# A Thermal Stress-Aware Algorithm for Power and Temperature Management of MPSoCs

Mehdi Kamal[1], Arman Iranfar[1], Ali Afzali-Kusha[1], Massoud Pedram[2]
[1]School of Electrical and Computer Engineering, University of Tehran, Iran
[2]Department of Electrical Engineering-Systems, Univerity of Southern California, USA
{mehdikamal, a.iranfar, afzali}@ut.ac.ir, pedram@usc.edu

***Abstract*—In this work, we propose a thermal stress-aware algorithm for the management of the power and temperature in MPSoCs. The algorithm, which uses a heuristic approach, controls the power consumption, maximum temperature, thermal cycles, and temporal/spatial thermal gradients of MPSoCs. At the top level, the decision on turning the cores on and off is made based on the constraints of peak temperature, maximum spatial thermal gradient, and power consumption. At the next tier, the optimal frequencies (and supply voltages) of the ON cores, formulated in a convex optimization problem, are determined again based on satisfying the constraints of the maximum total power consumption, peak temperature, thermal cycles, and also temporal thermal gradient. The technique may be applied to both the heterogeneous and homogenous MPSoCs. The efficacy of the proposed approach in reducing the thermal cycles as well as temporal thermal gradient is evaluated by comparing its results with a similar previous power and temperature management approach. The evaluation which is performed on 8-core processors under Splash2 benchmarks, demonstrates the ability of the suggested technique in limiting a considerable reduction in the thermal stress parameters.**

## I. Introduction

Shrinking the technology size has enabled the manufacturing of processors with more than one computational core for higher computation speed. These multiprocessor system on chips (MPSoCs) play a major role in modern computational systems [1]. The same as other integrated circuits containing large number of devices, the increase in the speed/performance of these systems, achieved through scaling down device feature sizes, could be accompanied by power and thermal problems. Regular homogeneous MPSoCs are simple for programmers, while they are more expensive and less energy efficient than heterogeneous architectures [2]. As the performance of these systems increases, the power density of the chip also increases. In addition, the non-uniformity of the temperature profile across the chip enlarges the spatial thermal gradient at some points strongly deteriorating the system reliability. As a result, both power and thermal management techniques should be exploited to have a lower power yet reliable system on chip.

Different power management techniques including Dynamic Voltage Scaling (DVS) [3] and task allocation and scheduling [4] have been proposed. These techniques, which lower the average power consumption, help reducing the chip average temperature which is proportional to the power. This alone improves the reliability of the system by reducing hard failures. The power management policies cause decreasing the average temperature by, *e.g.*, turning off the idle cores. This improves the reliability by mitigating some failure mechanisms such as Electromigration (EM) and Time-Dependent-Dielectric-Breakdown (TDDB) ) due to hot spots reduction [5]. These techniques, however, do not explicitly concentrate on suppressing the thermal stresses such as the temporal and spatial temperature gradients as well as thermal cycle which adversely affect the whole system reliability [6]. A thermal cycle occurs when temperature rises from an initial value and then falls to the starting value [7]. The thermal cycle may reduce the whole system mean time to failure (MTTF) as the cycle amplitude increases. Large amplitudes are normally induced due to imperfect task scheduling on a single core as well as some power management techniques which turn the cores off and on frequently [5]. This suggests that while reducing the power consumption using the power management techniques helps improving the reliability by reducing the average temperature, at the same time, it may deteriorate the lifetime by aggravating the thermal cycle effects (or more generally thermal stresses).

Most of the conventional power and thermal management policies, such as Dynamic Voltage and Frequency Scaling (DVFS) [3], Dynamic Power Management (DPM) [8], integrated DVS and DPM [9] and [10], adaptive task scheduling, and thread migration (*e.g.*, Heat-and-Run thread migration [11]), do not consider the deteriorating impacts of thermal cycle and temporal temperature gradient on the reliability. There are some temperature management techniques considering the thermal stresses (*e.g.*, [1], [5], and [12]) which perform online assignment and scheduling for improving the reliability of MPSoCs. These techniques, however, do not consider the maximum power as a design constraint. On the other hand, the thermal management method, proposed in [1], does not consider the minimum performance improvement as a design constraint. The temporal temperature gradient (the rate of temperature changes per unit time) which is another parameter affecting the reliability, has not been considered in [1] and [12]-[15].

