Multicore Thermal Management with Model Predictive Control

Francesco Zanini†, David Atienza⋆, Luca Benini‡, Giovanni De Micheli†
† Laboratory of Integrated Systems, EPFL, Switzerland;
⋆Embedded Systems Laboratory, EPFL, Switzerland;
‡Micrel Laboratory, DEIS , University of Bologna, Italy
e-mail: {name.surname}@epfl.ch, lbenini@deis.unibo.it

ABSTRACT

The goal of thermal management is to meet maximum operating temperature constraints, while at the same time tracking time-varying performance requirements. Current approaches avoid thermal violations by forcing abrupt operating points changes (e.g. processor shutdown), which cause sharp performance degradation. In this paper we aim at achieving a smooth thermal control action, that minimizes the variance of performance tracking error. We formulate this problem as a discrete-time optimal control problem, which can be solved using the theory and computational tools developed in the field of model-predictive control. Our optimization process considers the thermal profile of the system, its evolution over time, and time-varying workload requirements. Experimental results show that the proposed approach offers significant thermal balancing improvements over previous methods.

1. INTRODUCTION

With the advance of technology, the number of cores integrated on a chip is increasing. Today, several multicore architectures are already commercially available, such as Sun's Niagara [1] and Tilera's 64-core architecture [2]. Power and thermal management are critical challenges for high-end multicore systems [4]. Temperature gradients and hot-spots affect system performance and lead to reduced chip lifetimes [3].

In the last years, thermal management techniques received a lot of attention. Many state-of-the-art thermal control policies manage power consumption via *dynamic frequency and voltage scaling* (DVFS) [5]-[8]. DVFS can be targeted to power density reduction, which has the effect of reducing overall temperature [5]. Then, thermal control policies avoid violations of temperature bounds by transitioning processors in low-power modes, taking a performance hit to cool down.

Unfortunately, not only high temperature, but also thermal cycles raise the failure rate of the system [11]. In addition, abrupt power-mode transitions due to DVFS waste power [13]. Hence, smooth thermal control, which eliminates abrupt power-mode transitions and large thermal cycles is highly desirable.

In this work, we propose a novel closed-loop thermal management policy yielding a smooth optimum control on working frequencies and voltages of multicore systems, while satisfying max-temperature and performance constraints, and minimizing power (Figure 1). The problem is modelled using a control-theoretic approach based on *model predictive control* (MPC) [14]. This method provides optimum solutions for linear dynamic systems subject to constraints. It has several advantages with respect to state-of-the-art convex-based solutions for thermal control, such as the one presented in [8]. First, the MPC-based approach achieves a smoother control; thus, a reduction in performance fluctuations and losses is observed. Second, the system can manage different workload requirements for different cores using two approaches to implement the controller, defining computation vs. storage tradeoffs. The first one (implicit MPC) solves on-line the optimization problem thanks to an embedded numerical solver. The second one (explicit MPC) computes the solution on-line by multiplying the vector containing current thermal profile information and workload requirements by precomputed coefficients.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...$5.00.

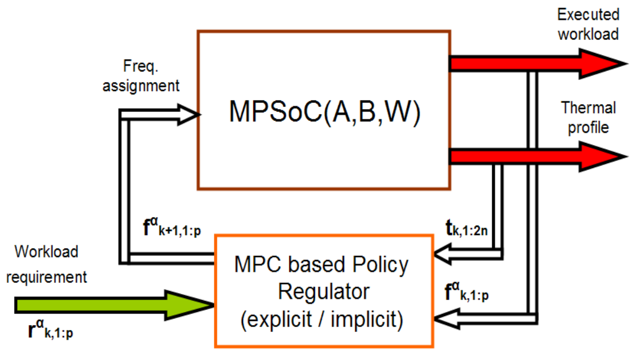

Figure 1: Proposed MPC-based thermal management policy

Our results show that the proposed MPC-based method guarantees that scenarios with dangerous thermal profiles are avoided while satisfying the application performance requirements. Moreover, thanks to the improved workload handling capabilities and the smooth transitions on both voltages and frequencies, the policy significantly reduces temperature profile variations over time. Thus, our proposed method achieves 2.5× to 5× improvements (for several control smoothness indexes) over state-of-the-art convex-based thermal management policies [8].

