Exploiting Thermal Coupling Information in MPSoC Dynamic Thermal Management

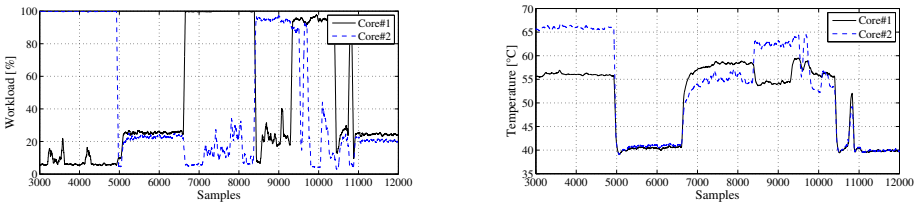
Simone Corbetta and William Fornaciari

Politecnico di Milano – Dipartimento di Elettronica e Informazione
Via Ponzio 34/5 – 20133 Milano, Italy
{scorbetta,fornacia}@elet.polimi.it

Abstract. Temperature profile optimization is one of the most relevant and challenging problems in modern multi-core architectures. Several Dynamic Thermal Management approaches have been proposed in literature, and run-time policies have been designed to direct the allocation of tasks according to temperature constraints. Thermal coupling is recognized to have a role of paramount importance in determining the thermal envelope of the processor, nevertheless several works in literature do not take directly into account this aspect while determining the status of the system at run-time. Without this information, the DTM design is not able to fully redistribute the roles that each core have on the system-level temperature, thus neglecting important information for temperature-constrained workload allocation.
Purpose of this work is to provide a novel *mechanism* to better support DTM policies, focusing on the estimation of the impact of thermal coupling in determining the appropriate status from a thermal stand-point. The presented approach is based on two stages: off-line characterization of the target architecture estimates thermal coupling coefficients, that will be used at run-time for proper DTM decisions.

1 Introduction

Microelectronics integration density is limited by the reliability of the circuits. Increasing power consumption of VLSI circuits causes thermal effects to become one of the most important concerns for circuit reliability. Experimental results show that more than 50% of integrated circuit failures are due to thermal issues [1]. In addition, it has been demonstrated that temperature spatial gradients have negative effect on the performance of the circuit [2]. Temperature profile optimization for reliable system design has become of paramount importance, and several Dynamic Thermal Management (DTM) approaches have been proposed in literature. The purpose of DTM is to control the varying temperature profile to optimize a given objective function: for instance, in reliable systems, the optimization goal is generally the maximum operating temperature or the maximum variability of the temperature (in time) compared to the average chip temperature. All these aspects require an accurate estimation of the status, from a thermal view-point, of the processor. With multi-core technology gaining further attention, this aspect is of paramount relevance since the problem of estimation is just a fraction of the DTM process, but the benefits of a good DTM are entirely based on the



(a) Workload (b) Temperature

Fig. 1. Thermal coupling effects on a dual-core architecture, showing (a) workload and (b) temperature of both cores.

estimation process itself. For this reason, it is belief of the authors that an accurate estimation methodology should be employed. The main important aspects of the proposed methodology are to be searched in the role that thermal coupling has on the temperature distribution in a chip [3].

1.1 Motivation example

This research work faces the problem of an appropriate dynamic thermal management support for multi-core processors, that focuses on the role of thermal coupling in the estimation process. The proposed work aims at the definition of a *mechanism* supporting DTM policies, while the definition of an appropriate *policy* is part of on-going research. The proposed novel methodology accounts for two important aspects: the coupling between pairs of cores, and the transient profile to reach steady-state temperature values. The importance of these two aspects will be discussed in this section, and a brief overview of the problem will be proposed from a formal stand-point.