In this work, we propose a thermal stress-aware extension of the variable power/thermal management (VPTM) framework introduced in [15]. The framework is called TSA-VPTM. The VPTM is a three-level framework that performs power and thermal managements by core consolidation and DVFS. The extension of the VPTM is performed by using a proposed heuristic algorithm for the consolidation and deconsolidation. The consolidation and deconsolidation is accomplished for reducing the power consumption and lowering thermal gradients by considering the temperature profile of the MPSoC. Also, the DVFS is performed by considering an adaptive temporal thermal gradient constraint which limits the amounts of the thermal gradients and the chip temperature. The constraint adjusts itself at each decision point

978-3-9815370-4-8/DATE15/©2015 EDAA

by considering the current temperature and the thermal profile of the cores in the last decision epoch. The rest of the paper is organized as follows. In Section II, the related works are briefly reviewed. Section III overviews the background concepts for the proposed technique. We present TSA-VPTM framework in detail in Section IV. The results are discussed in V while the paper is concluded in VI.

## II. Related Works

The existing methods dealing with the thermal optimization and power management may be classified into two categories of static and dynamic techniques. In the static techniques, offline application mapping and scheduling are performed to minimize the temperature and/or reduce the thermal cycles based on the thermal characterization at the design time. The workload characteristics, however, change dynamically in real applications making dynamic strategies more preferable than static ones. In [13], the authors propose a steady state temperature-aware task mapping and scheduling on a heterogeneous multicore architecture by considering the thermal cycle effect. The proposed approach optimizes the reliability with just a simple energy cost without considering the performance cost. In [5], an online learning method for the temperature management is proposed. The method uses a multivariate loss function to consider hot spots, thermal cycles, spatial gradients, and average load altogether. While the proposed approach reduces the hot spots and thermal cycle, it is not effective for managing the thermal gradients (only %5 improvements). Also, while the proposed approach considers different power management policies (DPM, DVS, thread migration, load balancing, and Adaptive-Random [6][8]), it does not consider the power consumption as a constraint. The approach proposed in [8] uses both static and dynamic methods to reduce the frequency of hot spots, and lowers the amounts of the spatial gradient and thermal cycles. For the static strategy, an integer linear programming scheduling method which optimizes the power and temperature under the performance constraint is introduced. However, when it performs thermal hot spots balancing and temperature variation suppressing, no spatial gradient reduction is provided. In the dynamic method, a heuristic technique is exploited that allocates ready jobs to the coolest processor which has idle neighbors. Also, in [8], an Adaptive-Random technique which considers the temperature histories of the cores as well as their current temperatures is proposed. In addition, the proposed consolidation policy does not consider the thermal cycle and thermal gradients. In [1], an online task assignment and scheduling technique for maximizing the lifetime reliability of MPSoCs based on heterogeneous architectures is proposed. The approach calculates the impacts of all failure mechanisms on the current wear state of the system and if the wear-out is mostly due to thermal cycle, the appropriate decisions will be taken. The power and performance constraints, however, are not considered in the proposed technique. In [12], a hierarchical controller based on aging sensor for improving the performance and user experience of homogeneous MPSoC has been proposed. It uses a long term controller to monitor the system reliability and a short term controller to adjust the voltage and frequency such that the minimum temperature can be obtained. While the impact of the thermal cycle is considered, the power consumption constraint is not included. A learning algorithm for improving the lifetime reliability of homogenous MPSoCs based on controlling the average temperature and thermal cycle is discussed in [14]. The proposed algorithm increases the power saving by reducing the leakage power. All of the above works have considered the thermal cycle as an important parameter to minimize. Some of these works, however, do not consider the power consumption constraint and/or improving the performance of the MPSoC. Additionally, some of the proposed methods are only applicable to homogeneous MPSoC.

## III. Background Concepts

In this section, first, the thermal cycle phenomenon is discussed. Then, the details of the VPTM framework which is proposed in [15] and is extended in this work are described.

### A. Thermal Cycle

The changes of the power consumption of the chip components result in temperature variations across the chip. These temperature fluctuations, regarded as thermal cycles, cause the expansion and contraction of the chip integrated components. Different expansion coefficients for different materials used on the chip induce different expansions and contractions for different material layers (and components). As a result, thermomechanical stresses occur due to this expansion coefficient mismatch, leading to some failure mechanisms such as dielectric/thin film cracking, fractured bond wire, solder fatigue, cracked die [16]. Thus, the thermal cycle, which corresponds to the temperature rise (expansion) and fall (contraction), adversely affects the reliability of the whole system. Therefore, in addition to the importance of the average and maximum temperature, the amplitude and frequency of the temperature oscillations must be considered [13]. Considering this problem, the management techniques which intend to suppress the thermal stresses should avoid frequent switching of cores as much as possible, which is usually done through consolidation/ deconsolidation in MPSoCs to reduce the power consumption.

