INTEGRATION, the VLSI journal 46 (2013) 33–43

Contents lists available at SciVerse ScienceDirect

INTEGRATION, the VLSI journal

journal homepage: www.elsevier.com/locate/vlsi

A combined sensor placement and convex optimization approach for thermal management in 3D-MPSoC with liquid cooling

Francesco Zanini [a,*], David Atienza [b], Giovanni De Micheli [a]

[a] Integrated Systems Lab (LSI), Switzerland
[b] Embedded Systems Lab (ESL), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ARTICLE INFO

Available online 17 January 2012

Keywords:
Thermal
Management
Placement
3D
MPSoC
Liquid
Cooling

ABSTRACT

Modern high-performance processors employ thermal management systems, which rely on accurate readings of on-die thermal sensors. Systematic tools for analysis and determination of best allocation and placement of thermal sensors is therefore a highly relevant problem. Moreover liquid cooling has emerged as a promising solution for addressing the elevated temperatures in 3D *Multi-Processor Systems-on-Chips* (MPSoCs).

In this work, we present a combined sensor placement and convex optimization approach for thermal management in 3D-MPSoC with liquid cooling. This approach first finds the best locations inside the 3D-MPSoC where thermal sensors can be placed using a greedy approach. Then, the temperature sensing information is subsequently used by our convex-based thermal management policy to optimize the performance of the MPSoC while guaranteeing a reliable working condition.

We perform experiments on a 3D multicore architecture case-study using benchmarks ranging from web-accessing to playing multimedia. Our results show a reduction up to 10× in the number of required sensors. Moreover our policy satisfies performance requirements, while reducing cooling energy by up to 72% compared with traditional state of the art liquid cooling techniques. The proposed policy also keeps the thermal profile up to 18 °C lower compared with state of the art 3D thermal management techniques using variable-flow liquid cooling.

1. Introduction

Today, several commercial multicore architectures ranging from few cores to several tens of cores, such as Sun's Niagara [14], are available. Power and thermal management are important challenges for multicore systems [15], and become even more critical with 3D integration. In the last years, thermal management techniques received a lot of attention. Many state of the art thermal control policies manage power consumption via *dynamic voltage and frequency scaling* (DVFS) [19]. DVFS can be targeted to power density reduction, which has the effect of reducing overall temperature. However, these techniques do not directly avoid hot-spots [16,22].

Moreover, heat removal is more difficult within 3D stacks using conventional air cooling methods [7]. Liquid cooling is a potential solution to address the high temperatures in 3D chips [13], due to the higher heat removal capability of liquids in comparison to air. Liquid cooling is performed by attaching a cold plate with built-in microchannels, and/or by fabricating microchannels in the silicon layers of the 3D architecture. Then, a coolant fluid is pumped through the microchannels to remove the heat. The flow rate of the pumps can be altered dynamically, and the pump power consumption increases quadratically with the increase in flow rate [13]. Thus its contribution to the overall system energy is significant [29], so there is the need for the development of thermal policies that consider pumping power to better exploit this new cooling technology.

Moreover, the problem with all aforementioned techniques is that they require online thermal profile information from the chip to perform frequency assignment optimization. Many solutions are based on techniques trying to reduce temperature differences between thermal sensors and hot-spots by using the minimum number possible of sensors for a certain accuracy. The problem with these approaches is that since hot-spots are application dependent, there is no guarantee that all hot-spots are detected during the lifetime of the device. There is basically a trade-offs between the number of sensors and the accuracy of the measurement. The goal of a smart sensor allocation strategy is to minimize the number of sensors while maximizing the thermal profile estimation accuracy.

In this work we focus on a combined sensor placement and convex optimization approach for thermal management in

* Corresponding author.
E-mail addresses: francesco.zanini@epfl.ch, francesco.zanini@gmail.com (F. Zanini), David.Atienza@epfl.ch (D. Atienza), Giovanni.DeMicheli@epfl.ch (G. De Micheli).

0167-9260/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.vlsi.2011.12.003

3D-MPSoC with liquid cooling. This method finds first the best locations inside the 3D-MPSoC where thermal sensors can be placed. To this end, it analyzes the balanced state-space realization of the system and the Hankel singular values decay rate [38,46]. According to user designer accuracy requirements, the number of states of the reduced order model is fixed and a specific location is assigned to each sensor. Once sensors are placed, temperature sensing information is then used by the thermal management policy. The thermal management policy uses these information to estimate the 3D-MPSoC thermal profile and uses both DVFS and a variable-flow liquid cooling to meet the desired requirements. The optimization requirements are expressed by an objective cost function consisting of two terms. The first one is related to the overall system power minimization (MPSoC power consumption and pump power consumption) and the second one to the performance loss (undone work). Then, the problem is modeled using a receding horizon approach [20] based on *convex optimization* [24]. The optimization process including also the thermal profile estimation is applied at run-time using the convex-solver proposed by [25]. At this stage the convex solver finds the optimum frequency assignment for the inputs of the MPSoC system that will maximize performance under temperature constraints. These operations have been performed on standard processors (i.e., Core 2 Duo running at 2 GHz) in few tenth of microseconds [38]. This time is 3 orders of magnitude smaller as compared with the time the policy is applied (i.e., 10 ms).

We have validated the proposed approach on a 3D multicore architecture case study, based on Niagara T1 UltraSparc2 cores [14], using benchmarks ranging from web-accessing to playing multimedia. Our results show a reduction up to 10× in the number of required sensors. Moreover, the proposed sensor location technique relies on a greedy approach that makes the sensor placement algorithm not computationally intensive. In addition to that, scenarios with dangerous thermal profiles are avoided while satisfying the application performance requirements. In addition, cooling energy is reduced by up to 72% compared with state of the art liquid cooling policies. In addition, the proposed policy keeps the average thermal profile up to 18 °C lower compared with state of the art polices using variable-flow liquid cooling, like [29].

2. Related work

A study of the thermal profile estimation problem has been analyzed in [43,44]. The problem with these approaches is that since hot-spots are application dependent, there is no guarantee that all hot-spots are detected during the lifetime of the device. In [40] the authors select the location of the sensing element according to a Gramian-based sensor strategy. In [41] the problem of making a system observable is solved by employing of graph theory. The problem of choosing a set of measurements from a much larger set that also minimizes the estimation error is solved by [42] using a convex optimization based approach. This last method approximately solves the problem and has no guarantee that the performance gap is always small. In [27], the authors present a sensor placement technique based on a design space exploration of the observability matrix of the MPSoC model. The problem with this technique is that it is unfeasible for large systems such as 3D-MPSoCs including liquid cooling.

The use of convection in microchannels to cool down high power density chips has been an active area of research since the initial work by Tuckerman et al. [12]. The heat removal capability of interlayer heat-transfer with pin-fin in-line structures for 3D chips is investigated in [13]. Also, several works [8–11] have explored the feasibility of having liquid cooling as cooling method for 3D-MPSoCs. Then, prior liquid cooling work in [10] evaluates existing thermal management policies on a 3D system with a fixed-flow rate setting.

Accurate thermal modeling of liquid cooling is critical in the design and evaluation of systems and policies. HotSpot [16] is a thermal model tool that calculates transient temperature response given the physical and power consumption characteristics of the chip. The latest versions of HotSpot include 3D modeling capabilities and liquid-cooled systems as well [17]. Finally, 3D-ICE [18] is a new thermal modeling tool specifically designed for 3D stacks, and includes interlayer liquid cooling modeling capabilities.

Many researchers in computer architecture have recently focused on thermal control for *Multi-Processor System on Chips* (MPSoCs) [19,22]. Processor power optimization and balancing using DVFS have been proposed in several works [19,37]. The work proposed by [39], performs thermal management by controlling the fan speed and applying voltage/frequency scaling to minimize the total power consumption of both the processors and the cooling systems. However in all aforementioned policies there is not guarantee to avoid hot-spots by performing this optimization. The reason is because the policy targets power optimization and not hot-spot avoidance.