In single-core processors the operating temperature is a function of the only core operating point: frequency, supply voltage and the load impacting on the switching activity of the core circuitry. In this context, temperature is mainly due to the self-heating process, since power consumption leads to thermal energy dissipation. Thermal management in this scenario is rather trivial, since control points converge into the processor itself. Nevertheless, this is not the case with multi-core architectures. Assume a double-core architecture with asymmetric load, i.e. each core is loaded with applications that are different from a power requirements view-point, and suppose to sample the temperature of each core once every τ time units. Then, consider the temperature profile as a function of the workload in each processor. Figure 1 reports the temperature profile and the workload of an Intel©Core™2 Duo processor; temperature values are sampled once every second using appropriate standard interfaces available in the GNU/Linux operating system. Two main aspects are clear from this scenario. The first one is on the effect of cores proximity on the temperature distribution inside the package: even though Core#1 is low loaded (under 10% on average) in the first period from sample 3000 to sample 5000, it actually *does* consume a non negligible portion of static power due to the temperature induced by the activity of Core#2. As a matter of fact, the operating temperature of Core#1 is 55°C in this time period, while its lowest value is around

40°C as it happens in the time frame between samples 5000 and 6500. This means that an inactive core should be actually subject to power/performance decisions, and also to reliability concerns, since it plays a relevant role in the system power dissipation and thermal envelope. Another important aspect is the one of transients. The rate at which a core heats up depends on several factors, both local (i.e., proper to the core) and non-local (i.e., induced by neighbor cores). The proximity to active cores makes the average cores temperature higher than in the case of isolated environments or when the system is entirely in low activity, also due to the package resistance and capacitance characteristics to spread thermal energy. Furthermore, the rate speed at which a core heats up is a function of the difference between the steady-state and the actual temperatures [4], the physical parameters determined by the architecture and fabrication of the chip, the workload of that core and of the neighbor ones, and power consumption.

1.2 Novel contributions

By exploiting the current temperature readings, the temperature history and appropriate thermal status information, purpose of our research work is giving a major source of information to the system-wide thermal management subsystem to control temperature envelope and to prolong device lifetime. The proposed approach tries to convey raw information and power/performance metrics to give a more accurate overview of the system status from a temperature view-point, considering the direct effects of thermal coupling and core-to-core proximity. To this extent, we define a methodology to estimate the status, from a thermal view point, of each core in a multi-core processor. We developed a two-stages methodology, where accurate thermal coupling information is collected once off-line, properly stressing the target architecture, and used at run-time to take appropriate power/thermal management optimizations. The novel contributions of this work can be summarized as follows: (i) a novel *mechanism* for supporting Dynamic Thermal Management is proposed, focusing on the role of thermal coupling in specifying the status of each core; (ii) a novel system-wide metric is proposed, through an appropriate formal model; (iii) a two-stage estimation methodology has been developed, general enough to be virtually employed in any multi-core architecture.

The remainder of the paper is organized as follows. The metric will be presented in a formal way in Section 2: the analytical expressions raising the metric will be presented, and the methodology to estimate the coupling coefficients in the target architecture will be also discussed. Experimental results are then given in Section 3. Related works on the topic are given in Section 4, while conclusions are drawn in Section 5.

2 Definition of a thermal status metric

The proposed methodology is based on the novel concept of *thermal status* (or *status*), combining power, performance and temperature metrics; the logic that glues together these aspects resides in the thermal coupling. It is meant to be employed on-line and to give a comprehensive status of the processor to support dynamic thermal management policy selection.

2.1 Neighbor-aware thermal status

As sketched in Section 1, in multi-core architectures the temperature reached by a core is a function of local and non-local contributions. In particular, the heat associated to a core j is a function of two different sources. A *generative* one accounts for the dynamic power consumption according to its activity; this is independent on the floorplan, but impacts cores in the proximity. Also, an *exchange* contribution takes into account coupling effects among cores that reside in the neighborhood, and it is proportional to the physical distance between cores. This last term models the spatial aspect of thermal heat diffusion. With these definitions, the generative contribution can be mapped to the power consumption profile of the generic core j, while the exchange contribution accounts for thermal proximity. We define the generative contribution $g_j(t)$ of core j at sample time t in Equation 1, through a function of the operating point tuple $\mathcal{O}_j(t) = \langle V_j(t), f_j(t), w_j(t), T_j(t) \rangle$, with $V_j(t)$ being the supply voltage applied to that core, $f_j(t)$ its operating frequency, $w_j(t)$ its average load and $T_j(t)$ its operating temperature. Notice that $V_j(t)$ and $f_j(t)$ can change dynamically according to the presence of a power management framework. To simplify the on-line estimation process, their values are defined relative to the maximum (architecture dependent) voltage V_{MAX} and frequency f_{MAX}. The additive term takes into account temperature-dependent leakage effects. The values ξ_{dyn} and ξ_{st} are used to weight the dynamic and static power contributions; their typical values are set according to reflect scaled technologies [5], e.g. $\langle \xi_{dyn}, \xi_{st} \rangle = \langle 0.65, 0.35 \rangle$.