The number of cycles which causes the first failure occurrence is obtained from the modified Coffin-Manson equation as [7]

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

where $\delta T$ is the maximum thermal amplitude change of the thermal cycles, $T_{th}$ is the threshold temperature at which inelastic deformation begins, $b$ is the Coffin-Manson exponent constant, $E_{aTC}$ is the activation energy, $T_{max}$ is the maximum temperature in the cycle, and $A_{TC}$ is an empirically determined constant [7]. As it is obvious from (1), failure occurs earlier when thermal cycle amplitudes are greater. Therefore, the prime goal of the thermal cycle management should be decreasing the thermal cycle amplitude in addition to the cycle number.

### B. VPTM

The VPTM framework, which is proposed in [15], is a hierarchical dynamic power/thermal management targeting heterogonous MPSoCs in the presence of the process variation. The proposed architecture for VPTM consists of Tier1 module, Tier2 module, and a PI (proportional-integral) feedback controller. Also, the proposed architecture contains a workload

analyzer which provides the IPC of the running application by applying a moving average calculation. Tier1 and Tier2 are called at the beginning of their corresponding decision epochs. Tier1 performs the core consolidation and avoids thermal emergencies using a greedy algorithm. The algorithm looks for a local optimum solution by comparing the current case with "one more ON core" and "one less ON core" cases in terms of the MPSoC throughput, power dissipation, and maximum die temperature at the beginning of each of its decision epochs. Tier2 applies a coarse-grain DVFS policy to maximize the throughput (*i.e.*, instructions per second, IPS, values) solving a convex optimization problem under the temperature and power budget constraints. The Tier1 epoch is larger than the Tier2 epoch, and hence, the DVFS is performed more often than the core consolidation. In Tier3, the PI controller measures the actual power of each core and changes its DVFS setting so that the set point determined by Tier2 could be obtained. More details about the VPTM framework may be found in [15].

## IV. Proposed Thermal Stress-Aware Technique

In this work, we use the hierarchical structure of VTPM as the base framework and modify its algorithms in Tier1 and Tier2. The greedy approach used for Tier1 in [15] is replaced by a proposed heuristic approach which considers spatial thermal gradient reduction as an objective. Additionally, the convex optimization problem in Tier2 is also modified to consider temporal thermal gradient and thermal cycles constraints. The details of the core consolidation and deconsolidation (in Tier1), and also the proposed DVFS formulation (in Tier2) are described in more details in the following subsections.

### *1) Consolidation*

In the proposed consolidation approach, the consolidation candidates (tuples of cores) are selected based on the load on the cores. Among the candidates, one tuple is selected by minimizing the temperature variation impact of performing the consolidation (turning off the source core) and the cost of thread migration from the source core (the core will be turned off) to the destination core. Note that by performing the consolidation, a core will be turned off while the load on the other core may be increased which leads to large spatial thermal gradients. Hence, selecting proper source and destination cores for the consolidation is vital.

In the proposed approach, first, we select a tuple of ($i$, $j$) cores which the $i^{th}$ core is the candidate to be turned off and the $j^{th}$ core is the destination core of the threads of the $i^{th}$ core (in the case where the $i^{th}$ core contains some threads). The $i^{th}$ core will be selected if its IPS is smaller than a predefined constant value ($IPC_{CONST,i}$), and the cost of its thread migration to the $j^{th}$ core is smaller than a predefined constant value ($Cost_{Const,Migration}$). The $j^{th}$ core will be selected if the consolidation of its thread and the threads of the $i^{th}$ core do not lead to an IPS which is more than the maximum IPS of the $j^{th}$ core. Note that in the case where the $i^{th}$ core does not have any thread, one does not need to worry about the destination core.

Now, for each tuple, the difference between the maximum and minimum temperatures of the chip is estimated by assuming that the consolidation is performed and the $i^{th}$ core is turned off. To estimate the temperature pattern after the consolidation, we consider a power of zero for the $i^{th}$ core, and increase the power of the $j^{th}$ core based on the IPS of the threads migrated to it. By assuming that the IPS of the $j^{th}$ core will be equal to the summation of the IPS of the $j^{th}$ core before the consolidation and the IPS of the $i^{th}$ core, the power consumption of the core is estimated. To estimate the power of the destination core, we use the power model which is proposed in [15]. In this model, the power is a function of the core frequency and the temperature as [15]

$$P(f,\theta) = d.f^{\beta} + l.f + k_{\theta}.\theta \quad (2)$$

where $d$, $l$ and $k_\theta$ are empirical coefficients, and β has a value between 2 and 3. Hence, the power consumption of the $j^{th}$ core is estimated by (2) assuming that its frequency is increased such that the core can handle the IPS value required after the consolidation. Therefore, the frequency is obtained from