More advanced solutions apply the concepts of model-predictive control to turn the control from open-loop to closed-loop [20,21]. In [32] a chip-level power control algorithm based on optimal control theory is presented. This algorithm can control the power consumption of the MPSoC and can maintain the temperature of each core below a specified threshold. In [31] a similar concept is tailored for multimodal video sensor nodes. In [26] a convex optimization-based approach is presented. The problem with all aforementioned techniques is that they are based on DVFS and target 2D circuits with no active cooling mechanism such as variable-flow liquid cooling. In [7,29], thermal management methods for 3D-MPSoCs using a variable-flow liquid cooling have been proposed. These policies use simple heuristics to control the temperature profile of the 3D-MPSoC while ensuring performance requirements to be satisfied. In this paper we compare the proposed method with state-of-the-art approaches including both air and liquid cooling policies.

3. Modeling 3D systems with liquid cooling

This paper deals with 3D-MPSoCs stacking two or more dies. As an example, Fig. 1(a)–(c) shows a 3D system consisting of 4-tiers. There are four silicon layers (A, B, C, D) (with various functional units grouped into p islands with independent clock frequency and voltage supplies), where *microchannels* are etched in silicon bulk for liquid cooling. The model abstracts the interconnect on chip as copper layers (A, B, C, D). For every silicon layer there is a total of nc linear microchannels $Ch_1 \ldots Ch_{nc}$. Microchannels are assumed to be equal in dimensions and a uniform coolant flux is assumed in channels of the same layer. All microchannels belonging to the same layer are connected to a pump. In the model shown in Fig. 1(c) there is a total of four pumps connected to the microchannels of the four silicon layers. Fluid flows through channels belonging to different layers with different flow rates, according to the power of each pump. The liquid flow rate provided by each pump can be dynamically altered at run-time.

We would like to highlight that microchannels for 3D-IC cooling has been proposed previously by different research labs [1,2,13], including our industrial partner in this work: IBM. Typically, as mentioned in the literature, microchannels are etched on the back-side of each silicon tier, to enable forced convective cooling. Typical cost of microchannels manufacturing is 20% of the total chip cost, as shown in related work [3].

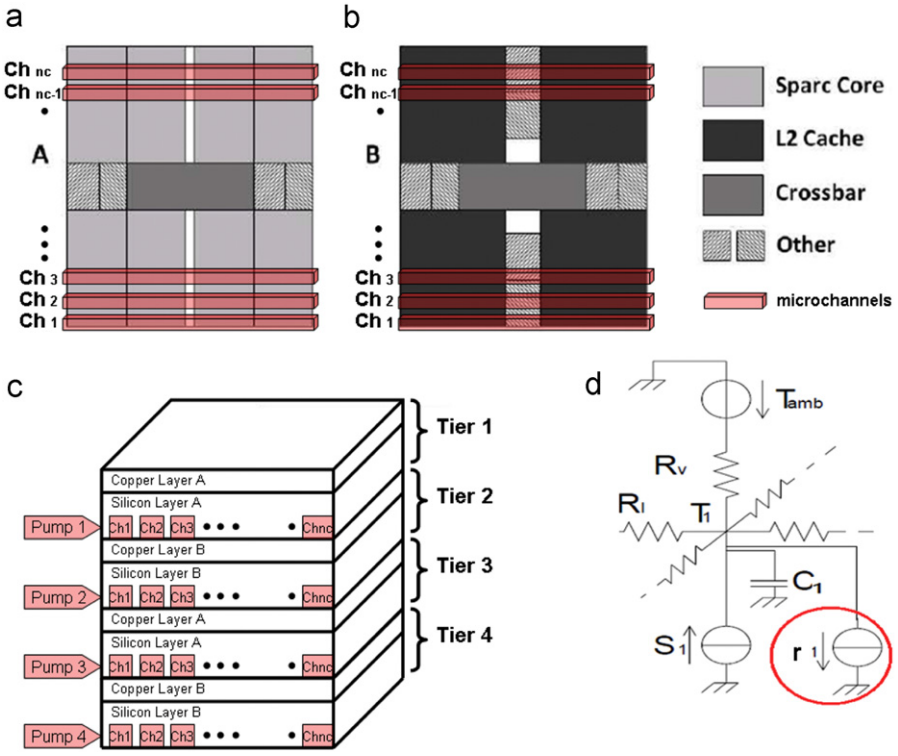


Fig. 1. 3-D stacked MPSoC with liquid cooling: silicon layer type-A (a), silicon layer type-B (b), overall MPSoC view (c), resistive network model (d).

In our explorations, we are using single-phase liquid cooling, using water as the coolant. This is the same applied coolant used in previous work [1–4,13]. The coolant does not need to be refilled. As shown in previous work [4], we are assuming that water comes from a reservoir, feed using a centrifugal pump, to a group of 3D ICs with liquid cooling, which are used in future data-centers. In fact, this is the same setup, but using liquid-cooled heat sinks for 2D chips, in the AQUASAR data center server developed by IBM zurich [5]. For coolant reliability analysis, we believe this is not the scope of our research effort, and it is being performed by our mechanical engineering partners in IBM [6]. In this respect, we are following the same operating conditions, in terms of fluid density, maximum inlet pressure and flow rate.

3.1. 3D heat propagation model

Our 3D thermal model is based on finite-element analysis, as used by typical system-level thermal analysis tools [18]. Heat propagation is modeled by thermal resistances and capacitances. To model the 3D-MPSoC architecture we use a state-space representation of the thermal system. A state space representation is a mathematical model of a physical system as a set of input, output and state variables related by first-order differential equations. The state variables are the variables that can represent the entire status of the system at any given time. To model the architecture shown in Fig. 1(a)–(c) we propose an extension of the model presented in [26]. In particular, the active cooling (for cell i) is modeled by a current sink r_i, as shown in Fig. 1(d) and highlighted by the circle. This current sink models the capability of the cooling system to remove heat in a specific location of the MPSoC.

Following [26], we model the heat propagation process as

$$\mathbf{t}_{\tau+1} = \mathbf{A}\mathbf{t}_{\tau} + \mathbf{B}\mathbf{p}_{\tau} \quad (1)$$

We assume that the total number of cells in all layers of the 3D-MPSoC structure is n, the total number of cores is p and the total number of pumps is z. Matrices $\mathbf{A}\epsilon\Re^{n\times n}$ and $\mathbf{B}\epsilon\Re^{n\times(p+z)}$ describe the heat propagation properties of the MPSoC. At time τ, the temperature of the next simulation step of cell i, i.e. $(\mathbf{t}_{\tau+1})_i$ can be computed thanks to Eq. (1). In this model $\mathbf{t}_{\tau+1}$ is the state vector and $\mathbf{p}\epsilon\Re^{p+z}$ is the input vector. The first p entries are the normalized power consumption for each of the p frequency islands (cores), while the remaining z entries are the normalized cooling power for each of the z pumps. The relation between the frequency assignment at time τ, $\mathbf{f}_{\tau}\epsilon\Re^{p}$, and the power consumption is assumed to be quadratic [16].

The law that relates the microchannel flow-rate to heat extraction has been taken from [18]. However, we consider that the amount of heat r_i extracted in cell i by the fluid in the microchannel controlled by pump j can be approximated by

$$r_i = m_j \cdot \gamma_{i,j} \cdot (t_i - t_{fluid}) \quad (2)$$

where the fluid temperature is t_{fluid}, t_i is the temperature of cell i and $\gamma_{i,j}$ is the constant modeling the channel heat extraction properties. Vector $\mathbf{m}\epsilon\Re^{z}$ is the normalized amount of heat that can be extracted for each of the z independent pumps. Thus, by varying vector $\mathbf{m}$, the cooling power (flow rate of the cooling liquid) is varied to achieve the desired heat extraction. In our model, we used the temperature mapping from [18] to derive $\gamma_{i,j}$. Experiments have shown that by updating $\gamma_{i,j}$ every time the policy is applied (10 ms in our simulation setup), our approximation leads to a maximum error up to $\pm 5\%$. Moreover, even if there are inaccuracies between the real and the simulated MPSoC model, the error does not propagate during the run-time execution of the system. The temperature profile of the MPSoC is indeed generated from real thermal sensors data every time the policy is applied (10 ms for the experimental setup used).

3.2. Workload model

The workload is generated from higher-level software layers (e.g., operating system). For each p clock islands (cores), the

workload is defined as the minimum value of the clock frequency that the functional unit should have to execute the required tasks within the specified system constraints.