The remainder of this paper is organized as follows. Section 2 revises related work on thermal control techniques. Section 3 presents our control model and the novel thermal management policy. Section 4 describes our experimental setup. Section 5 presents the experimental results and compares our approach with state-of-the-art thermal management approaches. Finally, Section 6 summarizes the main conclusions of this work.

2. PREVIOUS WORK

Many researchers have recently focused on power management and thermal control for multicore systems and *Multi-processor Systems-on-Chip* (MPSoCs) [6]. Processor power optimization using DVFS have been proposed in several works [7], [6]. Then, [13] tries to minimize abrupt changes in power mode transitions by solving the frequency assignment problem from a frequency prediction perspective [5]. These techniques reduce power density and overall temperature, but not necessarily thermal gradients and hot-spots.

In [8], convex optimization is used to solve the DVFS assignment problem considering power and hotspot minimization. The convex optimizer computes processor frequencies which minimize the gap between required and provided performance, subject to the operating temperature constraint. The main drawback is that it does not adapt smoothly to changes in performance requirements, leading to abrupt changes in processor DVFS assignments.

In contrast, the optimal MP controller considers the future thermal trajectory of the system, based on current workload and thermal state, hence it adjusts core frequencies in a smooth way to avoid reaching the maximum temperature constraint and taking abrupt corrective actions.

3. THERMAL MPC

In MPC, a state-space model of a linear dynamic system is needed as input to the calculation of the optimal controller. In our case, the model links frequency (and voltage) assignment to spatial temperature profile of the die over time, as described next.

3.1 Heat propagation model

Our thermal model is based on finite-element analysis [5]. Two layers have been used on the vertical direction, i.e., the silicon and the heat spreading copper layer. Then, the chip floorplan has been divided into several thermal cells of cubic shape, and every single functional unit in the floorplan can be represented by one or more thermal cells of the silicon layer. Thermal modelling is computed considering the heat conductances G and capacitances C of the cells, as computed in [5]. Thus, the differential equations modelling the heat flow are obtained in terms of the equivalent RC network. The solution of the system has the form:

$$t_{k+1,i} = t_{k,i} + \sum_{\forall j \epsilon Adj_i} a_{i,j}(t_{k,j} - t_{k,i}) + b_i p_{i,k} \qquad (1)$$

where $t_{k,i}$ is the temperature value of the cell i at time k. Constants $a_{i,j}$ and b_i are characteristic of the thermal behavior of the chip, and can be calculated as in [5]. Then, Adj is the adjacency matrix of the blocks inside the floorplan, and $p_{i,k}$ is the power consumption of block i at time k. Overall, the system described by Eq. 1 is linear, discrete-time and can be approximated as time invariant by making a worst case hypothesis on the thermal conductivity coefficient dependence over temperature.

Given the link between temperature distribution and power consumption, established by Eq. 1, we can easily tie processor operating frequencies to die temperature distribution:

$$t_{k+1,1:2n} = A[t_{k,1:2n}] + B[f^{\alpha}_{k,1:p}] + W \qquad (2)$$

where n is the number of blocks composing the floorplan for the two layers and p is the number of cores. Then, at time k, the temperature of the next simulation step of cell i $t_{k+1,i}$ can be computed using Eq 2. The working frequency of core j is $f^{\alpha}_{k,j}$. The coefficient α expresses the dependence between the frequency and the core's power dissipation. If $\alpha = 1$, we have a linear dependence (i.e., frequency scaling) while if $1 < \alpha \leq 2$ we obtain a quadratic or sub-quadratic dependence (i.e., voltage and frequency scaling) [8]. Matrices A and B describe the heat propagation properties of the MPSoC. Finally, W is an offset vector considering the room temperature effect in the heat spreading process.

