Thermal model identification of computing nodes in High Performance Computing systems

Roberto Diversi, Andrea Bartolini, *Member, IEEE*, and Luca Benini, *Fellow, IEEE*

***Abstract*—Thermal-aware design and on-line optimization of the cooling effort are becoming increasingly important in current and future high performance computing (HPC) systems. A fundamental requirement to effectively develop such techniques is the availability of distributed and compact models representing the system thermal behavior. System identification algorithms allow to extract models directly from the thermal response of the target device. This paper proposes a novel thermal identification approach for real, in-production HPC systems, which is capable of extracting thermal models from a computing node affected by quantization noise on the temperature measurements as well as operating in free-cooling mode, with variable ambient temperature. The approach allows also to identify the physical floorplan of the CPU dies in supercomputing nodes. The effectiveness of the proposed methodology has been tested on a node of the CINECA Galileo Tier-1 supercomputer system.**

***Index Terms*—Thermal modelling, system identification, supercomputing nodes, HPC systems.**

I. Introduction

AFTER the end of Dennard's scaling, one of the prominent limiting factor of the performance of today's and future computing systems is the so called "Thermal and Power Wall" [1]. Indeed, the power density of computing devices ha s increased across generations: higher power density increases silicon temperature, which in turn increases cooling costs and/or jeopardizes performance.

High Performance Computing (HPC) systems, warehouse-scale computing, and datacenters are gaining importance in today's society and industry [2]. Recent reports quantify the Return on Investment (ROI) produced by applying HPC in an industrial environment: in Europe each euro invested in HPC generates in average 867€ of increased revenues and 69€ in profit, while in the US a single dollar invested in HPC generates in average 43$ of profit [2].

Datacenters and supercomputing centers are large and complex industrial plants by their own [3]. These systems are composed of several computing rooms each filled with several racks containing tens/hundreds of computing nodes. Each computing node is composed of few computing elements (CPUs/GPUs) based on multi/many-core processors. The power consumption of these installations ranges from few to tens of MWatts. To remove the heat generated by the active electronics, additional power is required. A recent study of Gao shows that in average today's datacenters of Google pay an additional 12% of power consumption for power delivery and cooling dissipation [4].

Traditional cooling methods, based on computer room air conditioners (CRAC), or computer room air handlers (CRAH) have been enhanced with free-cooling mode, i.e., the capability to exploit the outside air, using only the AC blowers to circulate it in the room [5]. Moreover, cooling energy can be significantly reduced if hot water cooling is used to remove the heat [6]. In both these cases a hotter-than-nominal coolant is used to remove the heat, leading often to a higher silicon temperature in the computing unit [3], [7].

Today's processors use two mechanisms to protect the silicon die from over-temperature (referred as thermal management): (i) dynamic voltage and frequency scaling (DVFS) as well as (ii) duty-cycling (Thermal Throttling). While DVFS is used as primarily control mechanism for power management in firmware and operating systems, duty-cycling is used as a fail-safe HW protection mechanism when die temperature exceeds a critical threshold. Indeed, with DVFS the performance loss increases sub-linearly with the power reduction, while with thermal throttling the performance loss increases linearly with the power reduction and thus the latter has worse impact on the core performance.

One could think that today's best-in class computing elements are thermally stable for the entire commercial operating temperature range, however Moskovsky et al. [7] show that built-in mechanisms can use effectively the DVFS only for power management and fails in preventing thermal throttling [7]. This is primarily due to the reactive nature of the DVFS-based thermal controllers which take corrective actions when the chip is already too hot.

Optimal control and thermal-aware resource management strategies can ease this problem provided that a predictive thermal model and a thermal interaction map can be extracted from the deployed silicon die [3], [8]. In digital electronic devices temperature depends on power dissipated, which depends on the utilization, instruction composition and operating point (voltage supply and clock frequency). These parameters can be monitored directly form each deployed computing element by means of integrated power, temperature and performance sensors. System Identification algorithms can

Manuscript received December 2, 2018; revised May 1, 2019, and July 23, 2019; accepted September 11, 2019. This work was supported by the EU ERC Project MULTITHERMAN (g.a. 291125) and the EU FETHPC project ANTAREX (g.a. 671623). The authors would like also to thank Francesco Beneventi from the University of Bologna for his support on the framework and on the data collection and CINECA.

The authors are with the Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna, Bologna, Italy (e-mail: roberto.diversi@unibo.it; a.bartolini@unibo.it; luca.benini@unibo.it) Luca Benini is also with the Integrated Systems Laboratory, ETH Zurich, Switzerland (e-mail: lbenini@iis.ee.ethz.ch)

be used to extract a thermal model to predict the future thermal evolution directly from these sensor's traces.

Previous works [9], [10] have shown that classical AutoRegressive eXogenous (ARX) models do not lead to a physically meaningful model, which is expected to be passive (i.e with only stable poles) and with real and positive poles. Indeed, ARX models are widely used in system identification, as they constitute the simplest way of representing a dynamic process in the presence of uncertainties [11]–[15]. ARX models are suitable to represent the so-called "process noise", whose aim is to represent unavoidable model approximations, but are not able to represent input and output measurement noise [11], [12]. However, the thermal sensors embedded in the silicon die of server's class processors are affected by thermal [9], [10] and quantization [16] noise. To account for the presence of this noise, the relation between the core temperature and the dissipated power has been described by means of Multi-Input Single-Output (MISO) ARX models with additive noise on the output [9], [10], [16].

A. Related Works

Several works in the state on the art focused on extracting thermal models from real computing systems. We cluster these approaches in (i) Output Error, (ii) AutoRegressive, and (iii) AutoRegressive plus noise methods.

(i) The Output Error approaches are based on the solution of an optimization problem to extract the model parameters from input-output data without introducing a specific disturbance model. In the domain of multicore thermal modelling, Beneventi et al. [17] present an Output Error system identification strategy that is robust to quantization noise on the input temperature measurements. This is achieved by adding to the basic optimization problem a set of linear constraints that filters out the model parameters that are not physically valid. The approach is validated on a quad-core server platform and shows that a 2nd order model is required. This method can capture error in the output variable but cannot handle "process noise".

(ii) In contrast, AutoRegressive approaches do account for process noise but not for the input and output measurement noise. The simplest methods consist in extracting first-order dynamic thermal models by solving a linear least squares (LS) optimization problem [18], [19]. Coskun et al. [20] propose an AutoRegressive Moving Average (ARMA) technique for predicting the future thermal evolution of each core. The derived model predicts future temperature by using only its previous values. Since it does not account directly for workload-to-power dependency, a Sequential Probability Ratio Test (SPRT) technique is used to rapidly detect changes in the statistical residual distribution (average, variance) and, then, to re-train the model, when it is no longer accurate. Juan et al. [21] use a combination of a K-means clustering and an Auto-Regressive (AR) model to learn a compact model for fast thermal simulation. This approach is effective only when starting from a highly accurate thermal model of the HW. Moreover, the missing exogenous terms in both of the above approaches leads to neglecting the direct link between dissipated power and temperature. Bartolini et al. [22] present a distributed model learning approach based on a set of classical ARX models. Each core executes its own model learning routine generating a local thermal model. The model is used internally, in each core, by a local model-predictive controller. However this approach has been applied only to simulated systems and it is based on the assumption that per-core power traces and thermal sensor outputs are accurate and without noise. Reda et al. [23] describe a method for identifying thermal models and power consumption starting from the measurements of thermal sensors and total power consumption. This approach is based on a multivariable ARX model.