The workload requirement at time τ is defined as a vector $\mathbf{w}_\tau \epsilon \Re^p$, where $(\mathbf{w}_\tau)_i$ is the workload requirement value for input i at time τ. $(\mathbf{w}_\tau)_i$ is the frequency that cores associated with input i from time τ to time $\tau+1$ should have in order to satisfy the desired performance requirement coming from the scheduler.

We assume a continuous control on the frequency ranging from $\mathbf{f}_{\min}$ to a max value $\mathbf{f}_{\max}$, the maximum frequency at which the cores can process data, namely

$$\mathbf{f}_{\min} \preceq \mathbf{w}_\tau \preceq \mathbf{f}_{\max} \forall \tau \tag{3}$$

When $(\mathbf{w}_\tau)_i > (\mathbf{f}_\tau)_i$, the workload cannot be processed and so it needs to be stored and rescheduled in the following clock cycles. The way we measure the performance of the system in achieving the requested workload requirements at time τ is given by the vector $\mathbf{u}_\tau \epsilon \Re^p$.

$$\mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau \tag{4}$$

We call $\mathbf{u}_\tau$ the undone workload at time τ and it expresses the difference at time τ between the requested workload and the workload that is actually executed by the MPSoC.

4. Global optimization approach overview

In the model described by Eq. (1), a state is required for every block composing the floorplan. The reason is because we need n states to store n temperatures values. This requirement is expensive in terms of computational requirements for high accuracy MPSoC models. The higher the number of states modeling the MPSoC, the higher the number of sensors required for its state estimation. This could be a problem in case of a detailed model of a complex 3D-MPSoC including liquid cooling.

The concept behind this work is a combined sensor placement and convex optimization approach for thermal management in 3D-MPSoC with liquid cooling. This approach first finds the best locations inside the 3D-MPSoC where thermal sensors can be placed using a greedy approach. Then, the temperature sensing information is subsequently used by our convex-based thermal management policy to optimize the performance of the MPSoC while guaranteeing a reliable working condition.

The advantage of the combined approach is an efficient method to solve both the sensor placement, the model order reduction and the thermal management of the 3D-MPSoC system problems at the same time with a reduced computational cost. The block diagram of the proposed algorithm is presented in Fig. 2. The proposed methodology consists of two phases: a design-time phase and a run-time phase.

During the design-time phase the thermal management system is defined. The reduced order MPSoC thermal model and the sensor placement are the outputs of this off-line phase. The concept behind the proposed sensor placement technique is based on an analysis of the balanced state-space realization of the 3D-MPSoC system and its Hankel singular values decay rate. The Hankel singular values are subject to decay and they decrease at a rate proportional to their value. This rate is called Hankel singular values decay rate. The number of states of the reduced order model is fixed according to user designer accuracy requirements, and a specific location is assigned to each sensor.

During the run-time phase the defined thermal management system solves the frequency assignment problem using a predictive horizon methodology applied to the reduced order 3D-MPSoC thermal model. First, during this phase, the reduced order system state vector $\mathbf{x}$ is estimated thanks to a simple state

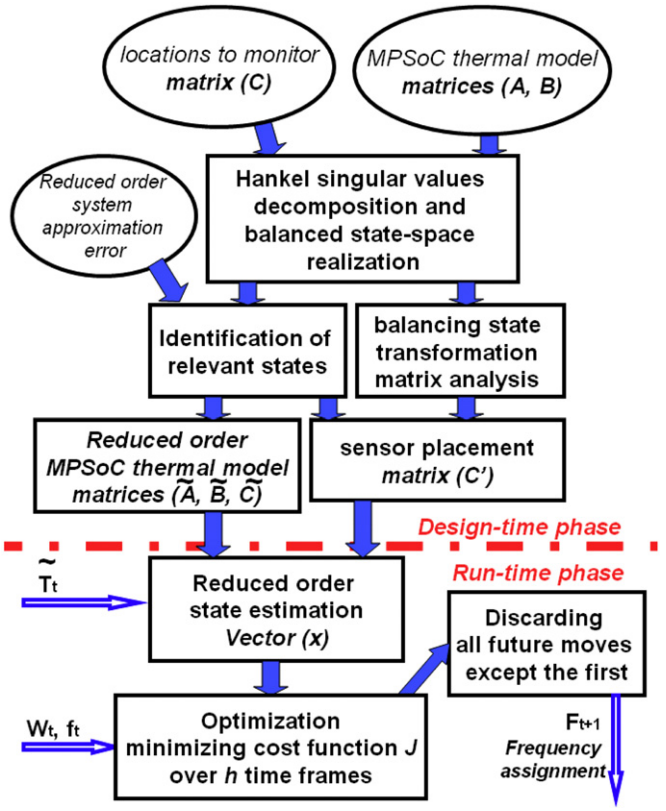


Fig. 2. Proposed policy global overview block diagram.

estimator (i.e. Kalman filter) and measurements coming from thermal sensors. Then, this information is used by the thermal model to perform the optimization on the reduced-order 3D-MPSoC model pre-defined in the design-time phase. The control problem is formulated over an interval of h time steps, which starts at current time τ. The result of the optimization is an optimal sequence of future control moves (i.e., frequency settings for both the cores and the liquid cooling pumps). Only the first sample of such a sequence is actually applied to the process, the remaining moves are discarded.

5. Design-time phase, sensor placement

The method is an off-line phase that consists of four steps: A, B, C and D. As a result we define both the reduced order thermal model and the sensor placement for the 3D-MPSoC system. The block diagram of this phase is in the top half of Fig. 2.

5.1. 3D-MPSoC model conversion: from structure-centric to energy-centric

First, an accurate 3D-MPSoC thermal model is created according to the model presented in previous section. This will determine matrices A and B according to Eq. (1). Locations that the policy needs to monitor to ensure safe working conditions are determined by the following relation:

$$\tilde{\mathbf{t}}_\tau = \mathbf{C}\mathbf{t}_\tau \tag{5}$$

Eq. (5) describes the choice of relevant locations to monitor inside the MPSoC. Matrix $\mathbf{C} \epsilon \Re^{s \times n}$ is a selection matrix. In this model we assume that we want to control locations on the silicon layer of each tier. We do this to ensure a full MPSoC temperature control in every location containing an active device on the silicon layer.

We assume that s is the total number of those locations. Namely $c_{i,j}$ is equal to 1 if thermal sensor i is located inside the cell j.

In the case study described in the experimental setup section, the number of states that is also the number of temperature values for each cell composing the 3D-MPSoC floorplan is 200. This means that 100 are the number of cells composing the silicon layers and 100 the ones composing the copper ones. This means that matrix A is composed by 10^4 entries and matrix C has 100 rows. These numbers are large if the model is used in a predictive horizon control policy [28].

To determine the states with negligible contribution to the input–output response, the system is balanced using a Gramian-based balancing of state-space realizations [45]. This technique computes a balanced state-space realization for the stable portion of the system. For stable systems, the output is an equivalent system for which the controllability and observability Gramians are equal and diagonal, their diagonal entries forming the vector $\mathbf{g}\epsilon\Re^n$ of Hankel singular values. These values provide a measure of energy for each state in the system. If the corresponding Hankel singular value for a certain state is a relatively small number, this means that state has a small influence in the dynamic of the system. The second output of the Gramian-based balancing [45] is the balancing state transformation matrix $\mathbf{T}\epsilon\Re^{n\times n}$ that converts the original system into the balanced one.

The rationale behind this operation is to change the 3D-MPSoC thermal model system perspective. The original model belongs to a geometric and physical view of the 3D-MPSoC where states are related to physical properties. The new model generated by the Gramian-based transformation is energy centric and every states is a heat propagation dynamic. This representation emphasizes how much a dynamic is relevant to the heat propagation response of the system. The ith row of the conversion matrix $\mathbf{T}$ describes the contribution that the temperature of each thermal cell in the original model gives to the i^{th} most important (in terms of energy) thermal dynamic of the new generated system.

Concluding, the balancing technique is used for sensors placement in the following way. The main outputs of this operation are: the conversion matrix $\mathbf{T}$ and the vector $\mathbf{g}$. Both these outputs are used in the proposed sensor placement technique. Vector $\mathbf{g}$ is used for the identification of relevant states described in Section 5B. In Section 5C, matrix $\mathbf{T}$ is used to find the sensor location that contributes the most to the thermal dynamic of each state. Section 5D places sensors using the just derived sensor locations until the thermal system is observable. This means that we can derive the overall thermal profile of the MPSoC according to data received from just placed thermal sensors.