$$g_j(t) = \xi_{dyn} \cdot \left(\left(\frac{V_j(t)}{V_{MAX}} \right)^2 \frac{f_j(t)}{f_{MAX}} w_j(t) \right) + \xi_{st} \cdot exp\left\{ -1/T_j(t) \right\} \tag{1}$$

Equation 1 provides a perspective with respect to local information on the cores, i.e. its validity is confined to the generic core j. In order to account for the heat conduction between adjacent cores, we apply the generative contribution definition to each core k in the neighborhood set N_j of core j, as given in Equation 2. The second contribution from $s_j(t)$ is due to neighbors activity. θ_{jk} represents the thermal coupling factor that holds between cores j and k. The generative value is then further weighted through an index $\alpha_j(t)$ accounting for aging aspects. The value of α changes over time as a consequence of temperature-accelerated reliability concerns, for instance considering the impact of Negative Bias Temperature Instability (NBTI) on the performance of the device [6]. In this work, we assume it constant ($\alpha_j = 1$) for simplicity.

$$s_j(t) = \alpha_j(t) \cdot g_j(t) + \sum_{k \in N_j} \theta_{jk} \cdot \alpha_k(t) \cdot g_k(t) \tag{2}$$

$$\Lambda = \frac{s_1 T_1 + s_2 T_2 + \cdots + s_N T_N}{T_1 + T_2 + \cdots + T_N} = \frac{1}{N T_p} \sum_{j=1}^{N} s_j T_j \tag{3}$$

The definition of thermal status given in Equation 2 provides local as well as coupling information, but focuses on a per-core perspective and does not provide a system-wide perspective. A system-wide thermal status metric Λ is defined in Equation 3, where N

is the number of cores [1]. The system-wide perspective is a weighted sum of the thermal status of each core in the processor and weights are determined by the local temperature values T_j; in this way, it jointly exploits local and package (average) temperature as well as thermal status information. The expression at the denominator equals N times the average temperature T_p in the processor.

2.2 Proactive thermal status

The model given in Equation 3 allows us to capture the heat-up rate of the system, in terms of perturbations of the (local) thermal status of each core. In general, considering the thermal coupling effects exploited in the model of Equation 2, a perturbation of the local thermal status s_j will have direct effects on the thermal status s_k in the neighborhood set N_j, by means of thermal coupling phenomenon. From a system-wide perspective, on the other hand, a perturbation of the (local) thermal status of each core will have additive effects on the value of Λ. According to Equation 3, these effects will be a function of the local temperature values with respect to the system's average temperature. This aspect ranks the cores according to their role in determining the system temperature: the higher the core-to-package ratio T_j/T_p, the higher the role of core j in determining temperature T_p; this can be due for instance to the different power/performance profile of core j, or to coupling contributions. To formalize the effects on the system-wide metric Λ by means of local perturbations, we have to consider the total derivative of Λ with respect to the perturbation s_j in the vector $\underline{S} = [s_1 s_2 ... s_N]$. Equation 4 reports the analytical expression of the derivative. By definition, we have to take care of three quantities: the partial effects on Λ of an increase in s_j; the partial effects on Λ due to an increase in s_k, for each k in the neighborhood set N_j, and the mutual effects of s_j and s_k.

$$\frac{\mathrm{d}\Lambda}{\mathrm{d}s_j} = \frac{\partial \Lambda}{\partial s_j} + \sum_{k \neq j} \frac{\partial \Lambda}{\partial s_k} \frac{\mathrm{d}s_k}{\mathrm{d}s_j} = \frac{T_j}{N\,T_p} + \sum_{k \neq j} \frac{T_k}{N\,T_p} \theta_{jk} \tag{4}$$

