Enhancing Two-Phase Cooling Efficiency through Thermal-Aware Workload Mapping for Power-Hungry Servers

Arman Iranfar, Ali Pahlevan, Marina Zapater, and David Atienza
Embedded Systems Laboratory (ESL), Swiss Federal Institute of Technology Lausanne (EPFL), Switzerland
{arman.iranfar, ali.pahlevan, marina.zapater, david.atienza}@epfl.ch

***Abstract*—The power density and, consequently, power hungriness of server processors is growing by the day. Traditional air cooling systems fail to cope with such high heat densities, whereas single-phase liquid-cooling still requires high mass flow-rate, high pumping power, and large facility size. On the contrary, in a micro-scale gravity-driven thermosyphon attached on top of a processor, the refrigerant, absorbing the heat, turns into a two-phase mixture. The vapor-liquid mixture exchanges heat with a coolant at the condenser side, turns back to liquid state, and descends thanks to gravity, eliminating the need for pumping power. However, similar to other cooling technologies, thermosyphon efficiency can considerably vary with respect to workload performance requirements and thermal profile, in addition to the platform features, such as packaging and die floorplan. In this work, we first address the workload- and platform-aware design of a two-phase thermosyphon. Then, we propose a thermal-aware workload mapping strategy considering the potential and limitations of a two-phase thermosyphon to further minimize hot spots and spatial thermal gradients. Our experiments, performed on an 8-core Intel Xeon E5 CPU reveal, on average, up to $10^{\circ}C$ reduction in thermal hot spots, and 45% reduction in the maximum spatial thermal gradient on the die. Moreover, our design and mapping strategy are able to decrease the chiller cooling power at least by 45%.**

I. Introduction

Nowadays, data centers consume more than 2% of the global energy consumption [1] with cooling energy accounting for about 30% of the share [2], [3]. Power Usage Effectiveness (PUE), defined as the ratio of total facility energy to IT equipment energy, indicates how efficient a data center is. While PUE improved from 2.5 in 2007 to 1.65 in 2013 [4], not major improvements have been recently reported. This is mainly due to inefficiency of air-cooling systems at traditional cooling facilities. A recent study by Cisco shows that through a set of modifications in all its facilities, a PUE drop from 1.48 to 1.36 would save US$2 million/year [5].

Liquid-cooling systems are expected to allow further improvement in PUE. In particular, the framework developed by [6] shows that Direct Contact Liquid Cooling (DCLC) systems can reduce the PUE down to 1.17. In addition, inter-layer liquid cooling in 3D stacked processors [7] is claimed to be very efficient in hot spot reduction. Nonetheless, inter-layer liquid cooling requires major changes in processor design and fabrication, making their solution costly.

One of the most recent technologies proposed for efficiently cooling servers is a micro-scale gravity-driven two-phase thermosyphon [8]. Since a thermosyphon is placed on top of the processor package, in contrast to inter-layer cooling, it does not require any changes in design and fabrication of existing processors. The thermosyphon designed and manufactured in [8] achieves a PUE of 1.05. Thus, such a design, if industrialized in a way that fully exploits all its potential, can save more money than any other cooling systems.

Apart from mechanical hardware design and manufacturing challenges [8], platform- and workload-awareness play a significant role in thermosyphon efficiency for removing high heat fluxes. This, in return, can enhance performance metrics of power-hungry servers. Moreover, once designed and manufactured, a thermosyphon still provides the user with adjustable parameters to tune its performance according to the workload requirements. Although similar profound research [9] is available for more conventional cooling technologies, two-phase micro-scale thermosyphons have not been studied yet. Hence, in this work, after profiling the workload performance requirements and measuring power consumption, we discuss the workload- and platform-aware design of a thermosyphon for power-hungry processors. Afterwards, we propose a thermal-aware workload mapping strategy specifically tailored to the designed thermosyphon to further reduce thermal hot spots and spatial gradients while meeting the workload requirements.

The main contributions of our work are as follows:

- we show the efficiency potential and limitations of a two-phase thermosyphon for power-hungry processors,
- we propose a workload- and platform-aware design and adaptation of a thermosyphon, together with a thread mapping policy for multicore enterprise servers when a two-phase thermosyphon is used,
- We evaluate our design and mapping strategy under different workloads requirements when compared to the state of the art. Our experiments show that thermal hot spots decrease up to $10^{\circ}C$ and the maximum spatial gradients diminish up to 45%,
- Finally, our work reveals at least 45% reduction in the chiller cooling power consumption, when compared to state-of-the-art designs and mapping strategies.

