Multicore thermal management using approximate explicit Model Predictive Control

Francesco Zanini†, Colin N. Jones‡, David Atienza⋆, Giovanni De Micheli†
† Laboratory of Integrated Systems (LSI), EPFL, Switzerland
⋆Embedded Systems Laboratory (ESL), EPFL, Switzerland
‡ Automatic Control Laboratory, ETH Zurich, Switzerland
e-mail: {name.surname}@epfl.ch, cjones@control.ee.ethz.ch

***Abstract*— Meeting temperature constraints and reducing the hot-spots are critical for achieving reliable and efficient operation of complex multi-core systems. In this paper we aim at achieving an online smooth thermal control action that minimizes the performance loss as well as the computational and hardware overhead of embedding a thermal management system inside the MPSoC. The optimization problem considers the thermal profile of the system, its evolution over time and current time-varying workload requirements. We formulate this problem as a discrete-time control problem using model predictive control. The solution is computed off-line and partially on-line using an explicit approximate algorithm. This proposed method, compared with the optimum approach provides a significant reduction in hardware requirements and computational cost at the expense of a small loss in accuracy. We perform experiments on a model of the 8-core Niagara-1 multicore architecture using benchmarks ranging from web-accessing to playing multimedia. Results show that the proposed method provides comparable performance(loss up to 2.7%) versus the optimum solution with a reduction up to $72.5\times$ in the the computational complexity.**

I. 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 architecture [1]. Power and thermal management are critical challenges for high-end multicore systems [3]. Temperature gradients and hot-spots affect system performance and lead to reduced chip lifetimes [2].

In recent years, thermal management techniques have received a lot of attention. Many state-of-the-art thermal control policies manage power consumption via *dynamic frequency and voltage scaling* (DVFS) [6]- [10]. DVFS can target power density reduction, which has the effect of reducing overall temperature [6]. Then, thermal control policies avoid violations of temperature bounds by transitioning processors in low-power modes, taking a performance hit to cool down.

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

In this work, as in our previous work [15], we propose an online smooth thermal control action to keep the maximum MPSoC temperature under a specific bound. The policy minimizes also the performance loss and the hardware overhead of the proposed thermal management system. The problem is modelled using a control-theoretic approach based on a novel approximate explicit *model predictive control* (MPC) [9]. Here, constraints on the maximum temperature of the MPSoC and on the required workload are enforced in the optimization process. Then, the optimal control problem is formulated over an interval of L time steps, which starts at current time t. For this reason, our approach is predictive [9].

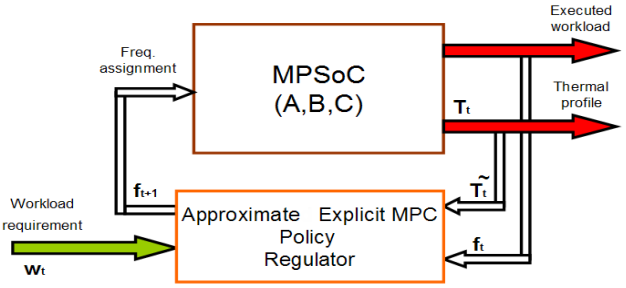

Fig. 1. Diagram of proposed approximate-explicit-MPC-based thermal management policy

In our previous work [15], the policy is computed on-line by multiplying the vector containing current thermal profile information and workload requirements by precomputed coefficients. The main problem with this approach is that the number of coefficients to store is usually large for a complex MPSoC system. As a result, this method can be implemented only in MPSoC described by simplified thermal models using a small number of states and input/output variables. In addition to that the policy had to be formulated over a limited interval of few time steps in the future to keep the implementation of the policy feasible.

In this paper we present a new explicit approximation method, that reduces computational complexity and enables its online implementation. As a result, this approach can be extended to complex MPSoC models without the need of a numerical embedded solver. We perform experiments on a commercial multicore architecture using benchmarks ranging from web-accessing to playing multimedia. Our results, compared with [15] show a reduction up to $45\times$ in the number of coefficients and a reduction up to $72.5\times$ in the computational complexity. The performance loss is negligible compared with the optimum controller. The tracking error has an average loss up to 2.7%. This metric represents the amount of undone work normalized to the total workload request.