(iii) In [9] Diversi et al. introduced a bias compensated least squares (BCLS) approach for identifying MISO ARX with noisy output (MISO ARX + noise) models, which has been extended in [10] with a distributed implementation. MISO ARX + noise models allow to take into account the presence of both process noise and measurement noise. These works have been conducted on the Intel Single Chip Cloud computer test device which featured "cheap" ring oscillators as thermal sensors. Nevertheless, as shown in [16], the performance of the BCLS identification algorithm is not satisfactory when applied to the thermal modelling of supercomputing nodes within in-production HPC systems, affected by quantization noise on the temperature measurements as well as operating in free-cooling mode, with variable ambient temperature. In [16], the identification of noisy ARX models has been performed by combining an ad hoc Frisch scheme and the instrumental variable method. This approach has proved to be more effective than previous ones for the considered real-life production HPC system.

B. Contribution

This paper extends the work in [16] by: (a) proposing an identification algorithm based on a bilinear system of equations whose unknowns are the model parameters and the additive noise variance. The proposed approach is computationally simpler than [16] since it is based on the iterative application of two least squares formulas. On the contrary, the method described in [16] is based on the constrained minimization of a loss function requiring a numerical search methods. Another advantage of the new algorithm concerns the possibility of obtaining a recursive version, which is of interest for online thermal management where the model has to be updated in order to track parameter changes. The proposed approach can effectively learn thermal models which are physically valid and capable of predicting the core's temperature with errors within the quantization noise. (b) we propose a formulation to extract directly from the identified core models the physical floorplan of the supercomputers' CPU dies. This is of primarily interest for optimizing the thermal map of the processors and mitigating hot spots. The spatial information that can be extracted from the CPU die floorplan can also ease the complexity of the cores' thermal model for online model predictive control and optimization algorithms.

The reminder of the paper is organized as follows. Section II describes the adopted identification methodology, highlighting the experimental framework and the adopted workloads.

Section III focuses on the thermal model of the node. Section IV described the identification of the core models by means of a two-step iterative least squares algorithm. Section V shows how to identify the CPU die floorplan starting from the core models. In Section VI, the experimental results obtained by applying the proposed approach to a node of the CINECA Galileo Tier-1 HPC system are reported and discussed. Section VII concludes the paper.

II. Thermal model identification methodology

A HPC cluster is a composition of several computing nodes. Each computing node is composed of several computing engine. We consider as computing engine high-performance multicore CPUs, which are the most common case (more than 77% of Top500 systems use Intel Xeon Processors, as reported by the June 2018 Top500 list). Each parallel processor (CPU/socket) is internally composed of N_C cores and uncore logic. The uncore logic accounts mainly for the memory controllers, last level cache and I/O. Processor, cores and uncore are equipped with sensors which can be read periodically from the software stack. These sensors can monitor different architectural events as well as physical parameters. We consider three main classes of sensors: per-core activity sensors, per-core thermal sensors, and per-CPU power gauges.

The proposed methodology to extract the thermal model consists of three steps: (i) Binary workload ($1 = active/0 = idle$) sequences are applied to each core to recreate a Pseudo Random Binary Sequence (PRBS) of power stress in each core. We schedule one power stressmark on each core and use POSIX signals to schedule and de-schedule it to follow the PRBS sequence. The power consumption values for each core are obtained by linear regression on the cores activities (cycles in active state and cycles in idle state) and per CPU power consumption. (ii) During this stress pattern, we monitor the sensors present on each core and CPU synchronously with the PRBS sampling time. The sampling time is fixed to two seconds to avoid interference of the monitoring and stress pattern injection. For lower sampling time the overhead of the monitoring becomes noticeable. It is worth noting that this choice does not prevent to estimate the fast time constants, see Table I. Due to the inherent difference of the speed at which PRBS power traces are applied and monitored, and at which power vary in silicon the binary power variations appear instantaneous. The values collected are pre-processed to translate each core activity in a per-core power profile. Finally, (iii) the proposed system identification algorithm is applied to the power traces and thermal responses to extract the thermal model.

III. Thermal Modelling of the Node

By following a distributed approach, the thermal dynamic profile of each CPU socket of each supercomputer's node with N_c cores is represented by a set of MISO ARX models with additive output noise, one for each core. The model of the generic k–th core is represented in Fig. 1, where $\bar{T}_k(t)$ is the actual core temperature and $T_k(t)$ is the measured core temperature corrupted by the additive measurement noise $v_k(t)$. $\bar{T}_k(t)$ is considered as affected by all the core powers $P_0(t), P_1(t), \ldots, P_{N_c-1}(t)$ and by the uncore power $P_{unc}(t)$ so that the input $P(t)$ is the following $N_c + 1$-dimensional vector

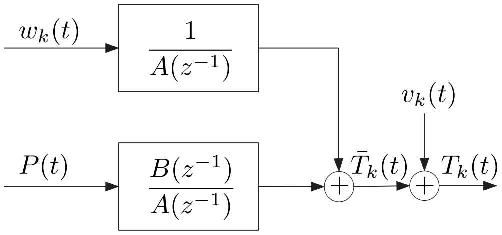


Fig. 1. MISO ARX + noise model of a single core.

$$P(t) = \begin{bmatrix} P_0(t) & P_1(t) & \cdots & P_{N_c-1}(t) & P_{N_c}(t) \end{bmatrix}^T, \quad (1)$$

where, for the sake of notation, the uncore power $P_{unc}(t)$ is denoted as $P_{N_c}(t)$. The actual core temperature $\bar{T}_k(t)$ is linked to the input $P(t)$ through the difference equation

$$A(z^{-1})\,\bar{T}_k(t) = B(z^{-1})\,P(t) + w_k(t) \quad (2)$$

where $A(z^{-1})$ is the polynomial

$$A(z^{-1}) = 1 + a_1\,z^{-1} + \cdots + a_n\,z^{-n}, \quad (3)$$

$B(z^{-1})$ is the $1 \times N_c$ polynomial matrix

$$\begin{aligned} B(z^{-1}) &= \begin{bmatrix} B_0(z^{-1}) & B_1(z^{-1}) & \cdots & B_{N_c}(z^{-1}) \end{bmatrix} \\ B_i(z^{-1}) &= b_{i1}\,z^{-1} + \cdots + b_{in}\,z^{-n} \quad i = 0, \ldots, N_c \end{aligned} \quad (4)$$

n is the system order and z^{-1} is the backward shift operator, i.e $z^{-1}\,x(t) = x(t-1)$. $w_k(t)$ is an input noise that generates the process disturbance $1/A(z^{-1})\,w_k(t)$.

The ARX model (2) is often used in system identification because it can be consistently estimated by means of the least squares method [11]. However, this model is not suitable for describing the thermal dynamics of the core [16]. In fact, all the identified ARX models present some negative real poles and/or complex conjugate poles so that they do not comply with the dynamic of thermal systems where only positive real poles have a physical meaning. The insufficient modeling power of ARX models may be explained by the presence of a relevant quantization noise on the temperature readings. For this reason, it is important to consider also the presence of an additive measurement noise $v_k(t)$ corrupting the (unknown) actual temperature $\bar{T}_k(t)$. Therefore, the available (measured) temperature $T_k(t)$ of core k is given by

$$T_k(t) = \bar{T}_k(t) + v_k(t). \quad (5)$$

The thermal dynamic of the core is thus described by a MISO ARX + noise model described by equations (2) and (5). This model allows to take into account the presence of both a process disturbance $1/A(z^{-1})\,w_k(t)$ and a measurement noise $v_k(t)$, see Fig. 1. The following assumptions will be considered.

A1. The input signals $P_0(t), \ldots, P_{N_c}(t)$ are persistently exciting of a suitable high order.

A2. $w_k(t)$ is a zero-mean white process with variance σ^2_{wk}.

A3. The measurement noise $v_k(t)$ is a zero-mean white process with variance σ^2_{vk}.

A4. $w_k(t)$ and the measurement noise $v_k(t)$ are mutually uncorrelated.

A5. The input signals are uncorrelated with both $w_k(t)$ and the measurement noise $v_k(t)$.

