A Control Theory Approach for Thermal Balancing of MPSoC

Francesco Zanini[†], David Atienza[⋆], Giovanni De Micheli[†]
[†] Laboratory of Integrated Systems
[⋆]Embedded Systems Laboratory
Ecole Polytechnique Federale de Lausanne (EPFL), Switzerland
e-mail: {francesco.zanini, david.atienza, giovanni.demicheli}@epfl.ch

Abstract— Thermal balancing and reducing hot-spots are two important challenges facing the MPSoC designers. In this work, we model the thermal behavior of a MPSoC as a control theory problem which enables the design of an optimum frequency controller without depending on the thermal profile of the chip. The optimization performed by the controller is targeted to achieve thermal balancing on the MPSoC thermal profile to avoid hotspots and improve its reliability. The proposed system is able to perform an on-line minimization of chip thermal gradients based on both scheduler requirements and the chip thermal profile. We compare this with state of the art thermal management approaches. Our comparison shows that the proposed system offers a better both thermal profile (temperature differences higher than $4°C$ have been reduced from 27.9% to 0.45%) and performance (up to 32% task waiting time reduction).

I. Introduction

With the advance of technology, the number of functional units and cores integrated on a chip is increasing. Today, several commercial multi-core architectures with few cores to several tens of cores such as IBM's Cell [1], Sun's Niagara [2] and Tilera's 64-core architecture [3] are available. In order to implement these systems, semiconductor industry is facing several technological challenges. It is predicted that in the near future, peak power dissipation and consequent thermal implications will be a major performance bottleneck for multi-core systems [5]. Temperature gradients and hot-spots not only affect the performance of the system, but also lead to unreliable circuit operation and affect the life-time of the chip [4], thus thermal management/balancing for MPSoCs is a critical matter to tackle.

In the last years, thermal management/balancing techniques received a lot of attention as a collateral effect of increasing power density. Adaptive mechanism focusing on handling key micro-architectural hotspots have been proposed in [9] and [14]. In [15] and [13] a significant reduction in localized hotspots has been obtained using thread migration techniques. The problem with these techniques is that they perform the optimization using task migration which requires extra operations to be performed and increases chip power consumption.

Another way, less power consuming to perform thermal balancing is by employing dynamic frequency and voltage scaling (*DVFS*) based techniques. The idea has been proposed in several works [8] - [10]. The major problem of all these approaches is that they are targeting power density reductions with the effect of reducing overall temperature. However this does not directly imply that thermal gradients between different components are minimized or individual hot spots do not appear [6], [13].

A very recent approach tackles processor power optimizations and thermal balancing optimization together using convex optimization [11]. The problem is that in order to make the system feasible from an implementation and convex modelling perspective, several simplifying assumptions needed to be made. These assumptions such as having the whole floorplan all at the same temperature, undermine the overall optimality of the policy.

In this work, there are two main contributions to the state of the art of thermal balancing for MPSoCs. The first one is the model of the thermal behavior of a MPSoC as a control theory problem. This representation enables the design of an optimum frequency controller without the need of having the thermal profile of the chip at design time. The overall system has been modelled using a state space representation [27] having as input parameters scheduler requirements and as output both the amount of workload executed and the MPSoC thermal profile. The thermal profile is also the feedback signal used by the controller as input data to perform thermal balancing. The second contribution is an optimum solution to the frequency assignment problem for thermal balancing of MPSoC based on a linear quadratic regulator.

Our results show that the proposed system guarantees that scenarios with dangerous thermal profiles are avoided while matching the application performance requirements. The proposed thermal balancing approach offers a better thermal profile since the time spent by the MPSoC in scenarios where temperature differences among cores are higher than $4°C$ has been reduced from 27.9% to 0.45% compared with state-of-the-art techniques. In addition to that, in contrast to compared methods, scenarios with temperature differences higher than $7°C$ are completely avoided. Performance is also improved since task waiting time before execution experiences up to 32.8% reduction compared with convex based techniques.