$$f_{AC,j} = \frac{IPS_i + IPS_j}{IPS_j} \times f_{c,j} \quad (3)$$

where $f_{AC,j}$ is the frequency of the $j^{th}$ core after the consolidation and $f_c$ is its frequency before the consolidation. Note that if the value of $f_{AC,j}$ becomes more than the maximum frequency of the $j^{th}$ core (*i.e.*, $f_{max,j}$), $f_{AC,j}$ is clamped to the $f_{max,j}$. Now, based on the relation between the temperature and power, $\theta(t+\Delta t)$ values for the units of the MPSoC are obtained from

$$\boldsymbol{\theta}(t+\Delta t) = \mathbf{A}.\boldsymbol{\theta}(t) + \mathbf{B}.P(t) \quad (4)$$

where $\mathbf{A}$ and $\mathbf{B}$ are $n \times n$ ($n$ is equal to the units of the MPSoC) coefficient matrices. These matrices are extracted based on the floorplan and also the materials used to fabricate the chip. Note that, in this work, these coefficients are extracted by using the Hotspot tool [17]. $\boldsymbol{\theta}(t+\Delta t)$ is an $n \times 1$ matrix where its $i^{th}$ row contains the temperature of the $i^{th}$ unit. Also, for each core, we consider five units of instruction fetch, renaming, execution, load store, and memory management. Additionally, we consider L2 cache and L3 cache units in the MPSoC.

After estimating the temperatures of different units, the temperature difference between the coolest and hottest units of the cores is considered as the thermal cost of the tuple. Note that due to the small temperature variation of the caches, we suggest estimating only the temperatures of the functional stages of the cores. After the chip temperature evaluation, the tuple with the smaller amount of migration ($Cost_{Migration}$) and thermal cost ($Cost_{Temperature}$) should be selected. Hence, one may use the merit function ($M$) given by

$$M_k = Cost_{Migration,k} + Cost_{Temperature,k} \quad (5)$$

for the tuple selection process. The tuple with the lowest merit value is selected.

### *2) Deconsolidation*

The core deconsolidation may be performed under two cases. In the first case, the temperature of a core reaches to a value more than the critical temperature ($\theta_{Critical}$) 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. Here, despite the approach addressed in [15] which turns the core off, we propose to decrease the load on the core to reduce the temporal thermal gradients of that core due to its deactivation. In the case where the core has only one thread, when its temperature reaches to a value higher than the critical temperature, we also deactivate the core (*i.e.*, turn it off) and move its thread to another core. In the second case, the frequency of the core is at its maximum value and the

core contains more than one thread. Here, one thread from the core is selected to be migrated to another core.

In both cases, the destination core for the selected thread is chosen based on the IPS of the selected thread and the IPS of the destination cores. First, among the active cores, those whose IPS value will remain smaller than the maximum one after the migration are selected as the eligible destinations for the selected thread. For each destination core, the temperature map of the MPSoC die is estimated based on the description in the previous subsection. Then, the core which leads to the smallest temperature difference is selected as the destination core. If there is not any eligible core, the inactive cores (if there exists any) are considered as the eligible destination cores. Based on the aforementioned description, an inactive core is selected as the destination core provided that its turning on and migrating the thread to it lead to the lowest temperature difference when compared to those of other inactive nodes. Note that if there are not any eligible active and inactive cores, in the first case, the core with the highest difference between its current IPS and its maximum achievable IPS is selected as the destination core. However, in the second case, no migration will occur in the deconsolidation process.

*3) DVFS*

In Tier2, to select the optimal frequency of each core, we have used the formulation proposed in [15] as the base of this tier. In addition to the maximum temperature and maximum power constraints, we propose to add the temporal thermal gradient constraint which should be defined for each unit of the MPSoC. For each unit, the increase and decrease rates of the temperature is limited to $\nabla\theta_{INC,Const}$ and $\nabla\theta_{DEC,Const}$, respectively. Therefore, the thermal gradient constraints for the $i^{th}$ unit may be defined by

$$\frac{\theta(t+\Delta t)_i - \theta(t)_i}{\Delta t} < \nabla\theta_{INC,Const_i} \tag{6-1}$$

$$\frac{\theta(t)_i - \theta(t+\Delta t)_i}{\Delta t} < \nabla\theta_{DEC,Const_i} \tag{6-2}$$

where $\theta_i(t)$ is the current temperature of the $i^{th}$ unit, $\Delta t$ is the Tier2 epoch duration, and $\theta_i(t+\Delta t)$ is the temperature of that unit after $\Delta t$. Note that based on (2) and (4), $\theta(t+\Delta t)$ is a function of the frequency. This constraint set the bounds on the thermal variation of the $i^{th}$ unit in each Tier2 epoch.