Remark 1: Assumption A1 is required to get a consistent estimation [11], [12] and is guaranteed by choosing a suitable PRBS sequence as input. A2 is a property of ARX models [11], [12]. A3 is a consequence of the quantization noise in the temperature measurements which is often considered as a white noise [24]. Assumptions A4 and A5 are quite standard in system identification and are required by the method described in Section IV. The correctness of all these assumption is empirically validated in Section VI.

The identification problem to be solved can be stated as follows.

Problem 1: Estimate, for each core, the coefficients of $A(z^{-1})$, $B_i(z^{-1})$, $(i = 0, \ldots, N_c)$ starting from a set of N input–output collected samples $P(1), \ldots, P(N), T_k(1), \ldots, T_k(N)$.

IV. Thermal Model Identification

The solution of Problem 1 cannot be obtained by using the least squares (LS) method. In fact, because of the presence of the additive noise $v_k(t)$, the LS estimate is asymptotically biased, as shown in the following. To get a consistent estimate it is necessary to identify also the variance σ^2_{vk} of $v_k(t)$ so that the effect of the noise can be removed. In the sequel, we present our approach showing how it is possible to write a bilinear system of equations whose unknowns are the variance σ^2_{vk} and the coefficients to be identified. This system can thus be solved by iteratively applying two least squares formulas.

With reference to the generic k–th core, model (2) can be rewritten as follows

$$\bar{T}_k(t) + \sum_{i=1}^{n} a_i\, \bar{T}_k(t-i) = \sum_{j=0}^{N_c} \sum_{i=1}^{n} b_{ji}\, P_j(t-i) + w_k(t). \quad (6)$$

Define the regressor vector

$$\begin{aligned}\bar{\varphi}_k(t) = [\,&-\bar{T}_k(t-1) \ldots -\bar{T}_k(t-n)\, P_0(t-1)\, \ldots\, P_0(t-n)\\ &P_1(t-1)\, \ldots\, P_1(t-n)\, \ldots\, P_{N_c}(t-1)\, \ldots\, P_{N_c}(t-n)\,]^T \quad (7)\end{aligned}$$

and the parameter vector

$$\theta_k = \begin{bmatrix} a_1 \cdots a_n\; b_{01} \cdots b_{0n} \cdots\; b_{N_c 1} \cdots b_{N_c n} \end{bmatrix}^T. \quad (8)$$

Eq. (6) leads to the regression form

$$\bar{T}_k(t) = \bar{\varphi}_k^T(t)\, \theta_k + w_k(t). \quad (9)$$

By introducing also the vectors

$$\begin{aligned}\varphi_k(t) = [\,&-T_k(t-1) \ldots -T_k(t-n)\, P_0(t-1)\, \ldots\, P_0(t-n)\\ &P_1(t-1) \ldots P_1(t-n) \ldots P_{N_c}(t-1) \ldots P_{N_c}(t-n)\,]^T \quad (10)\\ \varphi_k^v(t) = [\,&-v_k(t-1)\, \ldots\, -v_k(t-n)\, 0 \ldots 0\,]^T \quad (11)\end{aligned}$$

and taking into account (5) we have

$$\varphi_k(t) = \bar{\varphi}_k(t) + \varphi_k^v(t). \quad (12)$$

Finally, by inserting (5) and (12) in (9) we get the regression form of the MISO ARX + noise model:

$$T_k(t) = \varphi_k^T(t)\, \theta_k + w_k(t) + v_k(t) - \varphi_k^{vT}(t)\, \theta_k. \quad (13)$$

Multiplying both sides of (13) by $\varphi_k(t)$ and applying the expectation operator $E[\cdot]$ we get

$$r_k = R_k\, \theta_k + E[\varphi_k(t)\, (w_k(t) + v_k(t) - \varphi_k^{vT}(t)\, \theta_k)] \quad (14)$$

where

$$r_k = E[\varphi_k(t)\, T_k(t)], \quad R_k = E[\varphi_k(t)\, \varphi_k^T(t)]. \quad (15)$$

By taking into account (12) and Assumptions A3-A5 we obtain

$$r_k = R_k\, \theta_k - E[\varphi_k^v(t)\, \varphi_k^{vT}(t)]\, \theta_k \quad (16)$$

and finally, because of Assumption A3

$$r_k = R_k\, \theta_k - \sigma^2_{vk}\, J\, \theta_k \quad (17)$$

where $J = \begin{bmatrix} I_n & 0 \\ 0 & 0 \end{bmatrix}$. From (17) it is clear that, due to the presence of the measurement noise $v_k(t)$ (i.e. of the variance σ^2_{vk}) the least squares estimate $\hat{\theta}_{LS} = R_k^{-1}\, r_k$ is biased. Note that both the entries of θ_k and the noise variance σ^2_{vk} are unknown and the number of relations in (17) coincides with the dimension of θ_k. Therefore, at least one more equation is needed to get a consistent estimate. For this purpose, consider first the vector of input samples

$$\begin{aligned}\varphi_P^d(t) = [\, &P_0(t-1)\, \ldots\, P_0(t-d)\, P_1(t-1)\, \ldots\, P_1(t-d)\, \ldots\\ &\ldots\, P_{N_c}(t-1)\, \ldots\, P_{N_c}(t-d)\,]^T, \quad (18)\end{aligned}$$

where d is a user-chosen parameter $(d \geq 1)$. If we multiply both sides of (13) by $\varphi_P^d(t)$ and apply the expectation operator by taking into account Assumption A5 we get

$$r_k^d = R_k^d\, \theta_k \quad (19)$$

where

$$r_k^d = E[\varphi_P^d(t)\, T_k(t)], \quad R_k^d = E[\varphi_P^d(t)\, \varphi_k^T(t)]. \quad (20)$$

This is a set of additional equations whose number depends on the free parameter d. Relations (17) and (19) constitutes a system of equations in the unknonws θ_k and σ^2_{vk}. This set is nonlinear because of the term $\sigma^2_{vk}\, J\, \theta_k$ in (17). Nevertheless, Eq. (17) exhibits a bilinear structure so that it is possible to solve the system by means of an iterative least squares algorithm. To this end define the following vector and matrix

$$\rho_k = \begin{bmatrix} r_k \\ r_k^d \end{bmatrix}, \quad \Sigma(\sigma^2_{vk}) = \begin{bmatrix} R_k - \sigma^2_{vk}\, J \\ R_k^d \end{bmatrix}. \quad (21)$$

The system of equations (17), (19) can thus be written in the compact form

$$\rho_k = \Sigma(\sigma^2_{vk})\, \theta_k. \quad (22)$$

An estimate of θ_k and σ^2_{vk} can thus be found by solving the following minimization problem

$$\min_{\theta_k, \sigma^2_{vk}} f(\theta_k, \sigma^2_{vk}) = \|\rho_k - \Sigma(\sigma^2_{vk})\, \theta_k\|^2. \quad (23)$$

The bilinear parametrization of the loss function allows to split the minimization problem into two standard least squares problems. Indeed, if σ_{vk}^2 is known the parameter vector θ_k can be estimated as

$$\hat{\theta}_k = \Sigma(\sigma_{vk}^2)^+ \rho_k \tag{24}$$

where $\Sigma(\sigma_{vk}^2)^+$ denotes the pseudoinverse of $\Sigma(\sigma_{vk}^2)$. On the other hand, given θ_k, an estimate of σ_{vk}^2 can be obtained from (17):

$$\hat{\sigma}_{vk}^2 = \frac{\theta_k^T J^T (R_k\,\theta_k - r_k)}{\theta_k^T J^T J \theta_k}. \tag{25}$$

It is thus possible to identify the unknowns θ_k and σ_{vk}^2 by means of an iterative least squares algorithm, as shown in Subsection IV-A. To determine an estimate of the driving noise variance σ_{wk}^2 we can still exploit the regression form (13). Multiplying both sides of (13) by $T_k(t)$ and applying the expectation we get

$$\sigma_{T_k}^2 = r_k^T\,\theta_k + E[T_k(t)\,(w_k(t) + v_k(t) - \varphi_k^{vT}(t)\,\theta_k)], \tag{26}$$

where $\sigma_{T_k}^2 = E[T_k^2(t)]$ and r_k has been defined in (16). By considering (5) and Assumptions A2-A5 it is easy to get

$$\sigma_{T_k}^2 = r_k^T\,\theta_k + \sigma_{wk}^2 + \sigma_{vk}^2. \tag{27}$$

from which it is possible to compute an estimate of σ_{wk}^2 once that the estimates of θ_k and σ_{vk}^2 have been computed. It is worth noting that the estimate of σ_{wk}^2 plays a key role in both model validation and actual temperature estimation, see Subsections IV-C and IV-D.