5.2. Identification of relevant states

In this section we elaborate the information related to the analysis of the Hankel singular values vector $\mathbf{g}$. Fig. 3 shows the state energy distribution for our case study. As Fig. 3 shows, the energy magnitude drops quite fast and most of the states gives almost negligible contributions to the input–output response of the system. To define a threshold level to distinguish between relevant and not relevant states, we look at the rate of decay of the states energy.

Fig. 3 shows the decay rate for the normalized energy related to Hankel singular values for our case study. Red arrows points to change in the rate of the decay rate. To identify transition points we look at peaks in the third derivative of the function defined by vector $\mathbf{g}$. In Fig. 3 they are highlighted with red circles. All these points represent a set of possible threshold point to distinguish between relevant states and negligible states. They are indeed points in which the decay rate changes the way it decades. This means that by adding points after these transition points the advantage of adding states would be smaller in terms of reducing the approximation error.

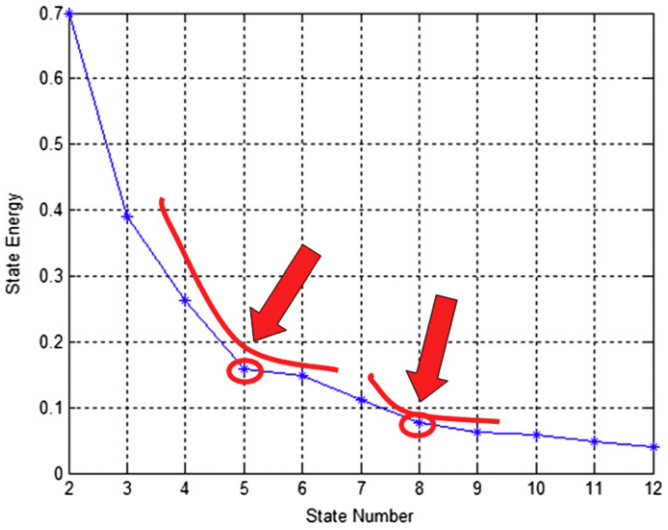


Fig. 3. Decay rate analysis for the normalized energy related to Hankel singular values for our case study. Red arrows points to change in the decay rate. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

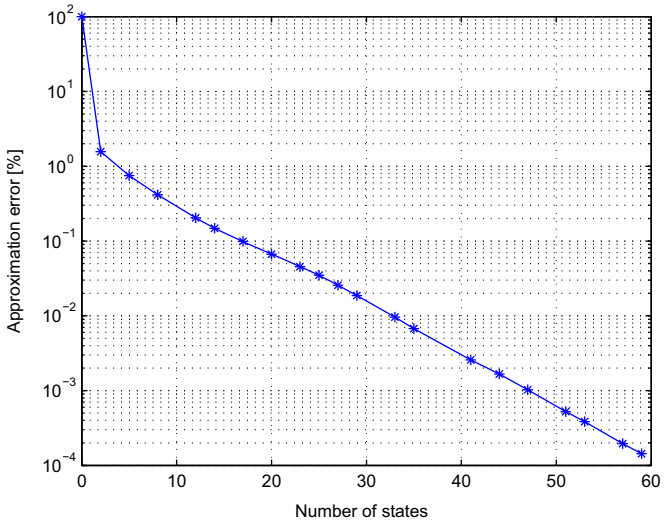


Fig. 4. Approximation error (%) of the reduced order model compared to the full model versus the number of states in the reduced model for our case study.

Fig. 4 shows the model approximation error in percentage versus number of states in the reduced model for our case study. As Fig. 4 shows that the decay rate goes pretty fast. It is important to notice that only threshold points have been considered in this plot. They are marked with '*'. Results show that an approximation error of the reduced order model compared to the full model of 6×10^{-2} in percentage can be achieved with only 20 states and 180 can be easily discarded with a reduction factor of $10\times$ for the given accuracy. It is also important to notice that it does not make much sense to go for higher accuracies because inaccuracies in the silicon, in the power model or in thermal sensors will add uncertainty in the results.

5.3. Balanced state transformation analysis

Here we elaborate information related to conversion matrix $\mathbf{T}$. The ith row of $\mathbf{T}$ describes the contribution that the temperature

of each thermal cell in the original model gives to the *i*th most important (in terms of energy) thermal dynamic of the new generated system. For this reason at this stage for each row *i* of **T**, we identify the most relevant component in absolute value. We call this component *j*. This means that if we place a sensor in the *j*th cell in the original model, among all the possible sensor locations, this position would be the one that will contribute more in terms of energy to the *i*th most important thermal dynamic of the new generated system.

5.4. Reduced order model and sensor placement

At this stage the user-defined parameter that is missing to complete the sensor placement is the desired accuracy of the reduced order model. If we accept an approximation error between the full model and the reduced order model of 6×10^{-2} in percentage, we fix the number of states to 20. By doing this operation we reduce by a factor of 10 the number of states in the model and so the computational complexity of Eq. (1).

At this point a new reduced order model is obtained from the original one after the balancing using a Gramian-based balancing of state-space realizations [45]. States corresponding to Hankel singular values smaller than a pre-defined threshold (in our case we selected the 20th) are discarded. Thus the full MPSoC thermal model is now described by the following system of equations:

$$\mathbf{x}_{\tau+1}=\tilde{\mathbf{A}}\mathbf{x}_{\tau}+\tilde{\mathbf{B}}\mathbf{p}_{\tau} \tag{6}$$

$$\tilde{\mathbf{t}_{\tau}}=\tilde{\mathbf{C}}\mathbf{x}_{\tau} \tag{7}$$

where matrix $\tilde{\mathbf{A}}\epsilon\Re^{l\times l}$ and matrix $\tilde{\mathbf{B}}\epsilon\Re^{l\times p}$. The number of states of the new thermal model is l and p is the number of inputs in the MPSoC model. Eq. (6) describes the state update for the reduced order model of the MPSoC. This equation is analogous to Eq. (1). The only difference is that, in this case, the states do not represent directly temperature values inside each cell. Matrix $\tilde{\mathbf{C}}\epsilon\Re^{s\times l}$ in Eq. (7) relates the value of the states to temperature in s specific locations (every cell in all silicon layers) inside the MPSoC. This equation is analogous to Eq. (5) and describes how the temperature measurements can be derived from the state vector **x**. Thus we need Eq. (7) and matrix $\tilde{\mathbf{C}}$ to extract the temperature vector $\tilde{\mathbf{t}_{\tau}}$ from vector $\mathbf{x}_{\tau}$. If we are interested in recovering the complete MPSoC thermal profile, in Eq. (5), matrix **C** is an identity matrix. Matrix $\tilde{\mathbf{C}}$ is computed in the model order reduction process by Eqs. (1) and (5) as well as matrices $\tilde{\mathbf{A}}$ and $\tilde{\mathbf{B}}$ in Eq. (6). In our case study $s=100$ because we are interested in knowing the temperatures of all the cells of all the silicon layers.

The purpose of sensor placement is to get reliable information on the 3D-MPSoC thermal profile. The reason is because this way, every time the policy is applied, it operates on reliable thermal profile temperature values. The key for this is to obtain the state vector **x**. In step 1, the balancing state transformation matrix **T** converts the original system into the balanced one. Thanks to this matrix, to obtain the estimate of the reduced state vector **x**, it is sufficient to multiply the thermal profile by matrix **T**.