II. Related Work

A. Data center cooling methods

Air cooling systems have been vastly employed at different levels from chips to data centers. At chip level, using fans and heatsinks [10] is the most popular system due to its design simplicity. At rack level, server placement in a rack plays an

This work has been partially supported by the EC H2020 MANGO (GA No. 671668) and RECIPE (GA No. 801137) projects, and the ERC Consolidator Grant COMPUSAPIEN (GA No. 725657)

important role in heat flux [11], whereas at room level airflow configuration is known as the main design parameter [12].

Nonetheless, air cooling approaches are inefficient especially when encountering power-hungry servers and Multi-processor Systems-on-Chip (MPSoCs) [12]. On the contrary, liquid cooling systems benefit from high heat transfer coefficient and are capable of removing high heat flux. In particular, single-phase cooling, in which no phase change occurs, has been used by IBM [13].

Two-phase cooling is another liquid cooling method in which liquid-vapor phase change occurs. The use of two-phase cooling, rather than single-phase liquid-based cooling systems, is strongly motivated due to their reduced mass flow-rates, lower pumping power, and smaller facility size [14], while providing higher heat transfer coefficients and more uniform temperature profiles [12]. Previous thermosyphon prototypes such as [15], however, had a very large footprint area ($1m \times 1m$) making them impractical in commercial servers. Nonetheless, recent work by [16] and [8] led to the design of micro-scale thermosyphons which can be placed directly on top of a CPU. Such a design, hence, necessitates careful evaluation of thermosyphon as a promising cooling device for different types of servers, which has not been performed so far in the literature.

B. Workload-Aware Thermal Management

Workload-aware thermal management for servers and data centers is quite rich and wide in literature. In particular, a thermal-aware job placement strategy is proposed by [17] considering the cooling system impact on data center power consumption. Chan et al. [18] propose a thermal management framework with respect to fan speed impact on performance and its energy cost. TheSPoT [19] proposes thermal-aware workload migration scheme for MPSoCs. Authors in [9] address thermal-aware workload balancing policies for MPSoCs. Finally, authors in [7] address job scheduling in 3D stack architectures to maximize liquid cooling efficiency.

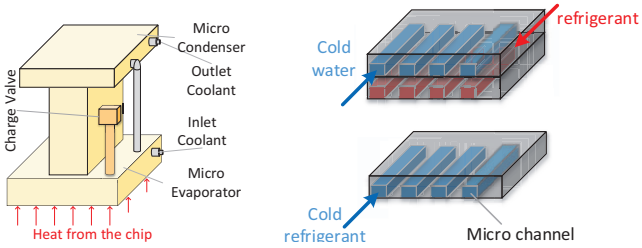
Despite such rich literature, micro-scale two-phase thermosyphon, which is one of the most promising next-generation cooling technologies, able to satisfy performance requirements and thermal constraints, has not been considered so far.

III. Gravity-Driven Two-Phase Thermosyphon

A. Overview

Fig. 1a and 1b show a thermosyphon schematic and refrigerant and coolant loops, respectively. A thermosyphon is composed of three major components, namely, micro-condenser, micro-evaporator, and pipes. The evaporator is located on top of a heat source (in this work, the heat spreader of a CPU), and initially contains a refrigerant in liquid state in its micro-channels. The heat from the CPU increases the evaporator temperature, and the refrigerant partially evaporates. The two-phase mixture composed of vapor and liquid ascends towards the condenser. The heat exchange between the coolant (i.e., cold water) and the hot two-phase mixture makes the two-phase mixture fall through the pipe thanks the gravity.

Considering that the evaporator size scales linearly according to the CPU dimension, the most important design-time



(a) Schematic of a two-phase thermosyphon

(b) Condenser and evaporator loops

Fig. 1. Thermosyphon prototype proposed by [8].

parameters that drive heat flux removal are the filling ratio of the refrigerant, and the refrigerant type. Other important parameters affecting thermosyphon efficiency are the inlet coolant temperature and its flow rate. These parameters can be tuned at runtime according to the CPU workload.