The first term $\partial\Lambda/\partial s_j$ defines the effect on Λ of an increase in the value of thermal status s_j; according to Equation 3 this derivative equals $T_j/(N\,T_p)$: an increase in the local status has an impact on the system-wide perspective that is a function of the relative temperature with respect to the system average temperature. The summation term extends the derivative to the adjacent cores; the first factor of the summation has the same meaning as the previous one but focuses on adjacent cores. As a second factor, the effects of considering the ratio of the differentials $\mathrm{d}s_k$ and $\mathrm{d}s_j$ from two cores k and j, models the mutual influence that a change in thermal status s_j has on the adjacent ones; the ratio $\mathrm{d}s_k/\mathrm{d}s_j$ can thus be understood to be the thermal coupling factor θ_{jk} from Equation 2. The values assumed by these coefficients are estimated through an appropriate methodology, discussed later on in Section 2.3. Hence, the model in Equation 4 is a suitable tool to keep track of the dynamic behavior of the system in response to an evolution of the thermal status of each core, and to assess the mutual influence of thermal status between pairs of cores as the system evolves. The only relationship

[1] From now on, the notation will be simplified avoiding the use of time variable t.

modeled in Equation 4 is the one between status s_j and Λ and nothing is said about the relationship between a change in g_j and Λ, necessary to estimate the effects of local power/performance requirements on system-wide metrics. Similar to what has been done in Equation 4, we can compute the effect of a perturbation of the self-heating contribution on Λ. In this case, however, additional care must be taken: s_j is a function of g_j, such that Λ is a compound function of s_j and g_j and $\Lambda = f(s_j \circ g_j) = f(s_j(g_j))$; as a consequence the impact on Λ of g_j is not directly exploited, but should pass through the analysis of the impact on s_j; last, the mutual influence between $\mathrm{d}g_j$ and $\mathrm{d}g_k$ is to be considered null, by definition of the self-heating contribution in Equation 2. The total derivative is reported in Equation 5.

$$\frac{\mathrm{d}\Lambda}{\mathrm{d}g_j} = \frac{\mathrm{d}\Lambda}{\mathrm{d}s_j} \cdot \frac{\mathrm{d}s_j}{\mathrm{d}g_j} = \left(\frac{\partial \Lambda}{\partial s_j} + \sum_{k \neq j} \frac{\partial \Lambda}{\partial s_k} \cdot \frac{\mathrm{d}s_k}{\mathrm{d}s_j} \right) \cdot \left(\frac{\partial s_j}{\partial g_j} + \sum_{k \neq j} \frac{\partial s_j}{\partial g_k} \cdot \frac{\mathrm{d}g_k}{\mathrm{d}g_j} \right) \tag{5}$$

Locally, we can compute the impact on the thermal status by an increase in the generative contribution in three generic cases: (a) when an increase in the core activity does not modify the operating point; (b) is in the opposite direction, in which the only operating point is changed; (c) refers to the mixed change in the activity and operating point. The common contribution in cases (a), (b) and (c) relies on the sole ξ_{dyn} factor, determining the impact of dynamic power contribution as a function of the power/performance point of core j. Last, the ratio between the two differentials $\mathrm{d}g_k$ and $\mathrm{d}g_j$ is null by definition.

2.3 Coupling coefficients estimation

The proposed estimation methodology accounts for the contribution given by the different subsystems composing the reference SoC architecture (e.g., the coupling contribution due to hardware blocks different from the cores). Also, it can be employed on different architectures, thus developing as a general methodology. The interest on this last aspect increases with the increasing availability of multi-core architectures in different domains, pushing researchers to find a suitable methodology to easily adapt their thermal-aware designs and thermal management solutions in a broad range of platforms and operating environments. The proposed approach accounts for the availability of on-chip resources and on-board cooling facilities (e.g., fans or fan less, heat sinks dimension and geometries...), through *direct* temperature observation and platform characterization at warm-up periods. The proposed methodology is divided into three steps. The generic mapping specification M is used to load the multi-core architecture to cover a broad range of allocation patterns: each pattern specifies which core will be assigned work to, and which load. For each mapping M, temperature values are sampled from each core until the steady-state temperature of the system is reached. Once the temperature vector T is known, coupling factor θ is estimated for that particular configuration, and for each core pairs. The flow ensures that different load levels are allocated to each core in the system; this is driven by the rationale that the effective thermal coupling coefficient is a function of the status of each core, in terms of power consumption, and it is not dependent on the sole floorplan. To estimate the coupling coefficients, we run a predefined set of typical applications with the objective to