This paper is organized as follows. In Section 2, we overview related work on thermal and power balancing techniques. In Section 3 we present our control model for MPSoC. Section 4 describes our thermal balancing policy. Then, in Section 5 we present our experimental results and we compare our proposed thermal balancing system with state-of-the-art solutions to this problem. Finally, in Section 6, we summarize the main conclusions of the paper.

978-1-4244-2749-9/09/$25.00 ©2009 IEEE

II. Related Work

Many researchers in computer architecture have recently focused on power balancing and thermal control for multi-core systems and MPSoCs [13], [8], [9]. Processor power optimization and balancing using DVFS have been proposed in several works [8], [10]. All these techniques are targeted to reducing power density. This has the effect of reducing overall temperature. However thermal gradients between different components are not definitely minimized or individual hot spots do not appear [6], [13].

Adaptive mechanism focusing on handling key micro-architectural hotspots have been proposed in [9] and [14]. In [15] and [13] a significant reduction in localized hotspots has been obtained using thread migration techniques. Temperature management at system-level, for a set of scheduling mechanism for MPSoC has been presented in [16] and [17].

Several groups have addressed the problem of thermal modelling and simulation at different levels of abstraction. Finite-difference time domain [18], finite element [19], and Green-function [20] based algorithms have been applied in order to model MPSoCs. In [6] a thermal/power model for super-scalar architectures is presented. In [28], the use of feedback control theory is proposed as a way to implement adaptive techniques in the processor architecture. Most of these existing thermal management techniques are based on monitoring and tuning processor frequencies or instruction fetch operations that do not result in optimum solutions. Moreover they fail to consider transient temperature variations and to guarantee that hotspot formation is avoided.

In [11] a new approach to the problem has been proposed. In this case, convex optimization has been used to solve the frequency assignment problem having as target power and hotspots minimization. This kind of optimization is a complex operation that cannot be performed online. The input parameters needed for the optimization are the thermal profile, chip physical parameters and scheduler requirements. However, apart from chip parameters, the other two input data can assume many values. Thus, to make the system feasible for a run-time optimization, only few configurations for both the thermal profile and scheduler requirements can be analyzed in practice and stored in a look-up table for the run-time operation of the system. This assumption undermines the optimality of this method. The problem is that the optimization system assumes a uniform thermal profile as initial condition of the convex optimization process. As this condition does not hold on the run-time real chip thermal profile, there is no guarantee that hotspots are avoided by applying the convex optimization to the MPSoC system.

III. Control theory model for MPSoC

A. High level description

The thermal balancing of a MPSoC can be seen from a control theory perspective as the problem of minimizing thermal gradients on the MPSoC having the chip thermal profile as a feedback signal and scheduler requirements as input reference signal. The block diagram of the proposed control system is shown in Figure 1. The architecture is a single loop feedback discrete time control system [27]. The overall system consists of a plant to control (the MPSoC) and the thermal balancing regulator. The function of the regulator is to control the plant to achieve thermal balancing on the MPSoC thermal profile. Its internal architecture is described in detail in next section. In order to be able to control the MPSoC, the regulator uses an input signal and a feedback signal coming from the plant. The input signal represents the scheduling requirements that the system has to satisfy, while the thermal profile is the feedback signal that the regulator uses to monitor the plant. The output of the regulator is the MPSoC frequency assignment. From the user point of view, input data to the system is the amount of workload that is translated to the scheduler in an input average frequency requirement. As output the user sees only executed tasks and the delay of tasks before execution.