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

II. Previous Work

Many researchers have recently focused on power management and thermal control for multicore systems and *Multiprocessor Systems-on-Chip* (MPSoCs). Processor power optimization using DVFS have been proposed in several works [4]. Jung et al. [13] try to minimize very fast changes in power mode transitions by solving the frequency assignment problem from a frequency prediction perspective [6]. These techniques reduce power density and overall temperature, but not necessarily thermal gradients and hot-spots.

Murali et al. [10] use convex optimization 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 very fast changes in processor DVFS assignments.

In our previous approach [15], model predictive control is used as a methodology to solve the frequency assignment problem in an MPSoC. Nevertheless the computational cost of this approach is high. As a result the MPSoC model had to be simplified to make the system feasible from a computational perspective.

III. Proposed Policy

A. System model

The abstraction of the system as a block diagram is shown in Figure 1. The regulator currently monitors the MPSoC state consisting of temperature values and working frequencies. The temperature state at time t is defined as a vector $T_t \in \Re^{2n}$, where $(T_t)_i$ is the temperature of cell i at time t. The thermal model consists of two layers, each one composed by n cells. For this reason the total number of cells representing the MPSoC thermal model is $2n$. The frequency state at time t is defined as a vector $f_t \in \Re^c$, where $(f_t)_i$ is the frequency value of input i at time t and c is the number of inputs. Working frequencies are controlled by the regulator, and are known while temperatures are monitored by on-die thermal sensors. Temperature measurements at time t are defined as a vector $\tilde{T}_t \in \Re^s$, where $(\tilde{T}_t)_i$ is the temperature measurement coming from sensor i at time t. The number of thermal sensors inside the MPSoC is denoted by s. Thus, the current state of the system T_t at time t is generated from data derived from real thermal sensor measurements $\tilde{T}$ on the real MPSoC.

The regulator monitors the workload generated from higher-level software layers (e.g., operating system or OS). At time t, it is defined as a vector $w_t \in \Re^c$. The regulator provides a frequency assignment that minimizes the undone work. This is defined as the difference between the offered and required workload. This quantity has a minimum value equal to zero when the controller sets processor working frequencies exactly matching the requests coming from the OS.

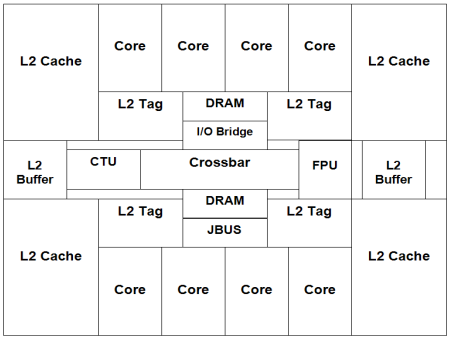

Fig. 2. Floorplan used of the Niagara-1 multicore case study

B. MPSoC thermal model

The thermal model is derived considering the heat conductances and capacitances of the cells as computed and validated in [6] and [5]. The differential equations modelling the heat flow are given by solving this network. The thermal model is slightly nonlinear since coefficients are temperature-dependent (relative error in the order of 0.16%) [5]. To represent the thermal model using a linear, time invariant discrete-time system, the solution of the differential equations modelling the heat flow inside the MPSoC has been linearized. The way the model is described is expensive in terms of computational requirements for high accuracy MPSoC models. The reason is because the model has a number of states equal to the number of thermal cells inside the MPSoC floorplan. This might be large for high-complexity MPSoC.

In this approach, we use a simplifying model using a smaller number of states. To reduce the number of the states, we performed a model reduction using a Gramian-based balancing of state-space realizations [14]. After that, we reduced the order of the state-space model by eliminating the states with corresponding small Hankel singular values. The full MPSoC model is now described by the following system of equations:

$$\tilde{X}_{t+1} = \tilde{A}\tilde{X}_t + \tilde{B}p_t \quad (1)$$

$$\tilde{T}_t = \tilde{C}\tilde{X}_t \quad (2)$$

where, at time t, l is the number of states of the new reduced order model, matrix $\tilde{A} \in \Re^{l\times l}$ and matrix $\tilde{B} \in \Re^{l\times c}$. Equation 1 describes the state update for the reduced order model of the MPSoC. Here the states do not represent directly temperature values inside each cell.