A. Identification algorithm

This subsection summarizes the whole procedure for identifying the thermal model of the generic k–th core of a CPU die. Since the CPU is composed of N_c cores, to identify the thermal dynamics of the whole CPU the identification algorithm described in the following has to be applied N_c times.

Iterative least squares (ILS) algorithm

The outputs of the algorithm are:

– an estimate of the parameter vector

$$\theta_k = \left[\, a_1 \cdots a_n \; b_{01} \cdots b_{0n} \cdots b_{N_c1} \cdots b_{N_cn} \,\right]^T$$

describing the thermal dynamics of the core, see (2) and (6);

– an estimate of the variance σ_{vk}^2 of the measurement noise $v_k(t)$, see (5);

– an estimate of the variance σ_{wk}^2 of the driving noise $w_k(t)$, see (2) and (6);

These estimates are computed starting from N samples of the measured k–th core temperature $T_k(1), T_k(2), \ldots, T_k(N)$ and N samples of the cores powers and uncore power $P_1(t), P_2(t), \ldots, P_{N_c}(t)$, collected in the vectors $P(1), P(2), \ldots, P(N)$, see (1). The identification procedure consists in the following steps.

(i) Compute, on the basis of the available input-output data, the sample estimates $\hat{r}_k$, $\hat{R}_k$, $\hat{r}_k^d$, $\hat{R}_k^d$ of the vectors and matrices r_k, R_k, r_k^d, R_k^d defined in (15) and (20):

$$\hat{r}_k = \frac{1}{N}\sum_{t=n+1}^{N} \varphi_k(t)\,T_k(t), \quad \hat{R}_k = \frac{1}{N}\sum_{t=n+1}^{N} \varphi_k(t)\,\varphi_k^T(t)$$

$$\hat{r}_k^d = \frac{1}{N}\sum_{t=d+1}^{N} \varphi_P^d(t)\,T_k(t), \quad \hat{R}_k^d = \frac{1}{N}\sum_{t=d+1}^{N} \varphi_P^d(t)\,\varphi_k^T(t)$$

where

$$\begin{aligned}\varphi_k(t) = [\,&-T_k(t-1)\ldots - T_k(t-n)\,P_0(t-1)\,\ldots\,P_0(t-n)\\ &P_1(t-1)\,\ldots\,P_1(t-n)\,\ldots\,P_{N_c}(t-1)\,\ldots\,P_{N_c}(t-n)\,]^T\\ \varphi_P^d(t) = [\,&P_0(t-1)\,\ldots\,P_0(t-d)\,P_1(t-1)\,\ldots\,P_1(t-d)\\ &\ldots\,P_{N_c}(t-1)\,\ldots\,P_{N_c}(t-d)\,]^T,\end{aligned}$$

The integer $d \geq 1$ in $\varphi_P^d(t)$ is a user-chosen parameter. We suggest to choose $d \geq n$. Finally, construct the vector $\hat{\rho}_k$ as in (21) and the matrix J:

$$\hat{\rho}_k = \begin{bmatrix}\hat{r}_k\\ \hat{r}_k^d\end{bmatrix}, \quad J = \begin{bmatrix} I_n & 0\\ 0 & 0\end{bmatrix}.$$

(ii) Set $i = 0$ and compute an initial estimate of the parameter vector θ_k. For instance, it is possible to start from the LS estimate $\hat{\theta}_k(0) = \hat{\theta}_{LS} = \hat{R}_k^{-1}\,\hat{r}_k$.

REPEAT

(iii) Set $i = i + 1$

(iv) Compute, on the basis of $\hat{\theta}_k(i-1)$, an estimate of the measurement noise variance:

$$\hat{\sigma}_{vk}^2(i) = \frac{\hat{\theta}_k^T(i-1)J^T(\hat{R}_k\,\hat{\theta}_k^T(i-1) - \hat{r}_k)}{\hat{\theta}_k^T(i-1)J^T J\hat{\theta}_k(i-1)}. \tag{28}$$

(v) Construct the matrix $\hat{\Sigma}\left(\hat{\sigma}_{vk}^2(i)\right)$ as in (21)

$$\hat{\Sigma}(\sigma_{vk}^2(i)) = \begin{bmatrix}\hat{R}_k - \hat{\sigma}_{vk}^2(i)\,J\\ \hat{R}_k^d\end{bmatrix}$$

and compute an estimate of the model parameters:

$$\hat{\theta}_k(i) = \hat{\Sigma}\left(\hat{\sigma}_{vk}^2(i)\right)^+ \hat{\rho}_k. \tag{29}$$

where $\hat{\Sigma}\left(\hat{\sigma}_{vk}^2(i)\right)^+$ is the pseudoinverse of $\hat{\Sigma}\left(\hat{\sigma}_{vk}^2(i)\right)$

UNTIL

$$\frac{\|\hat{\theta}_k(i) - \hat{\theta}_k(i-1)\|}{\|\hat{\theta}_k(i)\|} < \varepsilon, \tag{30}$$

where ε is a small positive number (the convergence threshold).

(vi) From the estimates $\hat{\theta}_k$ and $\hat{\sigma}_{vk}^2$ obtained after convergence, it is possible to get an estimate of the variance σ_{wk}^2. To this end compute first an estimate $\hat{\sigma}_{T_k}^2$ of the temperature variance: $\hat{\sigma}_{T_k}^2 = 1/N\sum_{t=n+1}^{N} T_k(t)^2$. Then, compute an estimate of σ_{wk}^2 as follows

$$\hat{\sigma}_{wk}^2 = \hat{\sigma}_{T_k}^2 - \hat{r}_k^T\,\hat{\theta}_k - \hat{\sigma}_{vk}^2. \tag{31}$$

Remark 2: Note that step (iv) solves the LS problem

$$\min_{\sigma_{vk}^2} f(\hat{\theta}_k(i-1), \sigma_{vk}^2) \tag{32}$$

whereas step (v) solves the LS problem

$$\min_{\theta_k} f(\theta_k, \hat{\sigma}_{vk}^2(i)). \tag{33}$$

It follows that

$$\begin{aligned} f(\hat{\theta}_k(i), \hat{\sigma}_{vk}^2(i)) &\leq f(\hat{\theta}_k(i-1), \hat{\sigma}_{vk}^2(i))\\ &\leq f(\hat{\theta}_k(i-1), \hat{\sigma}_{vk}^2(i-1)).\end{aligned} \tag{34}$$

The above property guarantees the convergence of the iterative algorithm.

B. Comparison with [16]

Compared with the Frisch scheme-based identification algorithm proposed in [16], the proposed ILS algorithm is simpler from the computational point of view. Indeed, it is based on the iterative application of the closed form expressions (29) and (28) whereas the Frich scheme-based method described in [16] relies on the minimization of a constrained loss function. The rationale behind the Frisch scheme consists in searching for the solution of the identification problem within a locus of solutions which are compatible with the covariance matrix of the noisy data. Therefore, the minimization has to be performed by means of some numerical search methods. For instance in [16], the MATLAB function *fminsearch* was exploited, that is based on the downhill simplex method. As shown in Section VI, the time requested by the ILS algorithm to identify the thermal model of a node (measured in MATLAB) is about two orders of magnitude lower than that requested by the algorithm in [16]. This feature plays an important role when the number of cores of a node is high.