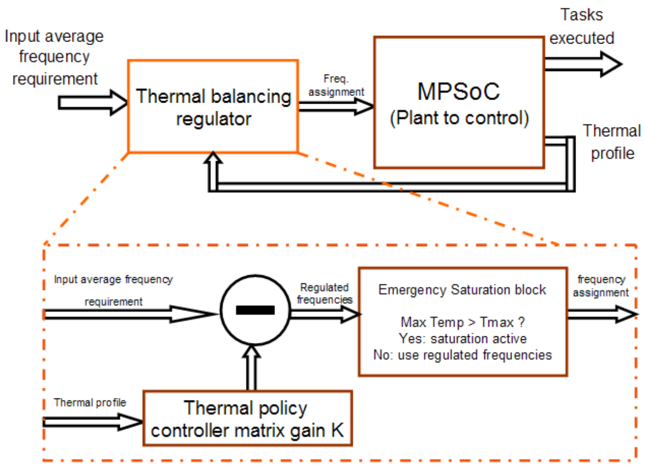
Input average frequency requirement
Thermal balancing regulator
Freq. assignment
MPSoC (Plant to control)
Tasks executed
Thermal profile
Input average frequency requirement
Regulated frequencies
Emergency Saturation block
Max Temp > Tmax ?
Yes: saturation active
No: use regulated frequencies
frequency assignment
Thermal profile
Thermal policy controller matrix gain K

Fig. 1. Control system block diagram.

B. State space heat propagation model

In order to model the physical structure of the MPSoC, two types of layers have been used: the silicon layer and the heat spreading copper layer [21]. The chip floorplan has been divided into several thermal cells of cubic shape. Every single functional unit in the floorplan can be represented by one or more thermal cells of the silicon layer. Thermal modelling is computed by considering the heat conductances G and capacitances C of the cells as calculated and validated in [6] and [21]. The thermal model that we want to represent is nonlinear and in addition to that coefficients are temperature-dependent [21]. To be able to represent the thermal model using a linear, time invariant discrete-time system representation, the solution of the differential equations modelling the heat flow inside the MPSoC system has to be linearized. This mathematical operation has been performed assuming a worst case scenario.

From control theory [27] we know that every linear, time invariant discrete-time system can be represented with the following equations:

$$x(k+1) = Ax(k) + Bu(k) + W \tag{1}$$

$$y(k) = Cx(k) + Du(k) \tag{2}$$

where at time k, $x(k)$ is the plant's state, $u(k)$ is its input and $y(k)$ is its outputs. The temperature value of each cell is the state x of our system. This means that in our case we have

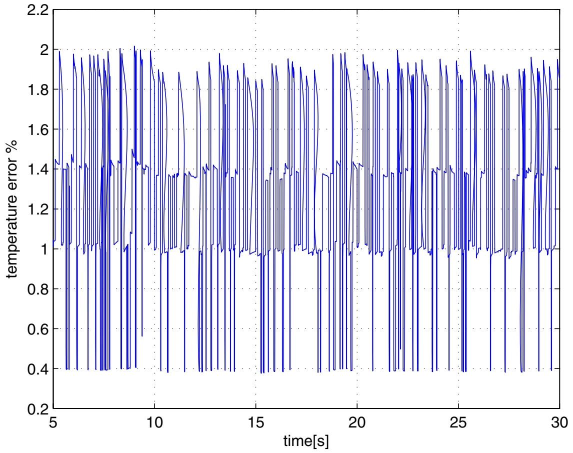

Fig. 2. Maximum percentage error between the temperature of cells of the silicon layer and the ones of the copper layer on it, normalized to the difference between the silicon temperature and the ambient one ($300^\circ K$).

x that is a vector with $2n$ entries where n is the number of blocks composing the floorplan for each of the two layers. The input of the system u is the square of the input frequencies of the cores. This means that assuming a p-core system, u is a vector of size p. The output y of our system is the temperature observed by the on-chip thermal sensors placed in the silicon layer. Matrixes A, B, C, D and W can be computed according to a bijective correspondence between the model describing the heat flow inside the MPSoC by using discrete-time differential equations and previous state-space representation [27].

In order to allow the regulator to control the system, the overall state of the system must be known by the regulator. This means that the temperature of every single cell in which the floorplan has been divided must be known. Temperature of silicon cells are obtained from on-chip thermal sensors, but thermal values of copper cells are not. The basic approach to estimate these temperatures is to use a state estimator [27], but this approach requires expensive circuits in terms of both area and power consumption [29]. Figure 2, shows that using our modelling method only a small approximation error (i.e. less than 2%) is committed by approximating the temperature of a certain copper cell and the one located in the same position of the silicon layer.