For the system identified by Eq. (1), it means that we are able to reconstruct completely the thermal profile of the chip given the inputs only by looking at the measurements coming from the sensors, placed in locations specified by the matrix C'.

$$\tilde{\mathbf{t}_{\tau}}=\mathbf{C}'\mathbf{t}_{\tau} \tag{8}$$

The number of states in the new thermal model equals to l. Matrix $\mathbf{C}'\epsilon\Re^{s'\times l}$ in Eq. (8) is a selection matrix that describes the sensor placement inside the 3D-MPSoC. This means that we are assuming to have in the output vector s' distinct temperature measurements coming from s' distinct cells every T_s seconds where T_s is the sensors sampling period. The rank of the observability matrix

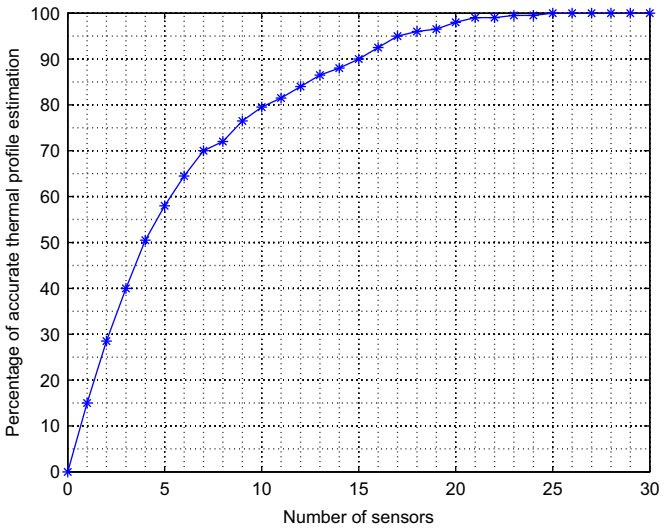


Fig. 5. Sensor placement algorithm: percentage of accurate temperature estimation according to the number of sensors placed with the proposed methodology.

Q expresses the number of states that can be reconstructed from the measurement vector $\tilde{\mathbf{t}_{\tau}}$. The observability matrix Q is expressed by the following equation (see [46]):

$$Q=[C';C'A;\ldots;C'A^{n-1}] \tag{9}$$

If the rank of Q equals n, the state vector x can be reconstructed completely from the measurement vector and the input vector. Then, A is the matrix describing the original system in Eq. (1).

The problem of selecting the right placement of thermal sensors to both minimize the number of sensors and maximize observability is the problem of choosing the matrix $\mathbf{C}'$ with the minimum number of rows that makes the rank of the observability matrix Q equal n. Given a determined MPSoC model, this problem depends on the location and the number of sensors inside floorplan (matrix $\mathbf{C}'$) and the sensor sampling period T_s.

To choose the sensor placement we used the information we got from step 3 about the locations that contribute most to each of the states in the balanced model. The proposed algorithm is a greedy technique that adds a sensor according to the placement suggested in step 3. The algorithm starts from the most relevant state (state number 1) and goes on adding sensors until the rank of the observability matrix equals the rank of **A**. Fig. 5 shows that this is achieved after 25 steps for our case study. This means that with only 25 sensors it is possible to estimate the thermal profile of the 3D-MPSoC. This means that we got a reduction of a factor 8 in the number of required sensors. Fig. 6 shows the resulting placement assuming a sensors sampling frequency T_s of 1 ms.

6. Run-time phase, thermal management optimization

The method is the run-time phase performing the thermal management optimization on the 3D-MPSoC system. The block diagram of this phase is in the bottom half of Fig. 2. During this phase, the reduced order system state vector **x** is estimated. To recover the thermal profile from thermal sensors measurements any state estimator can be used (e.g., a Kalman Filter [46]). Then, this information is used by the thermal model to perform the optimization on the reduced-order 3D-MPSoC model pre-defined in the design-time phase.

The proposed policy, uses both DVFS and variable-flow liquid cooling to meet the desired requirements. Requirements are

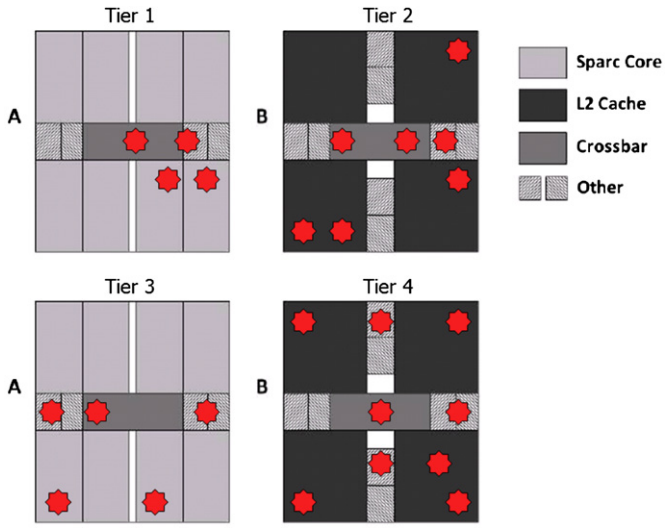


Fig. 6. Sensor placement for our case study with sensors (marked as red stars on the floorplan) sampling frequency T_s of 1 ms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

expressed by an objective cost function. This function consists of two terms. The first one is related to power minimization (3D-MPSoC power consumption and liquid cooling pumping system power consumption) and the second one to the performance loss (undone work). The solution of following minimization is the 3D MPSoC frequencies and cooling pumps speeds necessary to meet the desires requirements. The control problem is as follows:

$$J = \sum_{\tau=1}^{h} (\|\mathbf{Rp}_\tau\|_j + \|\mathbf{Vu}_\tau\|_b) \tag{10}$$

$$min\ J \tag{11}$$

$$subject\ to: \quad \mathbf{f_{min}} \preceq \mathbf{f}_\tau \preceq \mathbf{f_{max}}\ \forall\tau \tag{12}$$

$$\mathbf{x}_{\tau+1} = \tilde{\mathbf{A}}\mathbf{x}_\tau + \tilde{\mathbf{B}}\mathbf{p}_\tau\ \forall\tau \tag{13}$$

$$\tilde{\mathbf{C}}\mathbf{x}_{\tau+1} \preceq \mathbf{t_{max}}\ \forall\tau \tag{14}$$

$$\mathbf{u}_\tau \succeq 0\ \forall\tau \tag{15}$$

$$\mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau\ \forall\tau \tag{16}$$

$$\mathbf{l}_\tau \succeq \mu\mathbf{f}_\tau^{\alpha}\ \forall\tau \tag{17}$$

$$-\mathbf{w} \preceq \mathbf{m}_{\tau+1} - \mathbf{m}_\tau \preceq \mathbf{w}\ \forall\tau \tag{18}$$

$$0 \preceq \mathbf{m}_\tau \preceq \mathbf{1}\ \forall\tau \tag{19}$$

$$\mathbf{p}_\tau = [\mathbf{l}_\tau; \mathbf{m}_\tau]\ \forall\tau \tag{20}$$

It is important to highlight that the matrices $\tilde{\mathbf{A}}$, $\tilde{\mathbf{B}}$ used in previous equations are constant during the h time steps the system tries to minimize the cost function J, and are then updated every time the policy is applied. The time horizon of this predictive policy is defined as h [20].

Function J is expressed by a sum where the summation index τ ranges from 1 to h. The first term $\|\mathbf{Rp}_\tau\|_j$ is the j norm (in our implementation $j=1$) of the power input vector p weighted by matrix $\mathbf{R}$. Power consumption is generated here by two main sources: the voltage-frequency setting of the 3D-MPSoC and the liquid cooling pumping power. Vector p is a vector containing normalized power consumption data of both the cores and the cooling pumps. Matrix $\mathbf{R}$ contains the maximum value of the power consumption of both the cores (first p diagonal entries) and the cooling pumps (last z diagonal entries).

The second term $\|\mathbf{Vu}_\tau\|_b$ is the b norm (in our implementation $b=1$) of the amount of predicted required workload that has not been executed. The weight matrix $\mathbf{V}$ quantifies the importance that executing the workload required from the scheduler has in the optimization process.

Inequality (12) defines the range of working frequencies that can be used. It enables a continuous range of frequency settings but this does not prevent from adding in the optimization problem a limitation on the number of allowed frequency values. Eq. (13) defines the evolution of the system according to the present state and inputs. Eq. (14) states that temperature constraints should be respected at all times and in all specified locations. Since the system cannot execute jobs that have not arrived, every entry of $\mathbf{u}_\tau$ has to be greater than or equal to 0 as stated by Eq. (15). The undone work at time τ, u_τ is defined by Eq. (16). Eq. (17) defines the relation between the power vector $\mathbf{l}$ and the working frequencies. μ is a technology-dependent constant.