Another advantage of the ILS algorithm is that it allows to develop a recursive version, which is of interest for online thermal management where the model has to updated to track parameter changes. The development of a recursive algorithm goes beyond the scope of the paper, however in the following we give a high-level outline:

– Eq. (22) can be rewritten as follows

$$\rho_k = \bar{R}_k\, \theta_k - \sigma_{vk}^2\, \bar{J}\, \theta_k \tag{35}$$

where (see (21)) $\bar{R}_k = \begin{bmatrix} R_k \\ R_k^d \end{bmatrix}$ and $\bar{J} = \begin{bmatrix} J \\ 0 \end{bmatrix}$.

– From (35) it is easy to get

$$\theta_k = \bar{\theta}_k + \bar{R}_k^{+} \sigma_{vk}^2\, \bar{J}\, \theta_k \tag{36}$$

where $\bar{\theta}_k = \bar{R}_k^{+}\, \rho_k$ and $\bar{R}_k^{+}$ is the pseudoinverse of $\bar{R}_k$.

– An estimation of $\bar{\theta}_k$ can be computed directly from the available data. If σ_{vk}^2 were zero (no additive noise), $\bar{\theta}_k$ would be an extended instrumental variable estimate of θ_k [12]. The estimate $\bar{\theta}_k$ can be computed in a recursive way [12].

– A recursive estimation of θ_k can thus be derived starting from (36) and the online estimate of $\bar{\theta}_k$.

Note that it is hard to get a recursive version of the algorithm in [16]. In fact, the locus of solutions within which estimating the thermal model of a core changes when new input-output samples become available.

C. Model validation

To validate the goodness of the proposed identification method and of the related assumptions A1-A5 it is possible to exploit the statistical properties of the residual of the MISO ARX + noise model described by Eqs. (2) and (5). By inserting (5) into (2) we get

$$A(z^{-1})\, T_k(t) = B(z^{-1})\, P(t) + w_k(t) - A(z^{-1})\, v_k(t). \tag{37}$$

By comparing the standard ARX model (2) with the noisy ARX model (37), it is easy to see that the residual of the latter is no longer a white process because it is given by

$$e_k(t) = A(z^{-1})\, T_k(t) - B(z^{-1})\, P(t) = w_k(t) - A(z^{-1})\, v_k(t). \tag{38}$$

According to Assumptions A2 and A3, the stochastic process $e_k(t)$ is the sum of the white process $w_k(t)$ and the moving average (MA) process $A(z^{-1})\, v_k(t)$ so that its autocorrelation function $r_{e_k}(\tau) = E[e_k(t)\, e_k(t-\tau)]$ is given by

$$\begin{aligned} r_{e_k}(0) &= \sigma_{vk}^2 \Big(\sum_{i=0}^{n} a_i^2 + \sum_{j=1}^{N_c} \sum_{i=1}^{n} b_{ji}^2 \Big) + \sigma_{wk}^2 \\ r_{e_k}(\tau) &= \sigma_{vk}^2 \Big(\sum_{i=0}^{n-\tau} a_i\, a_{i+\tau} + \sum_{j=1}^{N_c} \sum_{i=1}^{n-\tau} b_{ji}\, b_{j(i+\tau)} \Big), \quad 0 < \tau \leq n \\ r_{e_k}(\tau) &= 0, \quad \forall \tau > n, \end{aligned}$$

then, it behaves like the autocorrelation function of a moving average process of order n. It is thus possible to consider the normalized autocorrelation function

$$\gamma_k(\tau) = \frac{r_{e_k}(\tau)}{r_{e_k}(0)}, \quad \tau > 0 \tag{39}$$

and check its statistical properties [25]. In practice, to validate the methodology the following steps must be performed. Once that an identified model of the core $\hat{A}(z^{-1}), \hat{B}(z^{-1})$ has been obtained, the residuals are first computed by using (38): $\hat{e}_k(t) = \hat{A}(z^{-1})\, y_k(t) - \hat{B}(z^{-1})\, P(t)$. Then, the sample estimates $\hat{r}_{e_k}(0), \hat{r}_{e_k}(1), \hat{r}_{e_k}(2), \ldots, \hat{r}_{e_k}(M)$ of the autocorrelations can be computed, where $M > n$. Finally, the statistical properties of the normalized autocorrelation sequence $\hat{\gamma}_k(1), \hat{\gamma}_k(2), \ldots, \hat{\gamma}_k(M)$ will be tested by using the Bartlett's approximation in order to check whether $\hat{e}_k(t)$ behaves like a MA(n) process [25]. This test allows to validate Assumptions A2-A4. The validation of Assumption A5 can be performed from the sample cross-correlations between the residual $\hat{e}_k(t)$ and the input signals $P_0(t), P_1(t), \ldots, P_{N_c}(t)$. In fact, if A5 is true, this cross-correlation has to be "very small", indicating that the residual does not contain any further information generated by the input signals. Statistical tests to check this assumption are available in the identification literature [11], [12].

D. Estimation of the actual core temperature

The core models $\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_{N_c}$ identified by means of the ILS algorithm, may be used to estimate the corresponding actual core temperatures $\bar{T}_1(t), \bar{T}_2(t), \ldots, \bar{T}_{N_c}(t)$. To this end, consider the following state space representation of the k–th core ARX + noise model (2), (5):

$$x_k(t+1) = A\, x_k(t) + B\, P(t) + G\, w_k(t+1) \tag{40}$$

$$T_k(t) = C\, x_k(t) + v_k(t) = \bar{T}_k(t) + v_k(t) \tag{41}$$

where

$$A = \begin{bmatrix} -a_1 & 1 & 0 & \cdots & 0 \\ -a_2 & 0 & \ddots & \ddots & \\ \vdots & \vdots & & \ddots & \\ \vdots & \vdots & & & 1 \\ -a_n & 0 & \cdots & \cdots & 0 \end{bmatrix} \qquad B = \begin{bmatrix} b_{01} & \cdots & b_{N_c 1} \\ b_{02} & \cdots & b_{N_c 2} \\ \vdots & & \vdots \\ \vdots & & \vdots \\ b_{0n} & \cdots & b_{N_c n} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \qquad G = C^T.$$

Starting form this state space model, a Kalman filter can be implemented by using the previously estimated parameters and noise variances $(\hat{\theta}_k, \hat{\sigma}^2_{wk}, \hat{\sigma}^2_{vk})$. It is thus possible to compute the filtered temperature $\hat{\bar{T}}_k(t|t)$, which is the minimum variance estimate of the actual core temperature $\bar{T}_k(t)$ [26]. As shown in Section VI, the filtering error

$$\varepsilon_F(t) = \bar{T}_k(t) - \hat{\bar{T}}_k(t|t), \quad t = 1, 2, \dots \tag{42}$$

can also be used to assess the model performance.

V. CPU DIE FLOORPLAN IDENTIFICATION

The estimated core models $\hat{\theta}_1, \hat{\theta}_2, \dots, \hat{\theta}_{N_c}$ can be exploited to identify the CPU die floorplan. This allows to evaluate the influence of each core on the other ones from the thermal point of view and is of primary interest for optimizing the thermal map of the processor and mitigating hot spots.