The relation between the power dissipation $p_t \in \Re^c$ and and the frequency of operation f_t is expressed by Equation 3.

$$f_t^\alpha = p_t \quad \forall\ t \quad (3)$$

where the constant α is chosen depending on the technology and usually it varies from 1 to 2. If $\alpha = 1$, we have a linear dependence(i.e., freq. scaling) while if $1 < \alpha \leq 2$ we obtain a quadratic or sub-quadratic dependence(i.e., DVFS) [10].

Matrix $\tilde{C} \in \Re^{s\times l}$. Equation 2 relates the value of the states with temperature measurements in s specific locations inside the MPSoC. In cases where the temperature of the overall MPSoC floorplan has to be controlled, s is set equal to $2n$.

C. Approximation Method

The proposed method aims at minimizing a cost function for a linear dynamic system under constraints. The solution of the optimization problem is solved off-line in a way that makes explicit the dependence of the solution of the frequency assignment problem f_{t+1} on input parameters f_t and $\tilde{T}_t$. The resulting explicit controller is *piecewise polynomial*. 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 [9]. Then, the controller selection can be efficiently performed on-line by simply checking region boundaries. The optimization problem is formalized as:

$$J(w,\tilde{X}) := \min_{f_0,\ldots,f_{L-1}} \sum_{t=0}^{L-1}\sum_{i=1}^{t}(w_i - f_i)$$

$$\text{s.t.} \quad \sum_{i=1}^{t} w_i \leq \sum_{i=1}^{t} f_i \tag{4a}$$

$$\tilde{X}_{t+1} = \tilde{A}\tilde{X}_t + \tilde{B}p_t$$

$$p_t \geq f_t^{\alpha}$$

$$\tilde{C}\tilde{X}_t \leq T_{\max} \tag{4b}$$

$$f_{\min} \leq f_t \leq f_{\max} \tag{4c}$$

$$\tilde{X}_0 = \tilde{X} \tag{4d}$$

The problem minimizes the undone work U_t at time t, $U_t = w_t - f_t$, subject to thermal constraints on each cell (4b), core frequency bounds (4c) and to the causality limitation (4a), which states that workload cannot be completed before it is issued. The convex problem is solved by using the approximation method proposed in [17]. This method produces an approximate controller of any desired complexity and provides a direct trade-off between the level of approximation and the storage requirements.

The proposed method begins by computing an approximate convex *Piece-Wise Affine* (PWA) lower bound of the optimal cost function of (4). Since the approach proceeds in an incremental greedy-optimal fashion, it is possible to stop the process when any desired level of complexity, or approximation accuracy, is reached. The control law is then derived from this lower bound by sampling (4) at the vertices of the bounding function and interpolating using the barycentric technique proposed in [18]. The result is a nonlinear and smooth piecewise polynomial control law. In particular, the algorithm is divided into two main phases.

The first phase of the algorithm iterates two steps. In the first step, we compute the level of approximation and a point that obtains this level. In the second step, the approximation is updated such that the error is maximally reduced around this point. These two steps are iterated until the desired accuracy is achieved. It can be shown that any desired approximation error can be achieved in finite time for any convex function. Within I iterations, the above procedure produces a lower PWA bound f_I of the optimal value function J consisting of I inequalities and an approximation error ϵ.