In the proposed formulation for the thermal gradient constraints, to control the amplitude of the thermal cycle along with the thermal gradients in the DVFS process, we propose to adjust the values of $\nabla\theta_{INC,Const}$ and $\nabla\theta_{DEC,Const}$ dynamically. The adjustment is performed based on the current temperature, the predefined constraint for the maximum difference between the maximum and minimum temperature ($\Delta\theta_{MAX,Const}$), the maximum predefined thermal gradients ($\nabla\theta_{MAX,Const}$) and also, the peak ($\theta_P$) and valley ($\theta_V$) of the temperature in the last thermal cycle. Note that the peak and valley temperature for each unit depends on its temperature profile in the previous epoch. At the beginning of each Tier2 epoch, before solving the convex optimization problem, the thermal gradient constraints are determined. In the case where the temperature of the $i^{th}$ unit in the last Tier2 epoch duration was increased (positive slope), the $\nabla\theta_{INC,Const_i}$ and $\nabla\theta_{DEC,Const_i}$ would be defined based on

$$\begin{aligned}
&if\ (\theta_{C,i} > \theta_{P,i})\\
&\quad \nabla\theta_{INC,Const_i} = \alpha\nabla\theta_{MAX,Const}\\
&else\ if\ (\theta_{P,i} - \theta_{C,i} > \Delta\theta_{MAX,Const})\\
&\quad \nabla\theta_{INC,Const_i} = \nabla\theta_{MAX,Const}\\
&else\\
&\quad \nabla\theta_{INC,Const_i} = \alpha\nabla\theta_{MAX,Const} + \left((1-\alpha)\nabla\theta_{MAX,Const}\right)\frac{e^{\frac{\theta_{P,i}-\theta_{C,i}}{\Delta\theta_{MAX,Const}}}-1}{e-1}
\end{aligned} \tag{7-1}$$

$$\begin{aligned}
&if\ (\theta_{C,i} < \theta_{V,i})\\
&\quad \nabla\theta_{DEC,Const_i} = \alpha\nabla\theta_{MAX,Const}\\
&else\ if\ (\theta_{C,i} > 0.5(\theta_{P,i} - \theta_{V,i}) + \theta_{V,i})\\
&\quad \nabla\theta_{DEC,Const_i} = \nabla\theta_{MAX,Const}\\
&else\\
&\quad \nabla\theta_{DEC,Const_i} = \alpha\nabla\theta_{MAX,Const} + \left((1-\alpha)\nabla\theta_{MAX,Const}\right)\frac{e^{\frac{\theta_{C,i}-\theta_{V,i}}{\Delta\theta_{MAX,Const}}}-1}{e^{0.5}-1}
\end{aligned} \tag{7-2}$$

In the proposed formulation, the temperature increase rate is clamped based on the current temperature and the peak temperature of the previous thermal cycle. The peak of the previous thermal cycle is considered as a reference temperature that the rate of the temperature increase is lowered when the difference between the current temperature and the peak temperature is decreased. Note that if the difference between the current temperature and the peak temperature is more than $\Delta\theta_{MAX,Const}$, the rate of the increasing temperature will be set to its maximum value ($\nabla\theta_{MAX,Const}$). If the current temperature exceeds $\theta_P$, the increase rate is limited to $\alpha\nabla\theta_{MAX,Const}$, which corresponds to the minimum allowed thermal gradients. Here, to guarantee that the thermal variation is within the range of 0 and $\Delta\theta_{MAX,Const}$, α which is a predefined number varying between 0 and 1 is exploited. The zero value for $\alpha$ may results in limiting the performance of the MPSoC. Based on our study, we suggest a small value (< 0.1) for α. Finally, if the current temperature is near the peak temperature (*i.e.*, $\theta_P - \theta_C < \Delta\theta_{MAX,Const}$), the temperature increase rate is reduced exponentially based on the difference between the current temperature and the peak temperature. However, the rate is not reduced to a value smaller than $\alpha\nabla\theta_{MAX,Const}$.

If the slope of the temperature in the last epoch is positive, it is possible that by choosing a frequency in the decision time, the temperature is decreased. Hence, along with the temperature increase rate constraint, the decrease rate constraint ($\nabla\theta_{DEC,Const}$) should be calculated. The decrease rate constraint, is calculated based on the current temperature and the valley of the previous thermal cycle. If the current temperature is smaller than the valley temperature, the minimum value (*i.e.*, $\alpha\nabla\theta_{MAX,Const}$) is considered for $\nabla\theta_{DEC,Const}$. However, if the current temperature is higher than the average of the peak and valley temperatures, the decrease rate constraint is set to its maximum value ($\Delta\theta_{Grad,Const}$). On the other hand, if the current temperature is lower than the average temperature, the rate constraint is decreased exponentially based on the difference between the current temperature and the valley temperature. Note that in the case of increasing the temperature, the increase probability is more than the decrease probability. Hence, in the proposed approach, the temperature increase rate constraint is defined more conservatively compared to that of the decrease rate constraint. This means that the increase rate is bounded smaller compared to the decreasing rate by the same distance to their reference

temperatures (*i.e.*, distance to the $\theta_P$ in the case of the increasing rate, and distance to the $\theta_V$ in the case of the decreasing rate).