Consider the following partition of the k–th core parameter vector θ_k, see (8):

$$\theta_k = \left[\, \theta_{ak} \; \theta^0_{bk} \; \theta^1_{bk} \; \cdots \; \theta^{N_c}_{bk} \,\right]^T \tag{43}$$

where

$$\theta_{ak} = \left[\, a_1 \cdots a_n \,\right]^T, \quad \theta^j_{bk} = \left[\, b_{j1} \cdots b_{jn} \,\right]^T, \tag{44}$$

and $j = 0, \dots, N_c$. The dissipated power $P_j(t)$ of the core j affects the actual temperature $\bar{T}_k(t)$ of the k–th core through the coefficients of θ^j_{bk}, see (8),(9) (the effects of the uncore power are represented by $\theta^{N_c}_{bk}$), then, a quantitative index for measuring the influence of the j–th core's power on the k–th core's temperature is given by

$$\eta_k(j) = \|\theta^j_{bk}\| / \eta^M_k, \tag{45}$$

where $\eta^M_k = \max\{\|\theta^0_{bk}\|, \|\theta^1_{bk}\|, \dots, \|\theta^{N_c}_{bk}\|\}$. Each core will be thus characterized by a set of indexes $\eta_k(0), \eta_k(1), \dots, \eta_k(N_c)$ and the whole node will be described by $N_c(N_c + 1)$ indexes from which it is possible to identify the node floorplan. Indeed, it is expected that:

- From the thermal viewpoint, the k–th core is mostly influenced by its dissipated power, so that $\eta^M_k = \|\theta^k_{bk}\|$ and then $\eta_k(k) = 1$.
- Among all other cores, only those in the neighborhood of the k–th one have a significant influence. The influence of the uncore power could be not negligible as well.

The neighbors of the k–th core can be identified as those whose dissipated powers are associated with an index $\eta_k(j)$ that exceeds a given threshold. It is worth noting that the knowledge of the node floorplan allows also to reduce the complexity of the cores' models. In fact, once that the neighbors of the generic k–th core have been detected, we can assume that only their dissipated powers have an influence on $\bar{T}_k(t)$. As a consequence, the input signals in the vector $P(t)$ (1) can be limited to the neighbors' powers and then the number of parameters in (8) can be reduced. The reduction in the model complexity can be a major factor when the number of cores of the node is quite high, especially in online applications. In this case, the identification procedure of the whole node can be divided in the following steps.

1) A first offline identification is performed for every core of the node. In particular, for each core, a MISO ARX + noise model is estimated by considering in the input vector $P(t)$ all the dissipated core powers $P_0(t), P_1(t), \dots, P_{N_c}(t)$.
2) The node floorplan is identified. First, for every core $k = 0, 1, \dots, N_c - 1$, the set of indexes $\eta_k(0), \eta_k(1), \dots, \eta_k(N_c)$ is evaluated. Then, the neighbors of the k–th core are identified by comparing the indexes with a given threshold $\bar{\eta}_k$, for instance $\bar{\eta}_k = 0.1\, \eta^M_k$. More precisely, the j–th core is assumed as a neighbor of the k–th core if $\eta_k(j) > \bar{\eta}_k$.
3) Once that the neighbors of each core have been detected, the identification procedure is repeated by considering different input vectors $P^k(t)$ for every core $k = 0, 1, \dots, N_c - 1$. In particular, the elements of the vector $P^k(t)$ are the dissipated powers of the neighbors of core k. Of course, it is expected that $P_k(t)$ belongs to $P^k(t)$. The number of parameters of the MISO ARX core models can thus be significantly reduced, see (8). The models with reduced complexity will thus be considered in the online identification, where the estimated parameters have to be updated to track model changes.

Remark 3: If the energy content of the input signals present remarkable differences, this must be taken into account in order to avoid misleading results. In this case the index $\eta_k(j)$ should be weighted by the normalized root mean square of the input signal $P_j(t)$: $\eta_k(j)' = \eta_k(j) * RMS_j / RMS_{max}$ where RMS_j is the root mean square of $P_j(t)$ and $RMS_{max} = \max\{RMS_0, RMS_1, \dots, RMS_{N_c}\}$.

VI. EXPERIMENTAL RESULTS

To evaluate the proposed methodology in an industrial relevant use case we use as a testbed a Tier-1 HPC system (namely Galileo, and located at CINECA) based on an IBM NeXtScale cluster. Each node of the system is equipped with 2 Intel Haswell E5-2630 v3 CPUs, with 8 cores with 2.4 GHz nominal clock speed and 85W Thermal Design Power (TDP). As regards the software infrastructure, SMP CentOS Linux distribution version 7.0 with kernel 3.10, runs on each node of the system.

We applied the methodology described in Section II to extract the thermal models of the eight core's CPU of a single node of this system. The sampling time was set to $T_{samp} = 2s$ and the considered data set consists in input-output sequences of length $5000s$ so that the number of available samples of each input and output signal is $N = 2500$. All the experiments have been performed in the presence of ambient temperature variations up to $6\ ^\circ C$. To account for the effects of ambient temperature variations on the estimated thermal models, the measured ambient temperature $T_{amb}(t)$ is subtracted from the measured core temperature $T_k(t)$ before applying the identification procedure. Therefore, the identified models will be effective in predicting the difference between the core and the ambient temperatures under constant or slowly-varying ambient temperatures, which is the most common scenario. The order n of each MISO ARX model has been set to 2 as in [16]. This choice has been validated by means of the procedure described in Subsection IV-C (see below). The ILS algorithm has been performed by setting $d = 10$ in step (i).

TABLE I
ESTIMATED DISCRETE-TIME POLES p_1, p_2 FOR EACH CORE AND ASSOCIATED TIME CONSTANTS τ_1, τ_2 IN THE CONTINUOUS-TIME DOMAIN.

	ILS algorithm				Approach [16]				ARX		OE	
	p_1	p_2	$\tau_1(s)$	$\tau_2(s)$	p_1	p_2	$\tau_1(s)$	$\tau_2(s)$	p_1	p_2	p_1	p_2
Core 0	0.850	0.016	12.31	0.48	0.846	0.024	11.97	0.54	0.440	−0.022	1.000	0.654
Core 1	0.843	0.004	11.74	0.36	0.841	0.007	11.57	0.41	0.488	−0.060	0.922	−0.055
Core 2	0.856	0.006	13.02	0.39	0.861	0.005	13.45	0.38	0.481	−0.048	0.757	0.009
Core 3	0.881	0.011	15.76	0.45	0.880	0.020	15.71	0.51	0.415	−0.046	0.654	0.366
Core 4	0.869	0.017	14.27	0.49	0.863	0.021	13.58	0.52	0.402	−0.036	0.907	0.019
Core 5	0.904	0.029	19.73	0.56	0.904	0.031	19.79	0.57	0.512	−0.031	0.872	0.134
Core 6	0.885	0.006	16.32	0.39	0.882	0.006	15.88	0.39	0.459	−0.064	0.999	0.119
Core 7	0.886	0.009	16.45	0.42	0.879	0.011	15.42	0.44	0.452	−0.065	0.828	−0.213

The convergence threshold in (30) has been fixed to $\varepsilon = 10^{-3}$. It has been observed that smaller values of ε increase the computational time without changing the obtained results. To compare the proposed approach with the state-of-the-art techniques adopted in the literature, the following approaches have been considered (see Subsection I-A concerning the related works): the Frisch-scheme based method described in [16], that belongs to the AutoRegressive plus noise methods (group (iii) in Subsection I-A); the ARX model identification, belonging to the family of AutoRegressive methods (group (ii)) and the Output Error (OE) model identification (group (i)).

Table I reports the discrete-time estimated poles of the eight cores and the associated time constants in the continuous-time domain. The results summarized in Table I lead to the following remarks:

– All the poles identified by means of the ILS algorithm are real and positive, according to the physics of thermal systems.

– The obtained poles and the associated time constants are in line with those estimated by means of the Frisch scheme-based method [16].

– The ARX and OE methods lead to some negative poles (the OE method presents also an unstable pole), that are not compliant with the physics of thermal systems, where only positive and stable poles can exist. For this reason, the associated time constants have not been reported.

This confirms that ARX and OE method are not suitable for extracting thermal models in the presence of a significant quantization noise.