3.2 MPC-based policy model

MPC [14] is an optimal control approach aimed at maximizing a performance metric for a linear dynamic system under input/output constraints. The solution of the optimization problem provides the feedback control actions, and can be either computed by embedding a numerical solver in the real-time control code (implicit solver), or pre-computed off-line and evaluated through a look-up table of linear feedback gains (explicit solver). Figure 1 shows the block diagram of our MPC-based policy.

The regulator monitors the MPSoC state at each time instant k. Thus, the state is defined as a vector composed by temperature values $t_{k,1:2n}$ and working frequencies $f^{\alpha}_{k,1:p}$. Temperatures are monitored by on-die sensors [16], while working frequencies are known and controlled by the regulator.

The regulator receives from higher-level software layers (e.g., operating system or OS) a workload requirement, expressed as a vector of required operating frequencies $r^{\alpha}_{k,1:p}$ for all the the p cores of the MPSoC. The regulator provides a frequency assignment that minimizes the *quadratic tracking error* $e_{k,1:p}$. This error, defined by Eq. 4, is proportional to the difference between the offered and required workload. Note that *the tracking error is a direct measure*

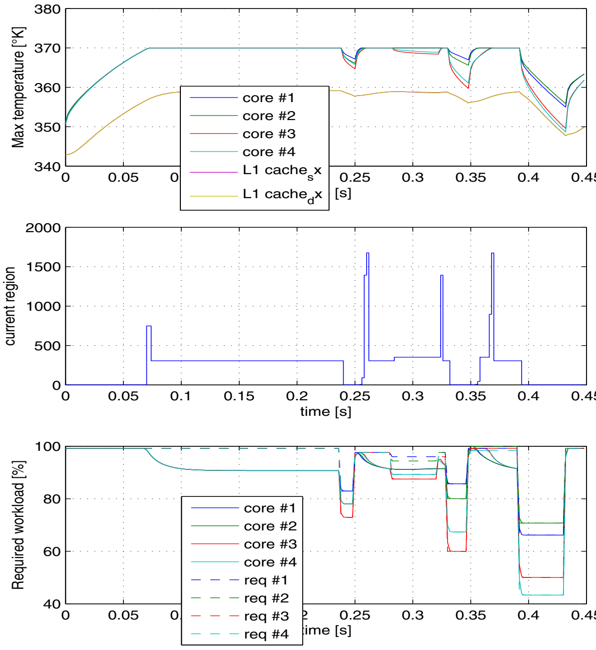

Figure 2: The explicit MPC method

of performance penalty, as it is greater than zero when the controller sets processor working frequencies not exactly matching the requests coming from the OS.

Constraints on the maximum temperature of the MPSoC are also enforced in the optimization process. Then, the optimal control problem is formulated over an interval of h time steps, which starts at current time k. For this reason, the approach is said predictive. The result of the optimization is an optimal sequence of future control moves (i.e., frequency settings for the cores).

Only the first sample of such a sequence is actually applied to the process; the remaining moves are discarded. At the next time step, a new optimal control problem based on new temperature measurements and required frequencies is solved over a shifted prediction horizon. Such a "receding-horizon" [14] mechanism represents a way of transforming an open-loop design methodology (i.e., the convex based policy proposed in [8]) into a feedback one, as at every time step the input applied to the process depends on the most recent measurements. Formally, the optimization is defined as:

$$min \sum_{j=0}^{h-1} (e_{k+j,1:p}) \cdot S \cdot (e_{k+j,1:p}) \qquad (3)$$

where h defines the prediction horizon, S is the identity matrix and

$$e_{k+j,i} = f^{\alpha}_{k+j,i} - r^{\alpha}_{k,i} \qquad (4)$$

The minimization process has to satisfy the following constraints:

$$0 \leq t_{k+1,1:2n} \leq t_{max}, k = 0...h-1 \qquad (5)$$

$$0 \leq f^{\alpha}_{k+1,1:p} \leq f^{\alpha}_{max}, k = 0...h-1 \qquad (6)$$

where f_{max} and t_{max} are the maximum working frequency and allowed temperature in the system.