IV. Thermal balancing policy

The issue we have to address is the temperature and power difference minimization problem of a linear time-discrete system subjected to constraints. Constraints are performance requirements to be satisfied, thermal balancing and hotspot prevention. In order to solve this problem, we propose the regulator scheme of Figure 1.

If Max Temp (the maximum MPSoC temperature) is less than a certain threshold (Tmax), the overall system presented in Figure 1 is basically a linear feedback system, where MPSoC frequencies are calculated simply by subtracting from the input average frequency requirement the product of the thermal profile and the controller matrix gain K. The emergency saturation block (in Figure 1) just saturates the regulated frequencies to a certain value when the maximum MPSoC temperature is higher than the threshold Tmax. This allows the MPSoC to cool down and so to reduce its maximum temperature in case of overheating.

As linear regulator, we decided to use a linear quadratic regulator (*LQR*). LQR is an optimum regulator obtained by minimizing a cost function J related to both the state and the control, thus the control problem we need to solve can be formalized in the following way:

$$\begin{aligned}
J(u) &= \sum_{k=1}^{inf}(x(k)^T Q x(k) + u(k)^T R u(k)) && (3)\\
min &: J(u) && (4)\\
s.t. &: 0 \le f_i(k) \le F_{max} \;\; \forall\, i,k && (5)\\
& \quad f^2_{1:p}(k) = u(k) \;\; \forall\, k && (6)\\
& \quad x(k+1) = Ax(k) + Bu(k) + W \;\; \forall\, k && (7)\\
& \quad y(k) = Cx(k) + Du(k) \;\; \forall\, k && (8)
\end{aligned}$$

where p is the number of cores processing the tasks, $x(k)$ is the state of the system at time k, $f_i(k)$ is the frequency of core i at time k, and F_{max} is the maximum allowable working frequency. Equations 7 and 8 have been described in section III.

According to Equation 3, matrix Q is related to hotspot minimization while matrix R is related to the power saving constraint. The weights associated with those matrixes represent respectively the importance that hotspot minimization or power saving has in the optimization process. Inequality 5 defines the range of working frequencies that can be used. Equation 6 defines the relation between the input signal u and the working frequencies. Equations 7 and 8 define the evolution of the system according to the present state and inputs. Regarding Equation 4, from control theory we know that that for every matrix Q semi-defined positive and for every matrix R positive defined, it always exist an optimum solution that minimizes the cost function J. This minimization process is independent from the average workload constraint $f_{avg}(k)$ coming from the scheduler at time k. To force the system to be regulated by taking into account this constraint, the bias signal $f_{avg}(k)$ called 'input average frequency requirement' is added to the control loop as shown in Figure 1. The result of this quadratic optimization problem is the gain matrix K which provides the thermal balancing frequency regulation. Current frequencies values are obtained by simply multiplying the thermal profile of the MPSoC by the gain of the regulator and subtracting it from the input average scheduler frequency requirements.

This is the main contribution that makes the major distinction with state-of-the-art techniques for thermal management presented in [11], [8] - [10]. The fact that the design of the optimum frequency controller does not require the run-time temperature profile of the system is a big advantage. Because of this reason, the exact result of the optimization can be obtained for any MPSoC thermal profile by simply multiplying the state vector x by matrix K. In [11] the optimization requires a perfect knowledge of the thermal profile that cannot be made at design time. This limitation undermines the optimality of its results. On the contrary, this method performs an optimum control on a model that perfectly represent the real state the MPSoC, by taking into account in a very exact way the dynamic state of the system.

A. Frequency regulator design phase

The state feedback controller gain K from literature [27] can be calculated in the following way:

$$K = (B_n^T S B_n + R)^{-1}(B_n^T S A_n) \tag{9}$$

where S is the infinite horizon solution of the discrete-time Riccati equation associated to this system [27]. A_n and B_n are equivalent to matrixes A and B of Equation 1, to model the same discrete-time system for the different sampling time used to apply the policy (i.e. 100ms, see section 5 for more details). The solution of the previous equation exists only if matrix Q is positive semi-definite and matrix R is positive definite. In addition, it has been proved [27] that if our system is stabilizable and detectable, by minimizing the cost function, we make also the system stable.