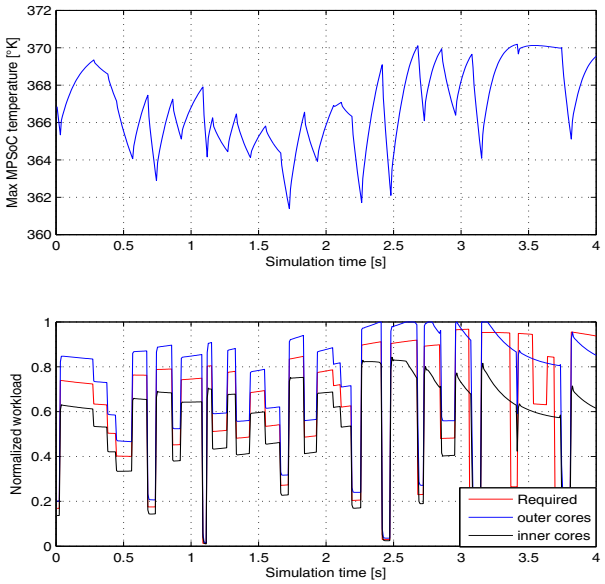

Fig. 3. Run-time simulation of the proposed method

We now apply the second phase of the algorithm, where we compute the optimal solution at the vertices of f_I. We then define a polynomial for each region R_i by interpolation of the optimal control law at the vertices of the region. The result is a smooth, piecewise polynomial control law.

IV. Experimental Results

A. Experimental Setup

As case study, we use the 8-core Niagara-1 (UltraSparc T1) chip from Sun Microsystems [1]. The Niagara-1 floorplan is shown in Figure 2. In this model the number of cores $c = 8$, the number of cells per each layer $n = 30$, the number of states in the reduced model $l = 3$. We used a value of $\alpha = 2$ [10]. As software benchmarks, we have used mixes of tasks ranging from web-accessing to playing multimedia [7]. In our experiments, our MPC-based thermal management policy is applied every 8 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 $370°K$. The prediction horizon $L = 4$.

B. Comparative Simulations

Figure 3 shows the run-time simulation of the proposed method. The maximum MPSoC temperature never exceed the threshold set to $370°K$. The policy splits the average required amount of workload per each core in an unbalanced way. A frequency higher than the average request is assigned to the outer cores. These cores are indeed surrounded by colder regions and so they can dissipate the power in a better way compared with the inner cores. At $3.5s$, the policy is not able to satisfy the average workload requirement since the MPSoC maximum temperature is exceeding the threshold. The frequencies of both the inner and the outer cores are decreased in a smooth way avoiding very fast changes as in [10].

Controller	Number of regions	Number of vertices	Comp. complexity (#operations) search	control law	total	Storage Space (#coefficients) search	control law	total	Perform. (TCK_{err})	Perform. (MU_w)	Time design [s]	Time run [ms]
A-100	1	66	0	858	858	0	366	366	67.34%	83.57%	40.13	4.29
A-200	14	136	30	1025	1055	204	915	119	10.92%	28.08%	46.92	5.27
A-300	32	233	50	1184	1234	1560	1770	3330	7.00%	23.11%	50.75	6.17
A-400	51	328	60	1286	1346	3186	2778	5964	4.56%	19.59%	55.90	6.73
A-500	72	431	60	1335	1395	4986	3771	8757	4.54%	19.59%	63.61	6.97
A-600	87	528	65	1354	1419	7212	4683	11859	4.51%	19.46%	71.46	7.09
Optimal	3770		89552	16	89568	89552	60320	149872	4.29%	19.04%	196.32	447.84

TABLE I

COMPARATIVE TABLE OF THE PROPOSED METHOD VS THE OPTIMUM STATE OF THE ART APPROACH [15]

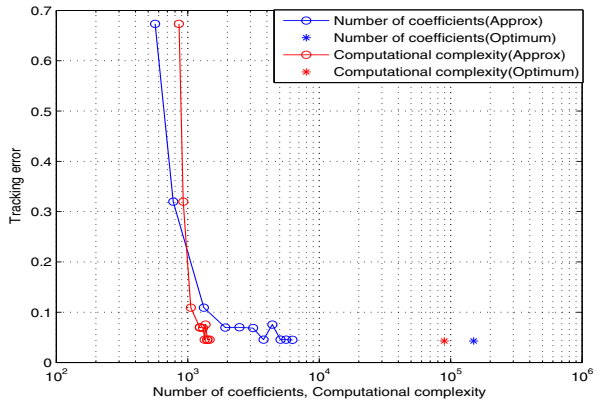

Fig. 4. Performance versus computational complexity trade-off comparison

In Table I approximation complexities are ranging from 100 to 600 vertices. The penultimate column reports the time in seconds that a MacBook Pro (2.8GHz, Core 2 Duo, 4GB ram) took to design the controllers. By comparing with the optimal controller, we get a reduction in the computation time ranging from $2.7\times$ to $4.9\times$. The last column reports the time needed to run-time execute the policy assuming a processor able to execute $200K$ FLOPs. A reduction versus the optimum approach ranging from $63.1\times$ to $104.4\times$ can be achieved.