The models obtained with the proposed method have been validated by means of the procedure described in Subsection IV-C. The middle picture of Fig. 2, that refers to core 0, reports the normalized autocorrelation $\gamma_0(\tau)$ for $\tau = 1, 2, \ldots, 20$ of the residual $\hat{e}_0(t)$ of the identified second order MISO ARX model. All values of $\gamma_0(\tau)$ for $\tau > 2$ lie within the 95% confidence level so that $\hat{e}_0(t)$ behaves like an MA(2) process and Assumptions A2-A4 are validated. As a comparison, the upper picture of Fig. 2 reports the normalized autocorrelation of the residual of an identified model of order $n = 1$. In this case, the model is not validated since there are a lot of values $\gamma_0(\tau)$ for $\tau > 1$ that falls outside the confidence level. It is thus possible to confirm that $n = 2$ is a proper order for the thermal model of the core. The models of the cores $1, 2, \ldots, 7$ lead to similar results. The cross-correlations between the residual and the input signals have also been tested. As an example, the lower picture of Fig.2 reports the normalized cross-correlation between $e_0(t)$ and $P_0(t)$ and the 95% confidence level. This confirms the validity of Assumptions A5. This test has been applied to other input signals and/or residuals and the obtained results are similar.

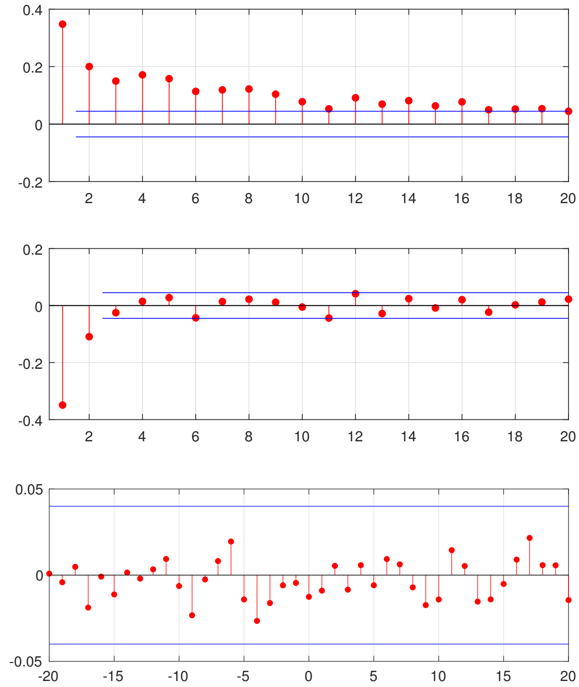


Fig. 2. Normalized autocorrelation function of the residual $\hat{e}_0(t)$ of core 0 for identified models of order $n = 1$ (upper picture) and $n = 2$ (middle picture). Normalized cross-correlation function between the residual $\hat{e}_0(t)$ and the input signal $P_0(t)$ (lower picture).

Another index that has been used for evaluating the identification performance is based on the filtering error $T_k(t) - \hat{\hat{T}}_k(t|t)$, see Subsection IV-D. In fact, since $\hat{\hat{T}}_k(t|t)$ is the best estimate of the actual core temperature that can be obtained from the data and the identified model, the corresponding estimation error should be contained within the $\pm 1°C$ confidence interval, which is the precision of the quantized temperature readings. In particular, for each core, the percentage of elements of the error sequence (42) that falls within this interval has been adopted as performance index. The obtained values

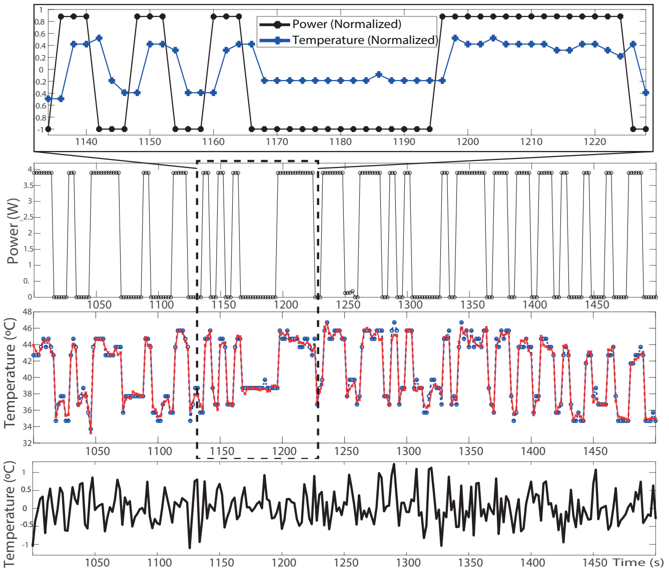


Fig. 3. Core 0 identification traces. From top to bottom: i) Power vs. temperature (normalized). ii) Dissipated power $P_0(t)$. iii) Measured temperature (blue) and filtered temperature (red). iv) Filtering error $(T(t) - \hat{\hat{T}}_k(t|t))$.

range in the interval $94\% - 97\%$. This confirms the goodness of the approach. As an example, Figure 3 (middle figures) reports a $500s$-long part of the power $P_0(t)$ dissipated by Core 0, the measured temperature of Core 0 and the associated filtered temperature.The filtering error $T(t) - \hat{\hat{T}}_k(t|t)$ is also reported (lower figure). The upper figure reports a zoom of the power and measured temperature of the dashed box, normalized for fitting a range $0 - 1$ and without offset. This subfigure shows that the temperature od core 0 is affected mainly by the power $P_0(t)$ even if the influence of the neighbors' powers and uncore power may be not negligible, see the discussion at the end of the section.

To compare the proposed method with the approach [16] w.r.t. the computation time we have collected 20 input-output traces each of 12 hours from the Tier-1 HPC system nodes. For each of these trace, a MISO ARX + noise model for each core has been identified by using both ILS algorithm and the approach [16]. Since each trace refers to two 8-core CPUs, the total number of identified model is 320 (16 models for each trace). The mean time requested to identify the thermal model of a node (measured by using the MATLAB function *cputime*) is reported below:

	ILS algorithm	Approach [16]
Mean time per node	0.0040	0.1975

It can be seen that the computation time associated with the proposed identification algorithm is about two orders of magnitude lower than that requested by the algorithm in [16], making the proposed approach more suitable for online usage, especially when the number of cores per node is high.

The estimated MISO ARX models of the eight cores can be exploited to identify the CPU die floorplan, as discussed in Section V. Figure 4(b) reports, for each core k, the set of indexes $\eta_k(0), \eta_k(1), \ldots, \eta_k(8)$ computed by using (45), describing the influence of the dissipated powers $P_0(t), P_1(t), \ldots, P_8(t)$ on the core temperature $T_k(t)$ ($P_8(t)$ denotes the uncore power $P_{unc}(t)$). As expected, each core is affected mainly by its dissipated power ($\eta_k(k) = 1$ for every core $k = 0, \ldots, 7$). By taking a closer look at Fig. 4(b) it is possible to divide the cores into two groups. For instance, we can note note that Core 0, apart from the effect of its dissipated power $P_0(t)$, is mainly influenced by the dissipated power of another core (Core 2). Similar behaviors are exhibited by Cores 1, 6 and 7. A different behavior characterizes Cores 2, 4, 3 and 5 that, apart from their dissipated powers, are mainly affected by the powers of two other cores. For example, Core 4 is mainly influenced by Cores 2 and 6. It is thus possible to infer that Cores 0, 1, 6 and 7 are side cores whereas Cores 2, 3, 4 and 5 are central cores. This leads to the estimated CPU die floorplan shown in Fig. 4(a).

VII. Conclusion

In this manuscript we have proposed a novel approach to identify the thermal models of real datacenter's processor, and the core's relative position in the die. In this scenario standard ARX approaches cannot extract physical valid models. For this reason, the thermal dynamics of each core has been represented by means of a MISO ARX model with additive noise on the output. The proposed approach is capable of estimating the variance of the additive noise and to compensate it in the estimation of ARX model parameter. Thanks to the proposed methodology we are capable of extracting physically-valid thermal models from real in production compute nodes with a prediction error within the quantization error.