B. Thermal balancing runtime phase

If the chip maximum temperature is under a predefined threshold during the on-line optimization phase, the optimum frequency assignment to achieve thermal balancing is calculated using the following equation:

$$u(k) = f_{avg}(k) - K \cdot x(k) \tag{10}$$

where at time k, $x(k)$ is the current state from equation 1 and $f_{avg}(k)$ the current average frequency constraint required in order to fulfill performance requirements. The number of multiplication and additions N_{op} required every time the policy is applied at runtime, is given by the following equation:

$$N_{op} = n \cdot p \tag{11}$$

where n is the number of cells of each layer of the floorplan and p is the number of cores of the system. All these operations are required every time T_{pol} the policy is applied (typically every 100ms). Moreover the time required to execute all the N_{op} operations should be small compared with both T_{pol} and the time required by the chip to change significantly its thermal profile T_{prof}. The value of T_{prof} depends on chip floorplan technological parameters and can be estimated using cycle accurate thermal simulators such as the ones presented in [6] and [21].

According to the previous considerations, and to our experimental model (for more details see next section) where the number of cores p equals 8 and n equals 30, the number of required multiplications and additions equal to $30 \cdot 8 = 240$. This operations need to be performed every $100ms$. In order to calculate power and area cost of the just designed control system, we use the circuits implementation data provided by [12], [26] and [23]. In our case we choose to have 4 subthreshold multipliers in parallel, bringing to a multiplication delay of $1.28ms$ and an overall area occupation (including sensors, multipliers, adders and the additional wiring and control logic) negligible to the one of the chip (less than $1mm^2$). Moreover the multiplication can be computed using a look-up table stored in an on-chip memory. This can be accomplished if the number of thermal cells is small and a certain degree of approximation in the thermal profile is accepted.

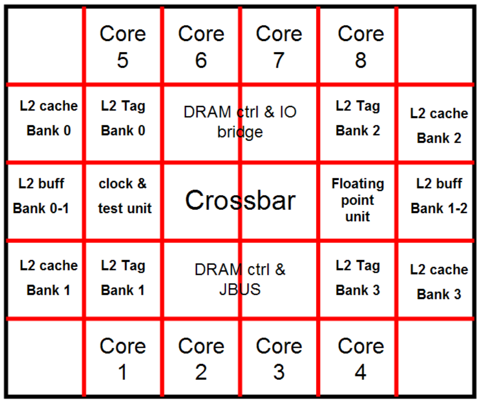

Fig. 3. The Simplified floorplan of a MPSoC architecture resembling the Suns Niagara MPSoC [22], [2].

V. Experimental results

A. Experimental setup

For the experiments, we consider an architecture resembling the 8-core Niagara architecture from Sun Microsystems [22], [2], which has a size of $378mm^2$. The floorplan of the architecture is presented in Figure 3. As this figure shows, we modelled the floorplan in order to have more or less the same effective size (empty areas are not considered in the simplification process). The floorplan has been modelled using blocks of 3mm side each.

The architecture has a maximum operating frequency of 1.2 GHz and the maximum power consumption of each processor core at this frequency to be 4 W [2]. In order to implement the voltage and frequency scaling techniques, we use 5 working frequencies and voltages in the range from 1.2-0GHz. The total power consumption of all the other elements of the MPSoC has been assumed to be 30% of the power consumption of the processing cores, according to [2]. Values regarding thermal resistances, silicon thickness and copper layer thickness have been taken from [24], [25] and [2]. The policy is applied every $100ms$ while the simulation step for the discrete time integration of the RC thermal model has been set to $200\mu s$.

With respect to thermal balancing, in our set-up we focus more on keeping the thermal profile uniform rather than minimizing power consumption. Thus in Equation 4 we minimize thermal unbalancing while respecting a certain maximum power limit. To simulate the system we use the execution characteristics of tasks from a mix of different benchmarks, ranging from web-accessing to playing multimedia [17].