3.3 Implicit vs. Explicit Regulator

The proposed control strategy can be implemented in two different ways. The first one is called implicit and requires to solve on-line the minimization problem of Eqs. 3-6 every time the policy is applied. Thus, a large amount of hardware resources are needed, since the result must be computed in a time frame shorter than the thermal time constants of the MPSoC.

However, an alternative approach has been proposed in [14]. In this case, the *quadratic programming (QP)* problem is solved off-line in a way that makes explicit the dependence of the solution of the frequency assignment problem $f^{\alpha}_{k+1,1:p}$ on input parameters

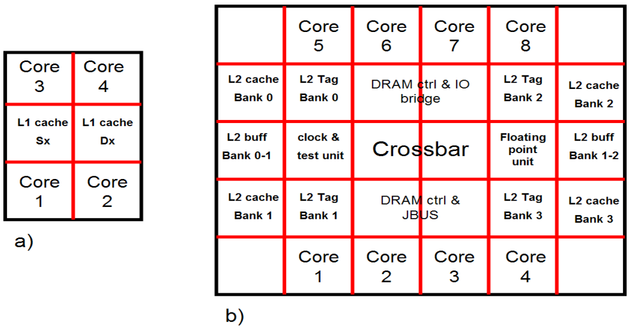

Figure 3: Floorplans of: a) generic 4-core processor; and b) the Niagara-1 multicore MPSoC

$f^{\alpha}_{k,1:p}$, $r^{\alpha}_{k,1:p}$ and $t_{k,1:2n}$. Bemporad et al. have shown that the optimal explicit controller is *piecewise affine*. In other words, the state space can be divided by in a set of regions, bounded by linear inequalities (i.e., a polytope), and in each region a different linear controller can be specified and computed off-line [14].

Then, the controller selection can be efficiently performed on-line by simply checking region boundaries. The resulting controller structure is defined in any of the M partitions as follows:

$$[f^{\alpha}_{k+1,1:p}] = F_i \begin{bmatrix} t_{k+1,1:2n} \\ f^{\alpha}_{k,1:p} \\ r^{\alpha}_{k,1:p} \end{bmatrix} + g_i \;\; if \;\; H_i \begin{bmatrix} t_{k+1,1:2n} \\ f^{\alpha}_{k,1:p} \\ r^{\alpha}_{k,1:p} \end{bmatrix} \leq K_i \tag{7}$$

where the matrix F_i and vector g_i are the gain and offset coefficients of the i^{th} region. Each region is identified by affine inequalities defined by the matrix H_i and vector K_i in 7.

If the partitions are properly stored, the number of operations depends logarithmically on the partitions [18]. Nonetheless, while the computer code for evaluating MPC in the explicit form is certainly simpler than the code embedding the QP solver, from the point of view of memory requirements, the explicit form may be more demanding, as M and the matrices to be stored in look-up tables are large. Thus, performance vs. area trade-offs exist between the implicit and the explicit form of the MPC controller.

To illustrate the memory requirements of an explicit MPC in a real case study, we consider the MPSoC shown in Figure 3b, composed by 4 processing cores and modelled by 12 cells. According to Eq. 7, the resulting explicit controller will span over a 20-dimensional space. 12 dimensions are used by temperature cells to model the thermal behavior of the MPSoC, 4 dimensions are used by current cores frequencies and 4 dimensions model the workload requirements requested from the system by the scheduler. The prediction horizon has been set to 2 (i.e., 8 ms). The resulting controller computed using the tool proposed in [19] has 1817 regions. The number of regions of the controller can be further reduced by performing a merging of regions which contain the same expression of the control law. After this heuristic-based merging, the number of regions is reduced to 1764, while preserving the performance and stability of the original controller.