The above discussion was about the case when the temperature is increased in the last Tier2 epoch duration. In the case of decreasing the temperature, the idea of determining the increase and decrease rate constraints is almost similar to the case of increasing the temperature. In this case, the formulation of defining the temperature decrease rate constraint ($\nabla\theta_{DEC,Const}$) is almost similar to the increase rate constraint ($\nabla\theta_{INC,Const}$) in the case of increasing temperature. Also, in this case, the temperature increase rate constraint is almost similar to the decrease rate constraint in the case of increasing temperature. Finally, the frequency optimization problem by considering the thermal cycle along with the power and thermal management is defined by

$$\begin{aligned} &Maximize \sum_{i=1}^{|Cores|} IPC_i Xi \\ &Subject\ to: \\ &A.\theta + B.P < \theta_{Critical} \\ &P < P_{budget} \\ &f_{MIN} < f < f_{MAX} \\ &P = D.f^{\beta} + L.f + K_{\theta}.\theta \\ &\frac{1}{\Delta t}\big((A.\theta(t) + B.P) - \theta(t)\big) < \Delta\theta_{INC,max} \\ &\frac{1}{\Delta t}\big(\theta(t) - (A.\theta(t) + B.P)\big) < \Delta\theta_{DEC,max} \end{aligned} \quad (8)$$

where $X_i$ is a binary variable which will be 1 if the $i^{th}$ core is active (*i.e.*, ON) and 0 if the core is OFF.

## V. Experimental Results

To assess the efficacy of the TSA-VPTM, we have studied the impact of the proposed approach on the thermal cycle and thermal gradients under four scenarios compared to the VPTM approach. All the scenarios have been run on an 8-core MPSoC. The details of the cores of the MPSoC, the benchmarks which were run in each scenario, the values of the constraints, and the ambient temperature in each scenario are shown in TABLE I. The benchmarks have been selected from Splash2 benchmark package. The inputs of selected benchmarks in the simulation scenarios were chosen from the input sets which are provided in the benchmark package. The size of the input for each selected benchmark and also the number of the threads for each benchmark are shown in TABLE I.

The simulation framework was implemented by Sniper multicore simulator [18]. The power consumption and the temperature of the MPSoC were estimated by using McPAT [19] and Hotspot [17] tools, respectively. Note that for each core, one L1 private cache (32 KB) and one L2 private cache (256 KB) were assigned while one shared L3 cache (32 MB) was considered. To extract the floorplan of the MPSoC, ArchFP tool [20] was exploited where the area of the different parts of the MPSoC was extracted using McPAT in 45nm technology. The TSA-VPTM and VPTM power and thermal management algorithms were implemented by using Python programming language, and the convex problem of Tier2 was solved by NLOPT tool [21]. In this work, for all scenarios, Tier1 (Tier2) epoch duration was 2ms (1ms). Note that in Tier1 decision times, consolidation and deconsolidation were performed alternately. Also, the migration cost, independent of the source and destination cores, was considered as 10us for each thread migration.

Fig. 1 shows the temporal thermal gradient (TTG) reduction, spatial thermal gradient (STG) reduction, thermal cycle frequency (TCF) reduction, and thermal cycle amplitude (TCA) reduction of the TSA-VPTM compared to the VPTM approach. It should be noted that Dowining simple rainflow-counting algorithm [22] was used to extract the thermal cycles. The results show that the proposed approach reduced the TTG, on average, about 38%. The proposed TSA-VPTM only reduced the STG of the first and third simulation scenarios while in two other scenarios, there were trivial improvements for the STG. It originates from the fact that the proposed approach uses a non-optimal approach to reduce the spatial gradients. Also, the TCF and TCA were reduced, on average, about 61% and 51%, respectively. It should be noted that power and thermal constrains were satisfied by both VPTM and TSA-VPTM in all simulation scenarios. However, for the sake of space, we do not report them here. Also, in the worst-case, the performance of the MPSoCs in the simulation scenarios decreased about 5% in the case of TSA-VPTM compared to the VPTM.