B. Comparisons and results

In this section we compare the proposed enhancements and solutions with previous ones using the simulation environment just described. More specifically, we assess the optimality of the proposed control theory based method for thermal balancing with respect to the following policies and techniques:

- Convex Optimization [11] using an 8X8 table: 8 frequencies values analyzed and 8 temperature values analyzed.
- Convex Optimization [11] using a 32X32 table: 32 frequencies values analyzed and 32 temperature values analyzed.

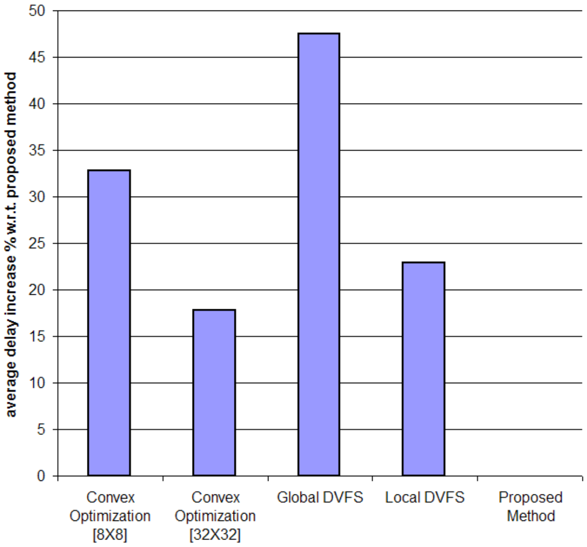

Fig. 4. Delay of tasks before execution. Percentage increase in average delay compared with the proposed method.

- Global DVFS: this technique matches the application performance level with the frequencies of the cores. The temperature control is performed when the maximum temperature of the chip reaches the threshold value of $375^\circ K$. In this case the overall system frequency is reduced by 50% until both the next time the policy is applied and the maximum temperature is inside the safety region.

- Local DVFS: it is exactly like the Global DVFS except that only the frequency of the core exceeding the temperature threshold level is reduced. Thus the overheating in some part of the chip does not have impact on the overall system, improving its performance.

B.1 Performance analysis

The first set of experiments compares the performance of the 8-core Niagara MPSoC when different thermal control and balancing techniques are applied. Figure 4 shows the average increase in waiting time of tasks for the different policies in relation to the proposed method. This figure shows how the proposed technique outperforms previous ones while satisfying temperature constraints all over the MPSoC. Up to 32.8% improvement is obtained by the proposed method respect to the 8x8 table convex technique proposed in [11]. Up to 47.5% delay reduction can be noted comparing with a global DVFS approach that does not provide any thermal balancing warranties and only prevents thermal runaway, as already outlined by [11]. Among all compared techniques, the best performance in average task delay before execution is offered by a 32x32 table based convex technique, which is still 17.7% worse compared with the proposed method.

Furthermore, it is important to notice that convex-based techniques and the proposed method study the problem from a global point of view ensuring that every single subpart of the MPSoC fulfills performance requirements while ensuring a safe thermal behavior of the system. On the contrary, DVFS based techniques do not analyze the system from a theoretical point of view, and they only react at the run-time thermal state of the system every time the policy is applied. Because of this fact, they do not perform any thermal balancing or ensure from a theoretical viewpoint that temperature constraints are fully satisfied during the overall system operation [11]. This is the main reason of their worst performance compared to the proposed method or convex based techniques.

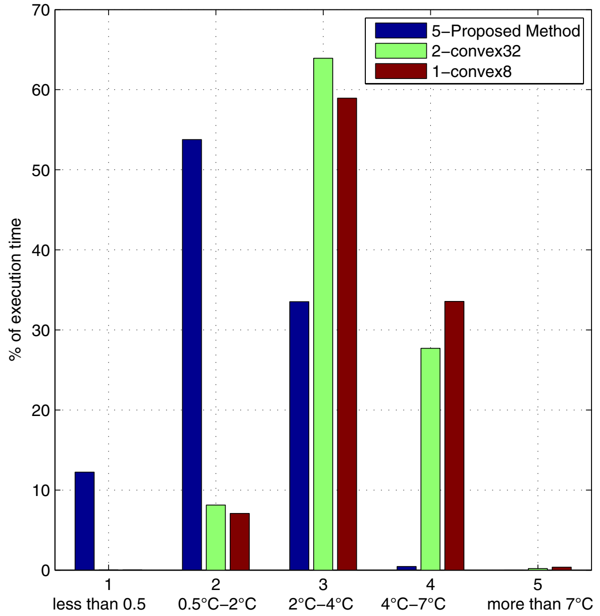

Fig. 5. Statistics related to the maximum temperature between the cores of the 8-core Niagara MPSoC.