Eq. (20) defines formally the structure of vector $\mathbf{p}$ as described in Section 3.1. Vector $\mathbf{l}\epsilon\Re^p$ is the power input vector, where p is the number of frequency islands composing the 3D-MPSoC. Vector $\mathbf{m}\epsilon\Re^z$ contains the normalized amount of cooling power for each of the z independent pumps. Eqs. (18) and (19) define constraints on the liquid cooling management. Eq. (19) states that $\mathbf{m}$ is a normalized value and it can range from 0 to 1. Eq. (18) defines the maximum increment/decrement that the normalized pump can have between two consequent applications of the policy. In other terms this value takes into account the mechanical time dynamics of the pump. Their values are stored in vector $\mathbf{w}\epsilon\Re^z$.

The control problem is formulated over an interval of h time steps, which starts at current time τ. For this reason, the approach is said to be predictive. The result of the optimization is an optimal sequence of future control moves (i.e., frequency settings for the cores of the 3D-MPSoC which are stored in vector $\mathbf{f}$). Only the first sample of such a sequence is actually applied to the process; the remaining moves are discarded.

At the next time step, a new optimal control problem based on new temperature measurements and required frequencies is solved over a shifted prediction horizon. Such a "receding-horizon" [20] mechanism represents a way of transforming an open-loop design methodology into a feedback one, as at every time step the input applied to the process depends on the most recent measurements. To increase the performance of our proposed policy, history information about the task arrival process are exploited by the proposed algorithm. These data are used to make prediction on future workload requirements.

7. Experimental setup

7.1. 3D-MPSoC model

The 3D-MPSoC architecture we are considering is presented in Fig. 1(a)–(c). The floorplan has been modeled using technological parameters and coefficients taken from [7] and [14]. This architecture has a maximum operating frequency of 1.2 GHz and the maximum power consumption of each of the eight processing cores at this frequency is 5 W.

To implement the voltage and frequency scaling techniques, we use frequencies ranging from $\mathbf{f_{min}}$ to 1.2 GHz, see [14] for details. In this range, only specific values of frequencies are allowed. These values are generated from the integer division of the maximum clock frequency by scaling factors as proposed in [30].

We compute the leakage power of processing cores as a function of their area and the temperature. We assume a base leakage power density of 0.25 W/mm^2 at 383 K for 90 nm, as in [37]. T_o accounts for the temperature effects on leakage power and we use the model proposed in [7]. In this case, the leakage power at a temperature T_o K is given by: $P(T) = Po \cdot e^{\beta(T-383)}$, where P_o is the leakage power at 383 K, and β is a technology dependent coefficient. Finally we set $\beta = 0.017$ [29].

7.2. Cooling model

The number of independent pumps is 4 and the spacing between two microchannels on the same layer is 100 μm. We assume that a pump connected to all microchannels of the same layer, such as a centrifugal pump EMB MHIE [34], is responsible for the fluid injection to the whole system. This pump has the capability of producing large discharge rates at small pressure heads.

Liquid is injected to the stacks from this pump via a pumping network. To enable using different flow rates for each stack, the cooling infrastructure includes valves in the network. We assume *normally closed valves* (NCV) provided by Festo group [35]. NCVs use external power to reduce the pressure drop and to increase the flow rate. Cooling microchannels parameters and cooling pump power consumption values are taken from [29].

7.3. Virtual platform environment

The 3D-MPSoC simulation framework is a SystemC-based simulation platform. The main device consists of 16 (8 per tier) 32-bit cores, 16 private memories included in the cores and 16 L2 shared cache memories distributed in four layers of a 3D stuck (as in Fig. 1). All these units communicate among each other by a crossbar interconnect. A floating point unit is also connected to it. The virtual platform environment provides also power statistics for the several hardware modules in the simulated platform. The simulation is based on applications generating functional data traffic on the target architecture. Power consumption data are coming from the 3D simulation platform while temperature data are extracted using the publicly available 3D-ICE thermal tool [18], as described in the previous sections. Modern OSes have a multi-queue structure, where each CPU core is associated with a dispatch queue, and the job scheduler allocates the jobs to the cores according to the current policy. In our simulator, we implement a similar infrastructure, where the queues maintain the threads allocated to cores and execute them.

We use workload traces collected from real applications running on an UltraSPARC T1. We record the utilization percentage for each hardware thread at every second using *mpstat* for several minutes for each benchmark. We use various real-life benchmarks including web server, database management, and multimedia processing. The web server workload is generated by SLAMD [36] with 20 and 40 threads per client to achieve medium and high utilization, respectively. For database applications, we experiment with MySQL using sysbench for a table with 1 million rows and 100 threads. Finally, we run several instances of the mplayer (integer) benchmark as typical examples of multimedia processing. The utilization ratios are averaged over all cores throughout the execution.

7.4. Policy setup

According to the general model of Eqs. (10)–(17), the problem formulation is the following. Matrix **T** is set to be an identity matrix while matrix **R** contains the maximum value of the power consumption of both the cores and the cooling pumps, which are extracted from [7]. In this policy we want to minimize the sum of all contributions to the 3D-MPSoC power consumption as well as the undone workload. For this reason, we set both the norms b and j to 1.

All the others constraints expressed by Eqs. (12)–(17) are considered inside the problem formulation. The policy is applied every $T_{pol} = 10$ ms, while the simulation step for the discrete time integration of the RC thermal model has been set to 200 μs. The sensors sampling period T_s has been set equal to 1 ms. The maximum temperature limit is set to 370 K. The room temperature and t_{fluid} are set to 300 K. In the problem formulation, we used $\alpha = 2$ (as in [33]) to establish the relation between the frequency setting and the power consumption. The linear predictor has been designed using a 3rd order polynomial equation, an observation window of 600 ms and a prediction length equal to 50 ms in the future.

The optimization process is done online using the convex solver proposed in [25]. These operations, have been performed on standard processors (i.e., Core Duo @ 2 GHz) in few tenth of microseconds. For more details on the complexity of the solving algorithm, see [38]. This time is 3 orders of magnitude smaller compared with the time the policy is applied (i.e.10 ms). The time constants needed by the mechanical dynamics of the cooling pumps to go from 0 to maximum power is set to 400 ms.

8. Run-time simulation results

In our experiments, we compare the proposed 3D thermal management method with state of the art thermal management techniques based on DVFS, load balancing and variable flow liquid cooling [7,10,22,23,29]).

Dynamic load balancing (**LB**) [22] balances the workload by moving threads from a core's queue to another if the difference in queue lengths is over the defined threshold. Temperature-triggered task migration (**TTTM**) [23] moves tasks from a core if that core exceeds the threshold temperature. TTTM has an impact on performance resulting from the time overhead required to move tasks between the cores (e.g., context switch overhead and cold start effects). In this work we assume a 1 ms overhead when a thread is migrated to a new core [7,10]. For previously mentioned polices, if the temperature goes higher than 420 K, the system shuts down until the maximum MPSoC temperature returns below 250 K. In temperature triggered DVFS (**TTDVFS**) [22] the voltage and frequency settings are reduced to 10% of the maximum value when the maximum MPSoC temperature exceed the threshold value set to 370 K. TTTM and TTDVFS can also be combined into a joint policy called (**TTTM_TTDVFS**) [10].

We experiment with both air-cooled (**AC**) and liquid-cooled (**LC**) systems for comparison purposes. In **LC_LB**, we apply 100% of the maximum flow rate (0.0323 l/min per cavity [29]). We also consider in the comparison state of the art liquid cooling methods recently proposed in [7,29]. These methods employ a variable-flow liquid cooling combined with DVFS. We refer to the first method as **LC_VF** and to the second one as **LC_Fuzzy**.

Thermal impact of all the policies on the system is shown in Fig. 7. This figure compares the percentage of time spent above the threshold temperature (set to 370 K). Thus each bar shows the area distribution of the dimension of the hot-spot as percentage of the overall MPSoC area.

The first four policies are air cooled methods, while the last four are liquid cooled. As Fig. 7 shows, the first ones are not able to avoid hot-spots. **AC_LB** and **AC_TTTM** present hot-spots for up to 67% of the execution time, and in addition to that, these hot-spots affect more than 80% of the total MPSoC area. Methods using temperature-triggered DVFS show a better performance. This is shown for **AC_TTDVFS** and **AC_TTTM_TTDVFS**.