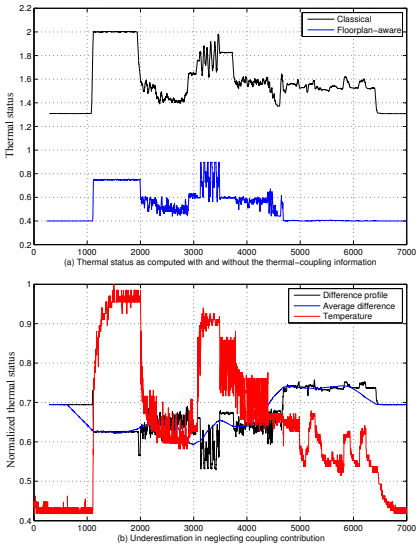
heat-up the processors in a controlled environment. Denoting with X the set of active processors, and with Y the available CPU activity, the mapping should cover a proper subset of $X \times Y$. Temperature profiles are used for computing the thermal coupling coefficients θ_{jk} between cores j and k as $\theta_{jk} = \left(T_j - T_k^{(q)}\right) / T_k^{(q)}$, where T_j is the maximum temperature reached by core j in a particular configuration $\langle x, y \rangle \in X \times Y$, while $T_k^{(q)}$ is the quiet-state temperature of core k, i.e. its temperature when the system is idle (for simplicity, this equals ambient temperature). The output of the second stage is a multi-dimensional matrix summarizing the thermal coupling phenomena at different configurations. Each entry defines the coupling factor as a function of the workload, and can be hard-coded in the DTM policy.

3 Results

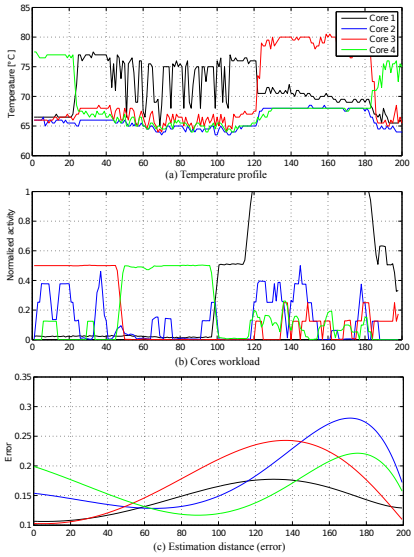
We conducted several experiments on representative Intel©i7 820QM quad-core processor, featuring 4 physical on-chip cores supporting up to 8 threads, operating at a maximum frequency of $1.73\mathrm{GHz}$ and fabrication node $45\mathrm{nm}$; the processor is design to support a Thermal Design Power of $45\mathrm{W}$. We used different real-life applications, ranging from scientific to multimedia workloads. Since the aspect of interest resides in the coupling phenomenon without any specific relationship with the application running on each core, we will not go through any workload characterization: from our perspective the heating process is the most important feature, independently on the application of interest. Data related to temperature, frequency and workload is taken during execution of mixed applications. Results here are grouped into different sections. Section 3.1 shows the difference of estimation in two cases: with and without the support of the thermal coupling contribution from Equation 2. Section 3.2 shows the results obtained from the estimation of thermal coupling with the methodology proposed in Section 2.3, considering quad-core processors. Finally, Section 3.3 gives an overview of the estimation process overhead as a function of the history window depth.

3.1 The impact of thermal coupling

There is a slightly different estimate in the proposed model with respect to neglecting coupling contributions. Such difference can be as high as as 30% in some cases. This difference is shown in Figure 2(a), highlighting the influence of operating temperature on the estimation. The top-most figure reports the thermal status estimate with and without considering the thermal-coupling contributions (respectively, Floorplan-aware and Classical scenarios); such difference demonstrates that without considering coupling contribution, there will be an underestimation, and this underestimation being as much as 65%. This will generally influence DTM decisions, since policies are biased to the information provided about the system from a thermal view-point. The bottom-part of the figure, on the other hand, shows the relative error (with its moving-average) superimposed to the normalized temperature profile. In this case, the error is higher when average temperature is lower; this empirical observation of our model is in concordance with reality, since at higher temperatures the impact of coupling decreases due to the