B.2 Thermal balancing analysis

We focus now on the convex-based methods [11] and the proposed approach. Thus, in this set of experiments, we compare maximum absolute temperature differences between all cores and derive statistical informations about their distribution. The results are shown in Figure 5.

The chip, according to the ideal thermal balancing target, should stay the highest percentage of time possible in bins with small temperature variations. As Figure 5 shows, all convex-based techniques show worse thermal balancing capabilities compared with the proposed one. Indeed, the proposed method is able to keep temperature differences among cores lower than $2^\circ C$ for more than 66% of the time. This is 8x more than convex based techniques. In addition to that, the regulator, in contrast to convex based techniques keeps the temperature between cores always below $7^\circ C$. Moreover, temperature differences between $4^\circ C$ and $7^\circ C$ are greatly reduced (less than 0.5%) compared with a 27% and 33% of convex-based methods. Thus Figure 5 shows that the proposed method offers better overall thermal balancing and prevents significantly better potential hotspots inside the MPSoC.

Figure 6 shows the maximum run-time temperature difference between two cores for the convex policy and the proposed method. As this figure shows, the proposed method is much more efficient in reducing temperature differences, achieving always values in the range between $0^\circ C$ and $4^\circ C$. Nonethe-

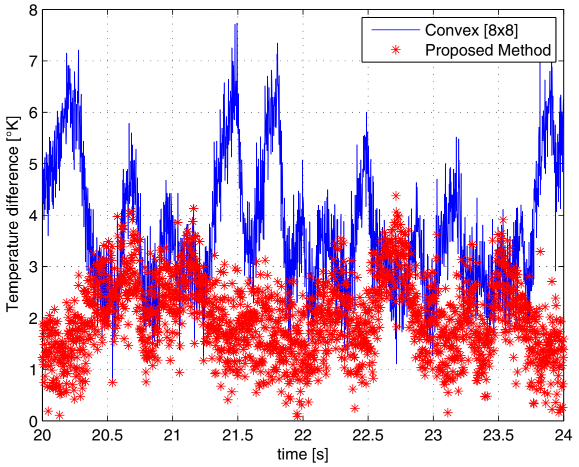

Fig. 6. Maximum temperature difference between two cores during run-time execution for the proposed policy and the convex one.

less, the convex method shows differences larger than 7°C.

The reason of this improved thermal balancing is given by the higher level of accuracy of our control theory based approach to capture the run-time thermal profile of the MPSoC and build the optimum frequency regulation system.

VI. Conclusion

Reducing hot-spots and achieving thermal balancing are two important challenges facing the MPSoC designers. In this work, we model the thermal behavior of an MPSoC as a control theory problem, and then, we propose an optimum solution to the frequency assignment problem for thermal balancing based on a linear quadratic regulator. We have compared the proposed approach with state-of-the-art thermal management methods on a industrial 8-core MPSoC platforms running real SoC benchmarks.

Our results show that, from the performance point of view, the proposed control theory approach achieves better performance figures than other approaches. Namely results show a 32.84% improvements in the task waiting time compared to an 8x8 table convex technique and a 47.5% improvement compared with a global DVFS scheme. Moreover, from the reliability point of view, the thermal profile of the MPSoC, when the proposed policy is applied, shows temperature differences among cores lower than 2°C for more than 66% of the time, which is 8× more than the best thermal balancing technique (i.e., convex-based optimization) presented in the literature. Hence, the overall statistical distribution of thermal profile differences over time is improved significantly with the presented approach in comparison to other state-of-the-art approaches.