In future works we will combine these thermal models with model predictive controllers to maximize the computing performance and cooling efficiency of the datacenter's computing elements.

References

[1] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *38th Annual International Symposium on Computer Architecture (ISCA 2011)*, June 2011, pp. 365–376.

[2] ETP4HPC. (2017) Strategic research agenda. [Online]. Available: http://www.etp4hpc.eu/sra.html

[3] A. Bartolini, C. Conficoni, R. Diversi, A. Tilli, and L. Benini, "Multiscale thermal management of computing systems-the multitherman approach," *IFAC PapersOnLine*, vol. 50, no. 1, pp. 6709–6716, 2017.

[4] J. Gao and R. Jamidar, "Machine learning applications for data center optimization," 2014.

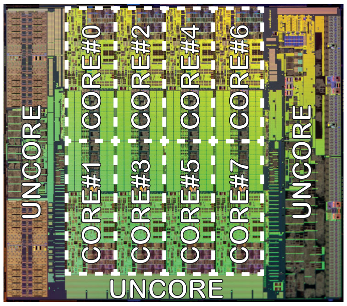
[5] M. Pore, Z. Abbasi, S. K. S. Gupta, and G. Varsamopoulos.

[6] M. Ot, T. Wilde, and H. Ruber, "Roi and tco analysis of the first production level installation of adsorption chillers in a data center," in *16th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (ITherm 2017)*, May 2017, pp. 981–986.

[7] A. Moskovsky, E. Druzhinin, A. Shmelev, V. Mironov, and A. Semin, "Server level liquid cooling: Do higher system temperatures improve energy efficiency?" *Supercomputing Frontiers and Innovation*, vol. 3, no. 1, pp. 67–74, 2016.

[8] A. Bartolini, R. Diversi, D. Cesarini, and F. Beneventi, "Self-aware thermal management for high performance computing processors," *IEEE Design & Test*, vol. 35, no. 5, pp. 28–35, 2018.

[9] R. Diversi, A. Bartolini, A. Tilli, F. Beneventi, and L. Benini, "SCC thermal model identification via advanced bias-compensated least-squares," in *2013 Design, Automation Test in Europe Conference Exhibition (DATE 2013)*, March 2013, pp. 230–235.



(a)

	$\eta(0)$	$\eta(1)$	$\eta(2)$	$\eta(3)$	$\eta(4)$	$\eta(5)$	$\eta(6)$	$\eta(7)$	$\eta(8)$
Core 0	1.000	0.038	0.189	0.026	0.025	0.043	0.015	0.032	0.088
Core 1	0.053	1.000	0.043	0.152	0.050	0.018	0.049	0.021	0.078
Core 2	0.171	0.053	1.000	0.046	0.107	0.046	0.008	0.049	0.074
Core 3	0.040	0.099	0.052	1.000	0.036	0.182	0.039	0.027	0.118
Core 4	0.018	0.038	0.157	0.039	1.000	0.035	0.171	0.046	0.089
Core 5	0.041	0.005	0.058	0.088	0.052	1.000	0.041	0.172	0.089
Core 6	0.016	0.030	0.033	0.030	0.216	0.030	1.000	0.038	0.072
Core 7	0.021	0.012	0.028	0.019	0.031	0.127	0.037	1.000	0.097

(b)

Fig. 4. CPU die floorplan identification: (a) estimated floorplan based on the table on the right; (b): the k–th row of the table, related to Core k, reports the set of indexes $\eta_k(0), \eta_k(1), \ldots, \eta_k(8)$ describing the influence of the dissipated powers $P_0(t), P_1(t), \ldots, P_8(t)$ on the core temperature $T_k(t)$ ($P_8(t)$ denotes the uncore power $P_{unc}(t)$).

[10] R. Diversi, A. Tilli, A. Bartolini, F. Beneventi, and L. Benini, "Bias-compensated least squares identification of distributed thermal models for many-core systems-on-chip," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 61, no. 9, pp. 2663–2676, 2014.

[11] L. Ljung, *System Identification – Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[12] T. Soderstrom and P. Stoica, *System Identification*. Cambridge, UK: Prentice-Hall, 1989.

[13] Y. Zhao, A. Fatehi, and B. Huang, "A data-driven hybrid ARX and markov chain modeling approach to process identification with time-varying time delays," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4226–4236, 2017.

[14] F. Guo, O. Wu, Y. Ding, and B. Huang, "A data-based augmented model identification method for linear errors-in-variables systems based on em algorithm," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 11, pp. 8657–8665, 2017.

[15] R. Zhang and J. Tao, "A nonlinear fuzzy neural network modeling approach using an improved genetic algorithm," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 7, pp. 5882–5892, 2018.

[16] R. Diversi, A. Bartolini, F. Beneventi, and L. Benini, "Thermal model identification of supercomputing nodes in production environment," in *42nd Annual Conference of IEEE Industrial Electronics Society (IECON 2016)*, 2016, pp. 4838–4844.

[17] F. Beneventi, A. Bartolini, A. Tilli, and L. Benini, "An effective gray-box identification procedure for multicore thermal modeling."

[18] Y. Yang, Z. Gu, C. Zhu, R. P. Dick, and L. Shang, "Isac: Integrated space-and-time-adaptive chip-package thermal analysis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 26, no. 1, pp. 86–99, 2007.

[19] R. Cochran and S. Reda, "Consistent runtime thermal prediction and control through workload phase detection," in *Design Automation Conference*, June 2010, pp. 62–67.

[20] A. K. Coskun, T. S. Rosing, and K. C. Gross, "Utilizing predictors for efficient thermal management in multiprocessor socs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1503–1516, 2009.

[21] D. C. Juan, H. Zhou, D. Marculescu, and X. Li, "A learning-based autoregressive model for fast transient thermal analysis of chip-multiprocessors," in *17th Asia and South Pacific Design Automation Conference*, 2012, pp. 597–602.

[22] A. Bartolini, M. Cacciari, A. Tilli, and L. Benini, "Thermal and energy management of high-performance multicores: Distributed and self-calibrating model-predictive controller," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 170–183, 2013.

[23] S. Reda, K. Dev, and A. Belouchrani, "Blind identification of thermal models and power sources from thermal measurements," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 680–691, 2018.

[24] B. Widrow, I. Kollàr, and M.-C. Liu, "Statistical theory of quantization," *IEEE Transactions on Instrumentation and Measurement*, vol. 45, no. 2, pp. 353–361, 1996.

[25] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. Hoboken, New Jersey: John Wiley & Sons, 2016.

[26] B. D. O. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

Roberto Diversi received the "Laurea" degree in Electronic Engineering in 1996 and the Ph.D. degree in System Engineering in 2000 both from the University of Bologna, Bologna, Italy. He is currently Associate Professor with the Department of Electrical, Electronic and Information Engineering, University of Bologna. His research interests are mainly in System Identification Theory and Applications, Fault Diagnosis, Signal Processing.

Andrea Bartolini received a Ph.D. degree in Electrical Engineering from the University of Bologna, Italy, in 2011. He is currently Assistant Professor in the Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna. Before, he was Post-Doctoral researcher in the Integrated Systems Laboratory at ETH Zurich. Since 2007 Dr. Bartolini has published more than 100 papers in peer-reviewed international journals and conferences with focus on dynamic resource management for embedded and HPC systems.

Luca Benini holds the chair of digital Circuits and systems at ETHZ and is Full Professor at the University of Bologna. Dr. Benini's research interests are in energy-efficient computing systems design, from embedded to high-performance. He is also active in the design ultra-low power VLSI Circuits and smart sensing micro-systems. He has published more than 1000 peer-reviewed papers and five books. He is a Fellow of the ACM and a member of the Academia Europaea. He is the recipient of the 2016 IEEE CAS Mac Van Valkenburg award and of the 2019 IEEE TCAD Donald O. Pederson Best Paper Award.