TABLE I MPSoC Architecture, Benchmarks and Constraints in each simulation scenario

| | MPSoC Architecture | | | Benchmarks | Constraints | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Scenario | Frequency ($f_{min}$-$f_{max}$) (GHz) | Voltage ($V_{min}$-$V_{max}$) (V) | Dispatch Width | (Name, Input Size, \|Threads\|) | $\theta_{Ambient}$ (K) | $P_{budget}$ (W) | $\theta_{Critical}$ (K) | $\Delta\theta_{MAX,Const}$ (K) | $\nabla\theta_{MAX,Co}$ (K/ms) |
| 1 | {(1.7-3.0), (1.5-3.2), (1.5,3.0), (1.5-2.5), (2.0-3.0), (1.5-2.6), (1.2-2.6), (1.8-3.2)} | {(0.9-1.2), (0.8-1.0), (0.9-1.2), (0.8-1.0), (0.9-1.2), (0.9-1.0), (0.9-1.1), (0.8-1.0)} | {4, 6, 8, 2, 4, 6, 4, 2} | {(ocean.cont,small,2), (fft,large,2),(radix,small,2), (cholesky,small,2)} | 325 | 80 | 370 | 20 | 3 |
| 2 | {(1.7-3.0), (1.7-3.0), (1.7-3.0), (1.7-3.0), (1.7-3.0), (1.7-3.0), (1.7-3.0), (1.7-3.0)} | {(0.9-1.2), (0.9-1.2), (0.9-1.2), (0.9-1.2), (0.9-1.2), (0.9-1.2), (0.9-1.2), (0.9-1.2)} | {4, 4, 4, 4, 4, 4, 4, 4} | {(radix,large,4), (fft,small,4), (raytrace,large,2), (fft,small,2)} | 350 | 80 | 370 | 10 | 3 |
| 3 | {(1.7-3.0), (1.5-3.2), (1.5,3.0), (1.5-2.5), (2.0-3.0), (1.5-2.6), (1.2-2.6), (1.8-3.2)} | {(0.9-1.2), (0.8-1.0), (0.9-1.2), (0.8-1.0), (0.9-1.2), (0.9-1.0), (0.9-1.1), (0.8-1.0)} | {4, 6, 8, 2, 4, 6, 4, 2} | {(radix,large,4), (fft,small,4),(fft,large,4)} | 345 | 100 | 370 | 15 | 5 |
| 4 | {(1.7-3.0), (2.0-3.2), (1.7-3.0), (2.0-3.2), (1.7-3.0), (2.0-3.2), (1.7-3.0), (2.0-3.2)} | {(0.9-1.2), (0.8-1.2), (0.9-1.2), (0.8-1.2), (0.9-1.2), (0.8-1.2), (0.9-1.2), (0.8-1.2)} | {4, 6, 4, 6, 4, 6, 4, 6} | {(ocean.cont,large,2) , (fft,small,2), (raytrace,large,2), (cholesky,small,2)} | 325 | 60 | 350 | 9 | 3 |

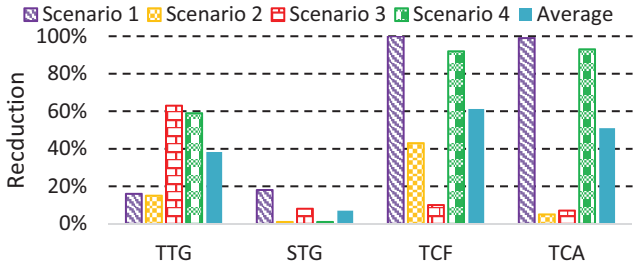


Fig. 1. Temporal thermal gradient (TTG) reduction, spatial thermal gradient (STG) reduction, thermal cycle frequency (TCF) reduction, and thermal cycle amplitude (TCA) reduction of the TSA-VPTM compared to the VPTM

Finally, Fig. 2, as an example, shows the thermal gradients constraints of the execution unit of the 1st core of the MPSoC of the third simulation scenario during the runtime. Note that these constraints are dynamically adjusted by the proposed approach during the runtime of the applications. In this figure, to clearly show the different variations of $\nabla\theta_{INC,Const}$ and $\nabla\theta_{DEC,Const}$, the negative value of $\nabla\theta_{DEC,Const}$ is shown. By increasing the temperature, $\nabla\theta_{INC,Const}$ is decreased while $\nabla\theta_{DEC,Const}$ is increased which this situation is observable at the beginning of the simulation (time < 54ms) in Figure 3. However, by decreasing the temperature, $\nabla\theta_{DEC,Const}$ is decreased and $\nabla\theta_{INC,Const}$ is increased. In Fig. 2, in lower temperatures (time > 72ms), the proposed approach to bound the temperature variation, limited $\nabla\theta_{DEC,Const}$ to small values which led to preventing the temperature reduction. On the other hand, due to the large difference between the current temperature and the peak temperature, when the temperature was decreased, highest value was considered for $\nabla\theta_{INC,Const}$ by the proposed algorithm in Tier2.