According to Eq. 7, the number of coefficients N_c to implement the optimum explicit approach of our method is given by the following expression:

$$N_c = N_{reg} \cdot (\overbrace{2p(p+n)}^{F_i} + \overbrace{p}^{g_i} + \overbrace{2N_{avg} \cdot (p+n)}^{H_i} + \overbrace{N_{avg}}^{K_i}) \tag{8}$$

N_{reg} is the number of regions of the explicit controller. p is the number of processing cores and n the number of thermal cells of the silicon layer. N_{AVG} is the average number of affine inequalities that identifies each region. In this equation, the symbol $\overbrace{\quad}$ highlights contributions for each of the matrices in Eq. 7. In relation to our example, the average number of affine inequalities for each region is 38.2, leading to $N_C = 886.2$ coefficients to be stored per region, for a total of $1.56E6$ coefficients. Overall, these values are low for latest on-chip memories and these coefficients can be reduced by using approximate approaches [16].

The computational complexity of the evaluation can be computed from [18] and 7 and, assuming a worst case scenario, is provided by the following equation:

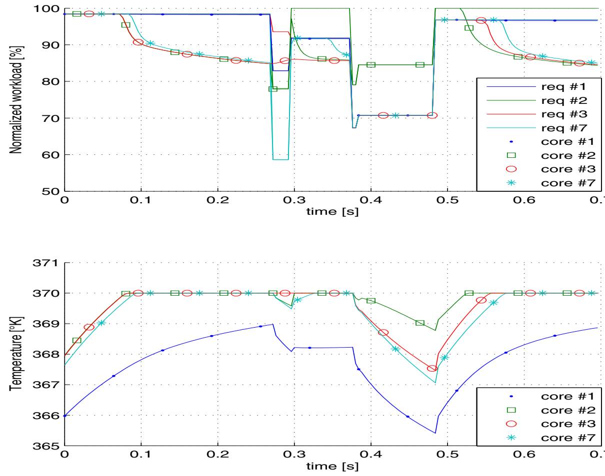

Figure 4: Behavior of MPC method for variable workloads

$$N_{comp} = 1.7 \cdot log_2(N_{reg}) \tag{9}$$

$$N_{Mult-Add} = 2p(p+n) \tag{10}$$

where N_{comp} is the number of required comparisons and $N_{Mult-Add}$ is the number of required multiplications/additions. In our case study, $N_{comp} = 18.3$ and $N_{Mult-Add} = 80$. All these operations are required every time T_{pol} the policy is applied (in this case every 4 ms). Moreover the time required to execute all the 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, like [5]. According to previous considerations, the minimum clock frequency of the multiplier required to implement the proposed method in our example is 40 KHz. Then, the area occupation to implement the proposed MPC-based method is dominated by the look-up table that stores the coefficients of Eq. 8. Figure 2 shows the run-time behavior of the explicit controller.

The temperature threshold has been set to $370°K$. Each core consumes a maximum power of $4W$, the chip has an area of $6mm^2$. Working frequencies are in the range 0-1.2 GHz. The other elements of the MPSoC consume 30% of the power consumption of the processing cores [1]. Thermal resistance, silicon thickness and copper layer thickness have been derived from [9, 10, 1]. The OS produces a different time-varying load for each core executing tasks ranging from web-accessing to playing multimedia [17].