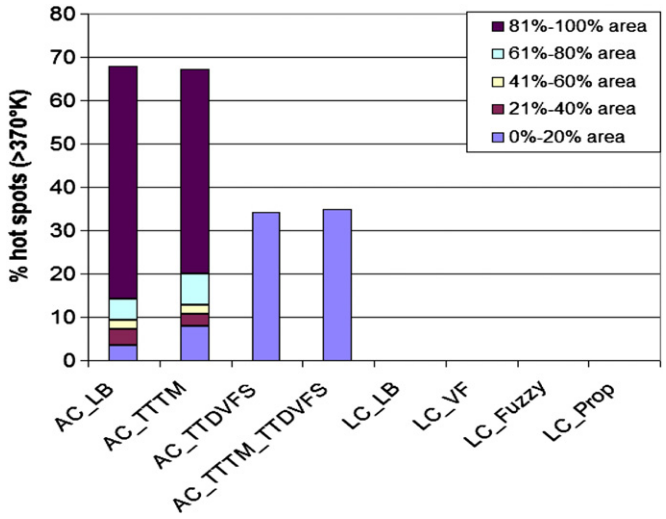


Fig. 7. Percentage of run-time execution where the maximum MPSoC temperature is higher than the threshold (370 K). The area of the hot-spot is also provided as a percentage of the overall MPSoC area.

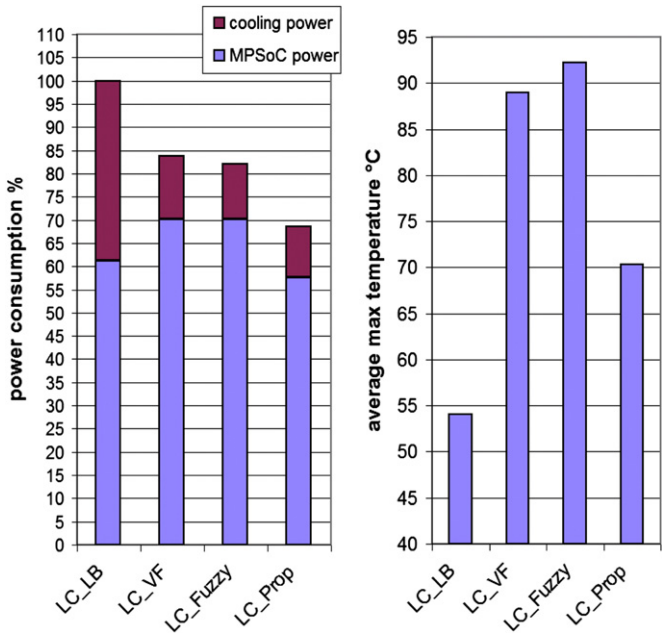


Fig. 8. Left graph: energy consumption of the overall system: 3D MPSoC power consumption and cooling network. Values are normalized to **LC_LB**; right graph: average maximum 3D-MPSoC temperature (°C).

Hence, they present hot-spots for only 34% and 35% of the execution time, respectively. In addition, these hot-spots cover less than 20% of the overall MPSoC area.

Nevertheless, overall air cooled policies do not completely avoid hot-spots. In fact, the 4-tier stacked architecture is unable to dissipate the heat of inner layers by using only a heat spreader. These results indicate the benefits of liquid cooling techniques in completely avoiding any hot-spots scenario. The reason is because of their capability to cool inner layers of the 3D-MPSoC of Fig. 1.

Liquid cooling policies provides a value of undone workload that is less than 1% of the overall executed workload. However, air cooled polices provide values ranging from 24% to 31% in the case of **AC_LB** and **AC_TTDVFS**, respectively.

Since we are interested in techniques that avoid hot-spots while satisfying performance requirements, we restrict from now on our comparison to liquid cooling methods. The following paragraphs compare the proposed policy versus state of the art liquid cooling methods.

The left graph of Fig. 8 shows the overall energy consumption of the 3D-MPSoC. It is divided here into two parts. The first one is the one absorbed by the cooling network (pumps and valves) while the second one is the energy absorbed by the MPSoC activity (switching and leakage). The simplest policy **LC_LB** shows the highest energy consumption. This is because the cooling pumps need to work always at maximum speed to avoid overheating and hot-spots. This causes the cooling power to be extremely high compared with other methods. The value of the cooling power here represents 39% the overall 3D-MPSoC energy consumption. Thus, **LC_VF** [7] and **LC_Fuzzy** [29] have been proposed to reduce the power consumption of the cooling system. We tested these policies on our experimental setup. They show a reduction in the cooling power consumption of 64% and 70%, respectively. The proposed technique has a cooling and an overall 3D-MPSoC power consumption that is respectively 72% and 31% lower compared with **LC_LB**. If we compare our policy with **LC_Fuzzy**, we see a 8% saving in terms of cooling power and a 16% additional saving in the overall MPSoC consumption.

Finally, Fig. 8 shows the average maximum 3D-MPSoC temperature for all the policies under comparison. The lowest thermal profile among the compared policies is generated by the **LC_LB**. In this case the maximum MPSoC temperature has an average value of 54 °C. **LC_LB** and **LC_Fuzzy** show a thermal profile having an average maximum temperature of 89 °C and 92 °C, respectively. The reason is because both these systems save energy by reducing the cooling cost and by having the system working at a temperature close to the threshold set to 97 °C. However, the proposed policy is able to save approximately as much energy as **LC_Fuzzy**, while being able to keep the thermal profile 18 °C lower. The main reason is because the predictive problem formulation of the proposed method is able to satisfy performance requirements by acting in advance and this allows the policy a smoother control on the system and save active power. Therefore, this keeps the overall 3D-MPSoC thermal profile colder.

9. Conclusion

In this work, we presented a combined sensor placement and convex optimization approach for thermal management in 3D-MPSoC with liquid cooling. This technique finds best locations by analyzing the balanced state-space realization of the 3D-MPSoC system and its Hankel singular values decay rate. The number of states of the reduced order model is fixed according to user designer accuracy requirements, and a specific location is assigned to each sensor. Once sensors are placed, temperature sensing information is then used by the thermal management policy. The thermal management policy uses these information to estimate the 3D-MPSoC thermal profile and uses both DVFS and a variable-flow liquid cooling to meet the desired requirements.

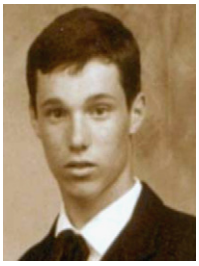
We performed experiments on a model of a 4-tier multicore architecture using benchmarks ranging from web-accessing to playing multimedia. Results show a reduction up to 10× in the number of required sensors. Moreover the proposed sensor location technique has a greedy approach that makes the sensor placement algorithm not computationally intensive. Our experimental results illustrated also that our policy satisfies performance requirements, maintains the temperature below the specified threshold, while reducing cooling energy by up to 72% compared with traditional state of the art liquid cooling techniques. The policy also keeps the thermal profile approximately 18 °C lower compared with state of the art polices using liquid cooling.

Acknowledgments

This research is supported in part by ERC Senior Grant # 246810 and in part by the Nano-Tera.ch RTD Project CMOSAIC (ref. 123618), which is financed by the Swiss Confederation and scientifically evaluated by SNSF.

