38] A. Iranfar, S. Shahsavani, M. Kamal, and A. Afzali-Kusha, "A heuristic machine learning-based algorithm for power and thermal management of heterogeneous MPSoCs," in *ISLPED*, pp. 291-296, 2015.

[39] S. Naffziger, "Amd's commitment to accelerating energy efficiency," 2015.

[40] R. Efraim, et al. "Power-management architecture of the intel microarchitecture code-named sandy bridge." *IEEE micro* 32.2, 2012, 20-27.

[41] Y. Han, I. Koren, C. M. Krishna, "TILTS: A fast architectural-level transient thermal simulation method," Journal of Low Power Electronics, 3(1), 2007.

[42] H. Wei, et al., "Accurate, Pre-RTL Temperature-Aware Design Using a Parameterized, Geometric Thermal Model," in *IEEE TC*, vol.57, no.9, pp.1277-1288, 2008.

[43] V. Craeynest, et al., "Scheduling heterogeneous multi-cores through performance impact estimation (PIE)," In *ACM SIGARCH Computer Architecture News*, vol. 40, no. 3, pp. 213-224. IEEE Computer Society, 2012.

[44] T.E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulation," in Int. Conf. for *High Performance Computing, Networking, Storage and Analysis (SC)*, pp.1-12, 12-18 Nov. 2011.

[45] K. Van Craeynest, et al., "Fairness-Aware Scheduling on Single-ISA Heterogeneous Multi-Cores," Int. Conf. on *PACT*. 2013. pp. 177-187

[46] G. Faust, et al. , "ArchFP: Rapid prototyping of pre-rtl floorplans." In *VLSI-SoC*, pp. 183-188. IEEE, 2012.

[47] B. Christian, et al., "The PARSEC benchmark suite: Characterization and architectural implications," In Proceedings of the 17th international conference on Parallel architectures and compilation techniques, pp. 72-81. ACM, 2008.

[48] L. S. Ahn, et al., "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *Int. Symp. on Microarchitecture*, pp.469-480, 12-16, 2009.

[49] S.G. Johnson, The NLopt nonlinear-optimization package, http://ab-initio.mit.edu/nlopt

[50] K. Skadron, et al. "Temperature-aware microarchitecture: Modeling and implementation." *ACM TACO* 1.1 (2004): 94-125.

[51] Srinivasan, Jayanth, Sarita V. Adve, Pradip Bose, and Jude A. Rivers. "The case for lifetime reliability-aware microprocessors." In *ACM SIGARCH Computer Architecture News*, vol. 32, no. 2, p. 276. IEEE Computer Society, 2004.

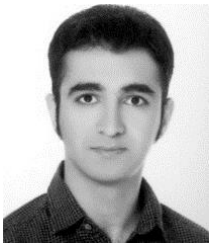
[52] Zh. Lu, et al., "Analysis of temporal and spatial temperature gradients for IC reliability." *University of Virginia Technical Report*, 2004.

[53] https://www.intel.com/content/www/us/en/support/server-products/000007224.html

[54] E. A. Gehan,. "A generalized Wilcoxon test for comparing arbitrarily singly-censored samples." *Biometrika* 52.1-2 (1965): 203-223.

[55] Y. Benjamini and, Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing." *Journal of the royal statistical society. Series B (Methodological)*, 1995, pp.289-300.

[56] S. Sharifi, A.K. Coskun, and T.S. Rosing. "Hybrid dynamic energy and thermal management in heterogeneous embedded multiprocessor SoCs." *Proc. of ASPDAC*. IEEE Press, 2010.



Arman Iranfar received the B.S. degree in Electrical engineering from Isfahan University of Technology, Iran, in 2013 and the M.S. degree in Electrical Engineering, circuits and systems from the University of Tehran, Iran. He is currently pursuing the Ph.D. degree in Electrical Engineering in Swiss Federal Institute of Technology Lausanne (EPFL). His research interests include reliability and power and temperature management of MPSoCs.

Mehdi Kamal received the B.Sc. degree from the Iran University of Science and Technology, Tehran, Iran, in 2005, the M.Sc. degree from the Sharif University of Technology, Tehran, in 2007, and the Ph.D. degree from the University of Tehran, Tehran, in 2013, all in computer engineering. He is currently the assistant professor with the School of Electrical and Computer Engineering of the University of Tehran, Iran. His current research interests include reliability in nanoscale design, approximate computing, neuromorphic computing, design for manufacturability, embedded systems design, and low-power design.

Ali Afzali-Kusha Ali Afzali-Kusha received the B.Sc. degree from the Sharif University of Technology, Tehran, Iran, in 1988, the M.Sc. degree from the University of Pittsburgh, Pittsburgh, PA, USA, in 1991, and the Ph.D. degree from the University of Michigan, Ann Arbor, MI, USA, in 1994, all in electrical engineering. He was a Post-Doctoral Fellow with the University of Michigan from 1994 to 1995. He was a Research Fellow with the University of Toronto, Toronto, ON, Canada, and the University of Waterloo, Waterloo, ON, in 1998 and 1999, respectively. He has been with the University of Tehran, Tehran, since 1995, where he is currently a Professor with the School of Electrical and Computer Engineering and the Director of the Low-Power High-Performance Nanosystems Laboratory. His current research interests include low-power high-performance design methodologies from the physical design level to the system level for nanoelectronics era.

Massoud Pedram received the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley, Berkeley, CA, USA, in 1991. He is currently the Stephen and Etta Varra Professor with the Ming Hsieh Department of Electrical Engineering, University of Southern California, Los Angeles, CA, USA. He holds 10 U.S. patents and has authored four books, 12 book chapters, and over 190 archival and 430 conference papers. His current research interests include low-power electronics, energy-efficient processing, and cloud computing to photovoltaic cell power generation, energy storage, and power conversion, and RT level optimization of VLSI circuits to synthesis and physical design of quantum circuits. Prof. Pedram and his students have received six conference and two IEEE Transactions Best Paper Awards for the research. He was a recipient of the 1996 Presidential Early Career Award for Scientists and Engineers and an ACM Distinguished Scientist, and currently serves as the Editor-in- Chief of the ACM Transactions on Design Automation of Electronic Systems. He has served on the Technical Program Committee of a number of premiere conferences in his field. He was the Founding Technical Program Co Chair of the 1996 International Symposium on Low-Power Electronics and Design and the Technical Program Chair of the 2002 International Symposium on Physical Design.

David Atienza (M'05-SM'13-F'16) is associate professor of electrical and computer engineering, and director of the Embedded Systems Laboratory (ESL) at the Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland. He received his PhD in computer science and engineering from UCM, Spain, and IMEC, Belgium, in 2005. His research interests include system-level design and thermal-aware optimization methodologies for 2D/3D high-performance multi-processor system-on-chip (MPSoC) and ultra-low power system architectures for wireless body sensor nodes. He is a co-author of more than 250 papers in peer-reviewed international journals and conferences, several book chapters, and seven patents. Dr. Atienza received an ERC Consolidator Grant in 2016, the IEEE CEDA Early Career Award in 2013, the ACM SIGDA Outstanding New Faculty Award in 2012, and a Faculty Award from Sun Labs at Oracle in 2011. He served as DATE 2015 Program Chair and DATE 2017 General Chair. He is a Senior Member of ACM and an IEEE Fellow.