Looking at Figure 2, at 0.05s and at 0.2s, a high workload is required to all cores. In the first case, since the maximum chip temperature is below the threshold, the system is able to fulfill its performance requirements. In the second case, the proposed controller clocks the cores with a frequency that respects the maximum on-chip temperature for this example, fixed to $370°K$. Furthermore, the MPC-based controller generates a very smooth transition between the two frequency modes. This smoothness is important to reduce thermal cycling that raises the failure rate of the system [11]. An example of how the proposed controller takes into account thermal properties of the MPSoC is reported at 0.35s where the system decreases the frequency of the hottest cores first and later that of less temperature-critical cores. Moreover, this explicit technique is optimal and no approximation to simplify the controller has been made. Hence, both the implicit and explicit implementations of the controller provide the same output and performance.

4. EXPERIMENTAL RESULTS

In this section we consider as MPSoC case study the 8-core Niagara-1 (UltraSparc T1) chip from Sun Microsystems [1], which has similar power consumption, voltage and frequency range for each processor as those used in the case study of the previous section. The Niagara-1 floorplan is shown in Figure 3b, which has

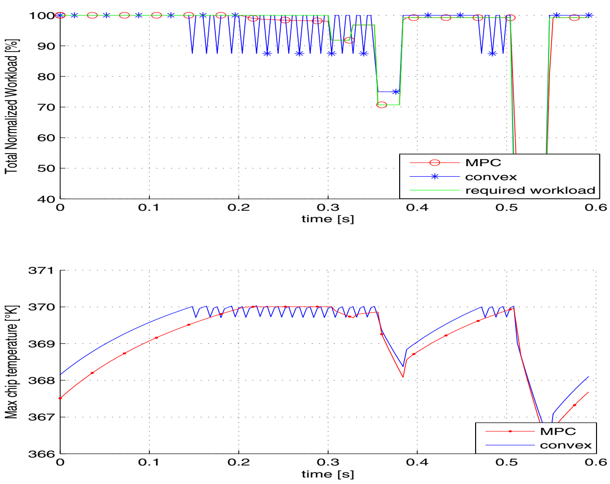

Figure 5: Time comparison of MPC vs. convex

been modelled using cubical blocks of 3mm. As software benchmarks, we have used mixes of tasks ranging from web-accessing to playing multimedia [17]. In our experiments, our MPC-based thermal management policy is applied every 4 ms, while the simulation step for the discrete time integration of the thermal model is 200μs. The MPC policy tracks workload requirements, minimizing power consumption while respecting a maximum temperature limit of $370^\circ K$. We have used as MPC design tool the Matlab-based development platform provided by [15].

4.1 Handling different workload scenarios

In Figure 4 we report the time domain plot of the normalized workload and temperature of each core. The normalized workload is proportional to the frequency setting of the MPSoC, and variations in its value produce cores voltages and frequencies changes.

In Figure 4, the OS requires different workloads for each core in a very unbalanced way. Cores 1 to 4 are assigned a workload higher than 4 to 8 and core 2 is required the highest workload (see for example at 0.35s). In the top graph of Figure 4, the first 4 curves are required workload by the scheduler for Cores 1, 2, 3 and 7. Then, the last 4 curves show the workload offered by the MPSoC under the MPC policy control. As Figure 4 depicts, the controller produces a smooth control and never makes the maximum chip temperature exceed the $370^\circ K$ threshold. Thus, in case of potential overheating, the regulator decreases the frequency of the hottest cores to achieve maximum performance while respecting temperature constraints, as shown at 0.45s.

4.2 MPC- vs. convex-based thermal control

We compare our MPC-based thermal management method with state-of-the-art convex-based thermal management techniques, in the case of time-varying workload requirements. We have implemented the convex optimization technique described in [8] using an 8x8 table. We chose the frequency to satisfy the maximum workload requirements that respects the maximum chip temperature constraint. Our MPC-based policy was conservatively implemented using a short time horizon of 3 (12 ms) to have a low-cost (from the computation viewpoint))thermal management policy.