References

[1] T. Brunschwiler, et al., Validation of the porous-medium approach to model interlayer-cooled 3d-chip stacks, in: 3D-IC, 2009.
[2] M.M. Sabry, et al., Towards thermally-aware design of 3D MPSoCs with inter-tier cooling, in: DATE, 2011.
[3] A.K. Coskun, J. Meng, D. Atienza, M.M. Sabry, Attaining single-chip, high-performance computing through 3D systems with active cooling, IEEE Micro 31 (4) (2011).
[4] M.M. Sabry, et al. Energy-efficient multi-objective thermal control for liquid-cooled 3D stacked architectures, IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems 30 (12) (2011).
[5] L.D. Paulson, IBM supercomputers heat will warm university structures, Computer 42 (9) (2009) 18–21.
[6] F. Alfieri, et al., 3D integrated water cooling of a composite multilayer stack of chips, Journal of Heat Transfer 132 (12) (2010).
[7] A.K. Coskun, et al. Energy-efficient variable-flow liquid cooling in 3D stacked architectures, in: DATE, 2010.
[8] A. Bhunia, et al., High heat flux cooling solutions for thermal management of high power density gallium nitride HEM, in: Inter Society Conference on Thermal Phenomena, 2004.
[9] T. Brunschwiler, et al., Direct liquid-jet impingement cooling with micron-sized nozzle array and distributed return architecture, in: ITHERM, 2006.
[10] A.K. Coskun, et al., Modeling and dynamic management of 3D multicore systems with liquid cooling, in: VLSI-SoC, 2009.
[11] H. Lee, et al., Package embedded heat exchanger for stacked multichip module, transducers, solid-state sensors, Actuators and Microsystems (2003), doi:10.1109/sensor.2003.1216956.
[12] D.B. Tuckerman, et al., High-performance heat sinking for VLSI, IEEE Electron Device Letters (1981), doi:10.1109/edl.1981.25367.
[13] T. Brunschwiler, et al., Interlayer cooling potential in vertically integrated packages, Microsystem Technologies 15 (1) 57–74, doi:10.1007/s00542-008-0690-4.
[14] P. Kongetira, et al., Niagara: a 32-way multithreaded SPARC processor, IEEE Micro (2005), doi:10.1109/MM.2005.35.
[15] S. Borkar, Design challenges of technology scaling, IEEE Micro (1999), doi:10.1109/40.782564.
[16] K. Skadron, et al., Temperature-aware microarchitecture: modeling and implementation, TACO (2004), doi:10.1109/ISCA.2003.1206984.
[17] A.K. Coskun, et al., Modeling and dynamic management of 3D multicore systems with liquid cooling, in: VLSISOC, 2009.
[18] A. Sridhar, et al., 3D-ICE: fast compact transient thermal modeling for 3D-ICs with inter-tier liquid cooling, in: ICCAD, 2010.
[19] R. Mukherjee, et al., Physical aware frequency selection for dynamic thermal management in multi-core systems, in: ICCAD, 2006.
[20] A. Bemporad, et al., The explicit linear quadratic regulator for constrained systems, Automatica 38 (1) (2002) 320.
[21] F. Zanini, et al. Multicore thermal management with model predictive control, in: ECCTD, 2009.
[22] J. Donald, et al., Techniques for multi-core thermal management: classification and new exploration, in: ISCA, 2006.
[23] A.K. Coskun, et al., Temperature management in multiprocessor SoCs using online learning, in: DAC, 2008.
[24] S. Boyd, et al., Convex Optimization, Cambridge University Press, 2004.
[25] M. Grant, et al., mphCVX: Matlab software for disciplined convex programming, Available at ⟨www.stanford.edu/~boyd/cvx/⟩.
[26] F. Zanini, et al., Online convex optimization-based algorithm for thermal management of MPSoCs, in: GLSVLSI, 2010.
[27] F. Zanini, et al., Temperature sensor placement in thermal management systems for MPSoCs, in: ISCAS, 2010.
[28] F. Zanini, et al., Multicore thermal management using approximate explicit model predictive control, in: ISCAS, 2010.
[29] M.M. Sabry, et al., Fuzzy control for enforcing energy efficiency in high-performance 3D systems, in: ICCAD, 2010.
[30] M. Ruggiero, et al., MPARM: exploring the multi-processor SoC design space with system C, Journal of VLSI Signal Processing 41 (2) (2005), doi:10.1007/s11265-005-6648-1.
[31] M. Magno, et al., Adaptive power control for solar harvesting multimodal wireless smart camera, in: ICDSC, 2009.
[32] Y. Wang, et al., Temperature-constrained power control for chip multiprocessors with online model estimation, in: ISCA, 2009.
[33] S. Murali, et al., Temperature control of high performance multicore platforms using convex optimization, in: DATE, 2008.
[34] WILO MHIE centrifugal pump. ⟨http://www.wilo.com⟩.
[35] Festo technology. ⟨http://www.festo.com⟩.
[36] SLAMD Distributed Load Engine. ⟨www.slamd.com⟩.
[37] P. Bose, Power-efficient microarchitectural choices at the early design stage, in: Keynote Address on PACS, 2003.
[38] J. Mattingley, et al., Code Generation for Receding Horizon Control, IEEE Control Systems Magazine (2011), doi:10.1109/CACSD.2010.5612665.
[39] D. Shin, et al., Energy-optimal dynamic thermal management for green computing, in: ICCAD, 2009.
[40] C. Sumana, et al., Optimal selection of sensors for state estimation in a reactive distillation process, Journal of Process Control (2009), doi:10.1016/j.jprocont.2009.01.003.
[41] S. Joshi, et al., Sensor selection via convex optimization, Transaction on Signal Processing (2009), doi:10.1109/TSP.2008.2007095.
[42] T. Boukhobza, et al., State and input observability recovering by additional sensor implementation: a graph theoretic approach, Automatica (2009), doi:10.1016/j.automatica.2009.03.011.
[43] S.O. Memik, et al., Optimizing thermal sensor allocation for microprocessors, IEEE TCAD (2008), doi:10.1109/TCAD.2008.915538.
[44] S. Sharifi, et al., An analytical model for the upper bound on temperature differences on a chip, in: Proceedings of GLSVLSI, 2008.
[45] A.J. Laub, et al., Computation of system balancing transformations and other applications of simultaneous diagonalization algorithms, IEEE Transactions on Automatic Control, AC-32, (1987), doi:10.1109/TAC.1987.1104549.
[46] G.F. Franklin, et al., Digital Control of Dynamic Systems, 3rd ed., McGraw Hill, December 29, 1997.



Francesco Zanini is currently a Ph.D. student at EPFL. He has 3 masters degrees in Electronic Engineering from the University of Parma, the National University of Ireland and the Advanced Learning and Research Institute. He got the best student award from the faculty of engineering of the university of Parma in 2004. He won for 2 consecutive years the Franchetti Award for excellent school career. His research interests include design methodologies for embedded MPSoC with particular emphasis on thermal management policies and algorithms.

David Atienza (M'05) received his MSc and Ph.D. degrees in Computer Science and Engineering from Complutense University of Madrid (UCM), Spain, and Inter-University Microelectronics Center (IMEC), Belgium, in 2001 and 2005. Currently he is Professor and Director of the Embedded Systems Laboratory (ESL) at Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland, and adjunct professor at the Computer Architecture Department of UCM. He is also scientific counselor of longtime research of IMEC Nederland (IMEC-NL). His research interests focus on design methodologies for high-performance Multi-Processor Systems-on-Chip (MPSoCs) and embedded systems, including new 2D/3D thermal-aware design, wireless sensor networks, dynamic memory optimizations and Network-on-Chip (NoC) design. In these fields, he is co-author of more than 140 publications in prestigious journals and conferences. Dr. Atienza is also Associate Editor of IEEE Transactions on Computer-Aided Design of Circuits and Systems, IEEE Embedded Systems Letters, and Elsevier Integration. He is an elected member of the Executive Committee of the IEEE Council of Electronic Design Automation (CEDA) since 2008, and of the Board of Governors of the IEEE Circuits and Systems Society (CASS).

Giovanni De Micheli (S'79-M'83-SM'89-F'94) is Professor and Director of the Institute of Electrical Engineering and the Integrated Systems Centre at EPF Lausanne, Switzerland. He is program leader of the Nano-Tera.ch program. Previously, he was Professor of Electrical Engineering at Stanford University. His research interests include several aspects of design technologies for integrated circuits and systems, such as synthesis for emerging technologies, networks on chips and 3D integration. He is also interested in heterogeneous platform design including electrical components and biosensors, as well as in data processing of biomedical information. He is author of Synthesis and Optimization of Digital Circuits, McGraw-Hill, 1994, co-author and/or co-editor of eight other books and of over 400 technical articles. Prof. De Micheli is the recipient of the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems. He is a Fellow of ACM and IEEE. He received the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000. He received the 1987 D. Pederson Award for the best paper on the

IEEE Transactions on CAD/ICAS, two Best Paper Awards at the Design Automation Conference, in 1983 and in 1993, and a Best Paper Award at the DATE Conference in 2005. He has been serving IEEE in several capacities, namely: Division 1 Director (2008–9), co-founder and President Elect of the IEEE Council on EDA (2005–7), President of the IEEE CAS Society (2003), Editor in Chief of the IEEE Transactions on CAD/ICAS (1987–2001). He is and has been Chair of several conferences, including DATE (2010), pHealth (2006), VLSI SOC (2006), DAC (2000) and ICCD (1989).