(a) Estimation mismatch



(b) Estimation mismatch breakdown for each core

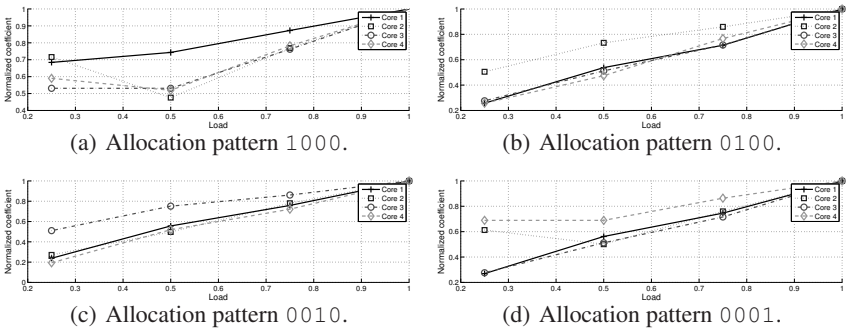
Fig. 2. Estimation mismatch and error profile in a quad-core processor, as a function of the operating temperature and workload.

fact that the neighborhood average temperature tend to saturate. For completeness, Figure 2(b) reports the absolute difference for each core considering a specified time span of 200 samples, as a function of the activity and temperature of each core. The difference in a quad-core processor can reach up to 65%, while experimental results (not reported for lack in space) on dual-core processors show the difference is below 50%. Similar conclusions as for the previous case can thus be drawn.

3.2 Thermal coupling coefficients

The results in this section show how to determine quantitative values for the $\theta_{j,k}$ coefficients from Equation 2, varying with the dynamic power and performance state of the processor. To simplify the characterization process, we consider the temperature ranges of each core as a function of the workload associated to each core, rather than as a function of the effective power consumption. The advantage of this approach is in the simplified estimation process, although its validity is bound to a particular power management policy. In this perspective, we conducted several experiments with different allocation patterns: we then discretized the available workload in steps of 25%, using tools available in GNU/Linux operating system. In order to find the values in the multi-dimensional matrix, an exhaustive set of tests should be performed covering all possible allocation patterns. Due to lack in space, only a selected subset of representative tests will be reported. Notice that we are not interested in selecting a specific benchmark

to stress the processor, because we are interested in heating-up the processors without being aware of the executing application, since the focus is on application-independent thermal management. Figure 3 shows how the coupling coefficient varies effectively with the load; the values of the coefficients are reported normalized with respect to the maximum. Figure 3 highlights a varying trend, relying upon the core's activity. The fact that all the cores' coefficients asymptotically reach the same trend line while load tends to 100% reveals the steady-state nature of thermal coupling phenomenon, as already mentioned earlier: at steady-state temperature the system is stable, influence decreases and the coefficients reach their maximum values. In this case, the system will remain stable until it changes its state.



(a) Allocation pattern `1000`. (b) Allocation pattern `0100`.

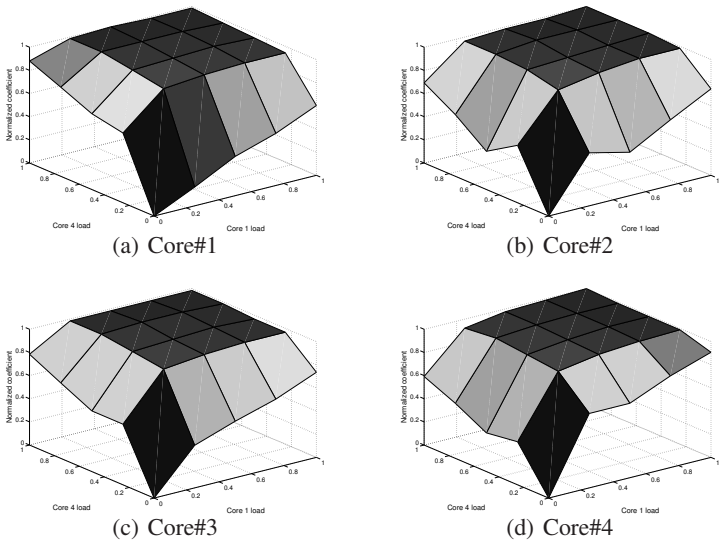
(c) Allocation pattern `0010`. (d) Allocation pattern `0001`.