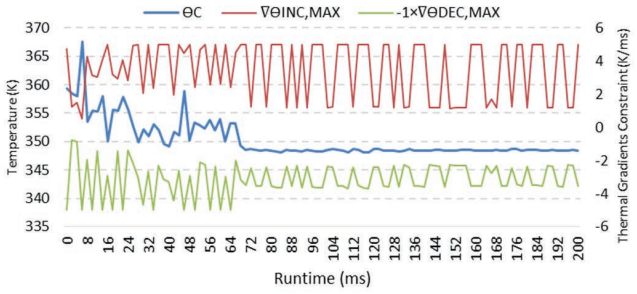


Fig. 2. Current temperature and thermal gradients constraint of execution unit of 1st core in the third simulation scenario.

## VI. Conclusion

In this paper, we proposed multi-level TSA-VPTM framework to manage power and temperature of MPSoC by considering the thermal stresses. In the top level, the proposed technique exploited a heuristic algorithm to perform core consolidation and deconsolidation by considering the thermal stresses. Also, in the second tier, the power and thermal management problem along with thermal stress management problem were formulated in a convex optimization problem. The results showed that thermal cycle frequency, thermal cycle amplitude, temporal thermal gradients, and spatial thermal gradients compared to the VPTM approach were reduced, on average, about 61%, 51%, 18%, and 7%, respectively as well as satisfying power and thermal constraints.

## References

[1] T. Chantem et al., "Enhancing Multicore Reliability Through Wear Compensation in Online Assignment and Scheduling," In DATE, 2013, pp. 1373-1378.

[2] Rakesh Kumar, et. al., "Processor Power Reduction Via Single-ISA Heterogeneous Multi-Core Architectures," Computer Architecture Letters, Volume 2, Apr. 2003.

[3] T. Ishira and H. Yasuura, "Voltage Scheduling Problem for Dynamically Variable Voltage Processors," In ISLPED, 1998, pp. 197-202.

[4] N. K. Jha, "Low Power System Scheduling and Synthesis," In ICCAD, 2001, pp. 259-263.

[5] A. K. Coskun, et al., "Temperature Management in Multiprocessor SoCs Using Online Learning," In DAC, 2008, pp. 890-893.

[6] A. K. Coskun et al., "Temperature aware task scheduling in MPSoCs," In DATE, 2007, pp. 1659-1664.

[7] Y. Xiang, T. Chantem, R. Dick, X. S. Hu, L. Shang, "System-Level Reliability Modeling for MPSoCs," In CODES+ISSS, 2010, pp. 297-306.

[8] A. Coskun et al., "Static and Dynamic Temperature-Aware Scheduling for Multiprocessor SoCs," IEEE Transactions on Very Large Scale Integration Systems (TVLSI), 16(9), pp. 1127–1140, 2008.

[9] B. Zhao and H. Aydin, "Minimizing Expected Energy Consumption through Optimal Integration of DVS and DPM," In ICCAD, 2009, pp. 449-456.

[10] V. Devadas and H. Aydin, "On the Interplay of Voltage/Frequency Scaling and Device Power Management for Frame-based Real-Time Embedded Applications," IEEE Transactions on Computers, vol. 61, no. 1, 2012, pp. 31-44.

[11] M. Gomaa et al., "Heat-and-Run: leveraging SMT and CMP to manage power density through the operating system". In ASPLOS, 2004, pp. 260-270.

[12] P. Mercati et al., "Workload and User Experience-aware Dynamic Reliability Management in Multicore Processors," In DAC, 2013, pp. 2:1–2:6.

[13] I. Ukhov et al., "Steady-state Dynamic Temperature Analysis and Reliability Optimization for Embedded Multiprocessor Systems," In DAC, 2012, pp. 197-204.

[14] A. Das et al., "Reinforcement Learning-Based Inter- and Intra-Application Thermal Optimization for Lifetime Improvement of Multicore Systems," In DAC, 2014, pp. 1-6.

[15] M. Ghasemazar et al., "Robust Optimization of a Chip Multiprocessor's Performance under Power and Thermal," In ICCD, 2012, pp. 108-114.

[16] J. W. McPherson, Reliability Physics and Engineering. Springer, 2013.

[17] W. Huang, et al. "Accurate, Pre-RTL Temperature-Aware Processor Design Using a Parameterized, Geometric Thermal Model Considerations." IEEE Trans. on Computers, 2008, 57(9):1277-88.

[18] T. E. Carlson et al., "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in Proc. SC, 2011, pp. 1–12.

[19] Li, S. et al., "Mc- PAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," In IEEE/ACM International Symposium on Microarchitecture, 2009, pp. 469-480.

[20] Gregory G Faust, Runjie Zhang, Kevin Skadron, Mircea R Stan, and Brett H Meyer. ArchFP: Rapid prototyping of pre-rtl floorplans. In VSLI-SoC, 2012.

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

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