Acknowledgements

This work is partially supported by the Swiss SNF Research Grant 20021-109450/1.

References

[1] D. Pham, et al., *Design and Implementation of a First-Generation Cell Processor.*, Proc. of ISSCC, 2005.

[2] P. Kongetira, et al., *Niagara: A 32-way multithreaded SPARC processor.*, IEEE Micro, 2005.

[3] Tilera Corporation, *Tilera's 64-core architecture*, www.tilera.com/products/processors.php, 2007.

[4] O. Semenov et al., *Impact of self-heating effect on long-term reliability and performance degradation in CMOS circuits*, IEEE Transactions on Devices and Materials, 2006.

[5] S. Borkar, *Design challenges of technology scaling*, IEEE Micro, 1999.

[6] K. Skadron, et al., *Temperature-aware microarchitecture: Modeling and implementation*, TACO, 2004.

[7] J. Srinivasan, et al., *The case for lifetime reliability-aware microprocessors*, Proc. of ISCA, 2004.

[8] R. Mukherjee, et al., *Physical aware frequency selection for dynamic thermal management in multi-core systems*, Proc. of ICCAD, 2006.

[9] D. Brooks, et al., *Dynamic thermal management for high-performance microprocessors*, proc. of HPCA, 2001.

[10] C J. Hughes, et al., *Saving energy with architectural and frequency adaptations for multimedia applications*, Proc of MICRO, 2001.

[11] S.Murali, et al., *Temperature Control of High Performance Multicore Platforms Using Convex Optimization*, Proc. of DATE, 2008.

[12] P.Ituero, et al.,*Leakage-based On-Chip Thermal Sensor for CMOS Technology*, Proc. of ISCAS, 2007.

[13] J. Donald, et al., *Techniques for multi-core thermal management: Classification and new exploration*, Proc. of ISCA, 2006.

[14] F. Bellosa, et al., *Event-driven energy accounting for dynamic thermal management.*, Proc. of COLP, 2003.

[15] P. Chaparro, et al., *Understanding the thermal implications of multi-core architectures.*, IEEE TPDS, 2007.

[16] W. Hung et al., *Thermal-aware allocation and scheduling for systems-on-chip.*, Proc. of DATE, 2005.

[17] A. K. Coskun, et al., *Temperature Aware Task Scheduling in MPSoCs*, Proc. of DATE, 2007.

[18] T.-Y. Wang, et al., *3-d thermal-adi: A linear-time chip level transient thermal simulator.* IEEE TCAD, 2002.

[19] B.Goplen, et al., *Efficient thermal placement of standard cells in 3d ics using a force directed approach.*, Proc. of ICCAD 2003.

[20] Y. Zhan, et al., *Fast computation of the temperature distribution in vlsi chips using the discrete cosine transform and table look-up.*, Proc. of ASPDAC 2005.

[21] G. Paci et al., *Exploring temperature-aware design in low-power MPSoCs*, Proc. of DATE, pp. 838-843, 2006.

[22] A. Leon, et al., *A power-efficient high-throughput 32-thread SPARC processor.* Proc. of ISSCC, 2006.

[23] Soeleman, et al., *Ultra-low power digital subthreshold logic circuits,* Proc. of ISLPED, 1999.

[24] M.-N. Sabry, *High-precision compact-thermal models.* IEEE Transactions on Components and Packaging Technologies, 2005.

[25] E.C. Samson, et al., *Interface material selection and a thermal management technique in second-generation platforms built on Intel Centrino Mobile Technology*. Intel Technology Journal, 2005.

[26] B. Paul, et al., *An 8x8 sub-threshold digital CMOS carry save array multiplier*, Proc. of ESSCC, 2001.

[27] G. F. Franklin, et al., *Digital Control of Dynamic Systems*, Mc-Graw Hill.

[28] K. Skadron et al., *Thermal-rc modeling for accurate and localized dynamic thermal management*, Proc. of IEEE HPCA, 2002.

[29] A.Radke, et al., *A survey of state and disturbance observers for practitioners*, Proc. of ACC, 2006.