Fig. 3. Thermal coupling coefficients trend, as a function of the load of the active core when a single core is active, for different allocation patterns.

The same experiments have been conducted while loading more than one core in the processor; the procedure has been exhaustively repeated for every allocation pattern. Figure 4 reports the coefficients for one single allocation pattern, for lack in space. The surface shows the influence of cores activity on each core in the architecture (active or idle). The values plotted on the surface are normalized to their maximum values. As already sketched in the previous results, the coefficients have much higher variance at lower temperatures (represented by lower cores' activity).

We can also capture additional relevant aspects in the thermal coupling phenomenon. The thermal coupling coefficient is monotonically non-decreasing for a given mixed workload. The first difference we encounter in the quad-core case is a steady-state flat region early toward higher values of the cores activity; this aspect was not experienced in the dual-core case, and it might be associated to the allocation pattern and to the relevant differences in the architecture design and floorplan (however, at the same technology node). Furthermore, the coupling trend is in general a function of the allocation pattern. Even though this dependence is weak at higher cores' activity, it is quite relevant when the cores are loaded weakly: this is clear if we compare the trend for Core#2 in the two allocation patterns.

Notice that the aforementioned aspects are in line with the concept of thermal impedance in chip packages. Indeed, for a given package, the thermal impedance is



(a) Core#1 (b) Core#2

(c) Core#3 (d) Core#4

Fig. 4. Thermal coupling trend of each core, as a function of the CPU activity of the active cores. Cores are reported in order (Core#1 on the extreme left and Core#4 on the extreme right). Data are reported for allocation patterns `1001`.

reported to be a function of both the duty cycle and the duration of the (equivalent) power consumption pulse. To the best of our knowledge, this is the first time a concept like this is employed in multi-core architectures to characterize the thermal behavior of the SoC importance of thermal coupling in future technology nodes and multi-core architectures, we can envision that in perspective a coupling pre-characterization of commercial multi-core processors will be integrated in the processor data sheet.

3.3 Estimation overhead

The estimation process reported in Section 2 and Section 2.3 has been completely developed under a GNU/Linux operating system, based on the `lm-sensors` interface to retrieve low-level information from the hardware architecture (e.g., temperature samples, workload activity, frequency and supply voltage values). The framework has been developed entirely as user-space, using `gcc` compiler version 4.4.3. Slow-down estimates as a function of the history depth have been proven to be in the order of a few milliseconds, making it suitable for real deployment. Furthermore, the performance requirements have been shown to be independent on the history depth: the bottleneck is given by the communication overhead in retrieving values from the `lm-sensors` API, since the framework has been entirely developed in user-space.

4 Related works

Dynamic Thermal Management refers to a set of techniques to optimize the temperature profile of microprocessor systems, generally under performance constraints [7]. DTM is defined by means of *policies* taking decisions according to a system-level or local view of the processor, and *mechanisms* supplying low-level information on the status of the system from a thermal view-point. Runtime profile optimization can usually be performed in either of two ways [8]: through thermal sensors readings, or solving formal models. Thermal sensors provide direct on-chip temperature samples, but these values are highly dependent on their placement; algorithmic approaches, on the other hand, are based on solving formal models at runtime, generally accurate at the cost of high computational overhead. Load balancing techniques for MPSoC architectures can benefit from multi-threaded support from the hardware [9] as well as multi-processor support [10]. Migration policies are investigated while achieving processor throughput, however without considering either temperature history nor the effects of thermal coupling on inactive cores (as sketched in Section 1). If application profile is available, thermal management can benefit from appropriate ordering approaches, such as the one presented in [11]. However, application profiling suffers from dependence of mapping policies. Off-line profiling is used also in other predictive approaches [12]. Temperature history and workload are weighted in [13] to predict future operating temperature of the processor. However, the authors do not consider the effects of cores proximity in determining the thermal status of active and inactive cores.