The results are reported in Figure 5, which show that both techniques satisfy the maximum temperature constraint. However, while MPC is able to track input frequency requirements, convex optimization is not able to follow accurately the required workload when the chip temperature is close to the threshold and maximum performance is required, as shown at 0.2s in Figure 5. Thus, our MPC-based method provides the maximum required workload while the convex method provides a workload 15% smaller than the required one. Moreover, temperature and frequency behavior over time are smoother for the MPC policy because the convex-based one performs an optimization based only on actual scheduler frequency requirements and chip thermal profile, while the MPC approach optimizes performance according to a predictive model of the future trajectory of the MPSoC.

Although the MPC method requires a higher computation complexity than the Convex approach it is very limited (i.e., few milliseconds of standard processing cores, as quantified by equations 9 and 10), which is negligible with respect to heat propagation speed in current MPSoCs.

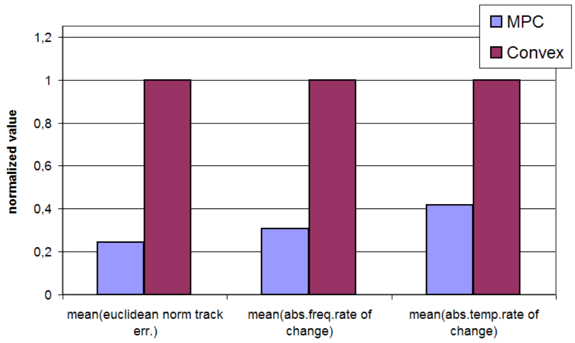

Figure 6: Normalized statistical comparison MPC vs. convex

We have finally quantified (from a statistical point of view) the improvements of the MPC policy with respect to convex-based thermal management, by analyzing the tracking error and smoothness in temperature and frequency variations. First, we have quantified the tracking error using the mean value of its Euclidean norm. Then, we have estimated the smoothness of both MPC-based and convex-based policies as the mean of the frequency/temperature absolute value of change rate. Figure 6 shows results normalized to the convex-based policy, which indicate that the MPC-based policy outperforms the convex-based approach between 2.5× and 5×.

5. CONCLUSIONS

We have presented a novel MPC-based thermal control policy. This policy respects temperature constraints while ensuring a smooth control on the temperature; thus, performance losses and thermal fluctuations are minimized in the target MPSoC. The solution of the MPC optimization problem provides the feedback control action, and can be computed on-line (implicit solver) or off-line (explicit solver). Our experiments show that this new MPC-based thermal management approach improves on several tracking error metrics ranging from 2.5× to 5× with respect to state-of-the-art thermal management solutions.

6. REFERENCES

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

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

[3] O. Semenov et al.,*Impact of self-heating effect on long-term reliability and performance degradation in CMOS circuits*, IEEE T-D&M, 2006.

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

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

[6] D. Brooks et al.,*Dynamic thermal management for high-performance microprocessors*, Proc. HPCA, 2001.

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

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

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

[10] 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.

[11] J. Haase et al.,*Reliability-aware power management of multi-core processors*. Proc. DIPES, 2006.

[12] JEDEC, *Failure mechanisms and models for semiconductor devices*, Jedec Solid State Tech. Association, 2003.

[13] H. Jung et al.,*Continuous Frequency Adjustment Technique Based on Dynamic Workload Prediction*, Proc. VLSI Design, 2008.

[14] A. Bemporad et al.,*The explicit linear quadratic regulator for constrained systems*, Automatica, 2002.

[15] A. Bemporad *Hybrid Toolbox for Matlab*, 2003-2007.

[16] S.O.Memik, R.Mukherjee, M.Ni and J.Long, *Optimizing Thermal Sensor Allocation for Microprocessors*. IEEE T-CAD, 2008.

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

[18] P. Tøndel, et al., *Evaluation of piecewise affine control via binary search tree*, Automatica, 2003.

[19] M. Kvasnica, et al, *Multi-Parametric Toolbox (MPT)*, 2006.