As a performance metric, we use the normalized maximum undone work MU_w and the tracking error TCK_{err}. The first one represents the maximum amount of undone work ($U_t = w_t - f_t$) normalized to its total workload request, the second one the average value of it. If we consider finer controllers with more than 200 vertices, they provide a performance loss in terms of TCK_{err} that is only up to 2.7% lower than the optimal case. However, for these controllers the number of regions and vertices got greatly reduced. Results show a reduction up to $45\times$ in the number of coefficients and the computational complexity is reduced as well by a factor of $72.5\times$. The predictive behavior of the controller makes the undone work pretty uniform during the run-time execution of the policy. This is shown by the ratio between MU_w and TCK_{err} that ranges from 4.4 to 3.3. In Figure 4, under a certain number of vertices, the approximation error of the controller becomes significant and the tracking error increases exponentially. The first two approximations show a tracking error that is respectively 63% and 6.6% lower than the optimal case. However, for controllers with more than 200 vertices, we get points in the design space having almost optimal performance but with a much lower implementation overhead.

V. CONCLUSION

In this paper we propose an online smooth thermal control action, that minimizes the performance loss as well as the computational and hardware overhead. We formulate the problem as a discrete-time control problem using an approximate explicit model predictive control. Compared with the optimum solution [15], our results show a reduction up to $45\times$ in the number of coefficients and the computational complexity is reduced as well by a factor of $72.5\times$. The corresponding tracking error has an average loss up to 2.7%.

ACKNOWLEDGMENT

This research has been partially funded by the Nano-Tera.ch NTF Project CMOSAIC (ref. 123618), which is financed by the Swiss Confederation and scientifically evaluated by SNSF.

REFERENCES

[1] P. Kongetira, K. Aingaran, and K. Olukotun, *Niagara: A 32-way multi-threaded SPARC processor.*, IEEE Micro, March-April 2005.
[2] 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, March 2006.
[3] S. Borkar, *Design challenges of technology scaling*, IEEE Micro, 1999.
[4] C J. Hughes, et al. *Saving energy with architectural and frequency adaptations for multimedia applications*, Proc MICRO, 2001.
[5] G. Paci et al., *Exploring temperature-aware design in low-power MP-SoCs*, Proc. DATE, pp. 838-843, 2006.
[6] K.Skadron et al., *Temperature-aware microarchitecture: Modeling and implementation*, TACO, 2004.
[7] A.K.Coskun et al., *Temperature Aware Task Scheduling in MPSoCs*, Proc. of DATE, 2007.
[8] G.F.Franklin, et al., *Digital Control of Dynamic Systems*, McGraw Hill.
[9] A. Bemporad et al.,*The explicit linear quadratic regulator for constrained systems*, Automatica, 2002.
[10] S.Murali et al.,*Temperature Control of High Performance Multicore Platforms Using Convex Optimization*, Proc. DATE, 2008.
[11] J. Haase et al.,*Reliability-awarepower 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. J. Laub, et al., *Computation of System Balancing Transformations and Other Applications of Simultaneous Diagonalization Algorithms*, IEEE Trans. Automatic Control, AC-32, 1987.
[15] F.Zanini et al. *Multicore Thermal Management with Model Predictive Control*, ECCTD 2009.
[16] F.Zanini et al. *Optimal Multi-Processor SoC Thermal Simulation via Adaptive Differential Equation Solvers*, VLSISoC 2009.
[17] C.N.Jones et al. *The Double Description Method for the Approximation of Explicit MPC Control Laws*, CDC, 2008.
[18] J.Warren et al. *Barycentric coordinates for convex sets*, Journal of Advances in Computational Mathematics, 2007.