5 Conclusions

This paper presented a novel methodology to *support* Dynamic Thermal Management in complex MPSoC architectures, in which the thermal requirements are limiting the integration capability as well as the efficiency of optimization policies. Since thermal coupling phenomena are a major source of challenge, and proactive approaches are preferred in reliability-aware designs, the proposed methodology focuses on a coupling-driven estimation of the effects of self-heating contributions on system-wide thermal status metrics. The proposed methodology is able to avoid underestimations in predicting the thermal status. Indeed, even a typical difference of $20 \div 30\%$ in the local status estimation process has a negative impact on the system-wide perspective. Our approach is able to provide a better estimation support, employing architecture-dependent information on the dynamic behavior of heat exchange among the different subsystems in the processor. Such support is of paramount importance in DTM to dynamically reveal the evolution of the system from a temperature stand-point. The proposed approach presents a *mechanism*, rather than a policy, to support DTM decisions. The overhead is kept low thanks to a double-phase approach: off-line pre-characterization of the target architecture (e.g., at bootstrap time) collects sensible information that will be conveniently used at run-time for proper DTM decisions. Experimental results have been collected on real commercial processors, and temperature values reflect the real usage of the system, with typical workloads from different scenarios.

Acknowledgments

This research work is partially supported by the EU-funded 2PARMA FP7 research project (http://www.2parma.eu/) focusing on resources management techniques and methodologies in multi-core and many-core architectures.

References

1. Lasance, C.J.M.: Thermally driven reliability issues in microelectronic systems: status-quo and challenges. Microelectronics Reliability **43**(12) (2003) 1969–1974
2. Ajami, A., Banerjee, K., Pedram, M.: Modeling and analysis of nonuniform substrate temperature effects on global ULSI interconnects. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **24**(6) (June 2005) 849–861
3. Janicki, M., Collet, J.H., Louri, A., Napieralski, A.: Hot spots and core-to-core thermal coupling in future multi-core architectures. In: 2010 26th Annual IEEE Semiconductor Thermal Measurement and Management Symposium (SEMI-THERM), IEEE (February 2010) 205–210
4. Incropera, F.P., DeWitt, D.P., Bergman, T.L., Lavine, A.S.: Fundamentals of heat and mass transfer. Wiley (2007)
5. International Technology Roadmap for Semiconductor, Design chapter 2010, available from http://www.itrs.net/
6. Alam, M., Kang, K., Paul, B., Roy, K.: Reliability- and process-variation aware design of vlsi circuits. In: Physical and Failure Analysis of Integrated Circuits, 2007. IPFA 2007. 14th International Symposium on the. (july 2007) 17 –25
7. Brooks, D., Martonosi, M.: Dynamic Thermal Management for High-Performance Microprocessors. In: High Performance Computer Architecture, 2001. HPCA '01. 17th International Symposium on. (2001)
8. Siozios, K., Rodopoulos, D., Soudris, D.: Quick_hotspot: A software supported methodology for supporting run-time thermal analysis at mpsoc designs. In: Architecture of Computing Systems, 2011. ARCS '11. 23rd International Conference on. (2011)
9. Gomaa, M., Powell, M., Vijaykumar, T.: Heat-and-run: leveraging smt and cmp to manage power density through the operating system. In: Proceedings of the 11th international conference on Architectural support for programming languages and operating systems. ASPLOS-XI, New York, NY, USA, ACM (2004) 260–270
10. Donald, J., Martonosi, M.: Techniques for Multicore Thermal Management: Classification and New Exploration. In: Computer Architecture, 2006. ISCA '06. 33rd International Symposium on. (2006)
11. Yang, J., Zhou, X., Chrobak, M., Zhang, Y., Jin, L.: Dynamic thermal management through task scheduling. In: Performance Analysis of Systems and software, 2008. ISPASS 2008. IEEE International Symposium on. (april 2008) 191 –201
12. Srinivasan, J., Adve, S.V.: Predictive dynamic thermal management for multimedia applications. In: ICS '03: Proceedings of the 17th annual international conference on Supercomputing. (2003)
13. Yeo, I., Liu, C.C., Kim, E.J.: Predictive dynamic thermal management for multicore systems. In: DAC '08: Proceedings of the 45th annual Design Automation Conference, New York, NY, USA, ACM (2008) 734–739