B. Motivational Example

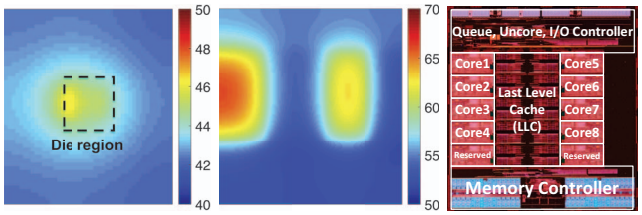
Authors in [8] evaluate the thermosyphon efficiency considering a uniform heat flux over the whole chip. However, this is not a realistic assumption for current applications and CPUs, as different workload mappings lead to non-uniform heat flux, which ultimately cause hot spots and spatial thermal gradients [20]. Moreover, Seuret et. al. [8] assume that the heat flux received by the thermosyphon is the total power generated by the die divided by the surface of the package. This assumption, however, is too simplistic, as the heat flux is larger on the package area right above the die, even in the presence of a heat spreader [20].

In addition, CPU package temperature and die temperature are reciprocally dependent. In fact, hot spots and spatial thermal gradients on the die are scaled-up of those on the package. For instance, a thermal hot spot of $46°C$ and spatial gradient of $0.5°C/mm$ on the package, may lead to, respectively, a hot spot and spatial gradient of $66°C$ and $6.6°C/mm$ on the die, as shown in Fig. 2. In this work, we use 3D-ICE thermal simulator [20], [21] in order to obtain the die temperature. As indicated in the figure, it is of great importance to design a thermosyphon that achieves the most homogeneous thermal profile with the smallest thermal hot spots on the evaporator side. More importantly, Fig. 2b demonstrates that despite its efficiency in removing large heat fluxes, the thermosyphon is not capable of alleviating thermal gradients on the die without an adequate thermal-aware workload mapping strategy.

IV. Overview of The System and Power Model

A. Server CPU Architecture and Floorplan

We consider the Xeon E5 v4 platform [23] for our target workloads. This processor includes an 8-core Broadwell-EP CPU with dual clock frequency domains (Core and Uncore). The memory subsystem comprises L1 instruction and data cache both of 32 KB, a private L2 of 256 KB, and a Last-Level Cache (LLC) of 25 MB. Fig. 2c shows the die shot of the Broadwell CPU in 14nm process. The die area is 246 mm^2 and two cores are reserved for a deca-core CPU chip design.



(a) Package temperature (b) Die temperature (c) Die floorplan of Intel Xeon E5 v4 [22].

	θ_{max} (°C)	θ_{avg}(°C)	$\nabla\theta_{max}$(°C/mm)
Die	66.1	55.9	6.6
Package	46.4	42.9	0.5

(d) Temperature comparison

Fig. 2. Die thermal profile vs. package thermal profile when using thermosyphon with non-optimized design and workload mapping strategy.

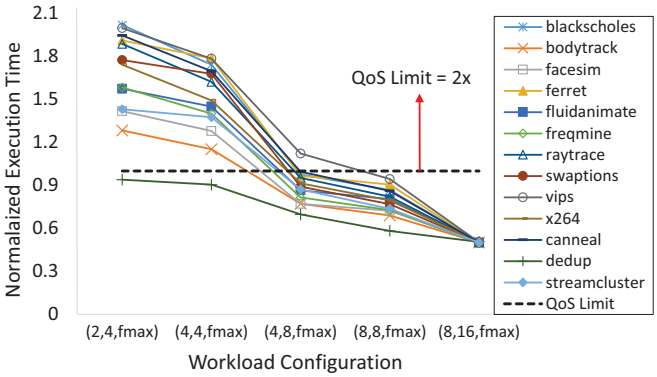


Fig. 3. Execution time normalized to QoS limit for some workload configurations @f_{max}.

B. Workload Configuration and QoS Requirement

We use the PARSEC 3.0 benchmark suite [24] featuring multithreaded workloads. We evaluate the power consumption of PARSEC benchmarks as a function of the assigned number of cores (N_c), threads (N_t), and frequency scaling (f) that provide different thermal maps and profiles on the CPU die. Hence, based on these characteristics, we consider different configurations per benchmark, i.e., (N_c, N_t, f).

QoS constraints are defined in terms of the maximum allowable degradation in workload execution time. In this work, we consider a QoS constraint of 1x, 2x, and 3x degradation [25], w.r.t. a baseline execution through a native 8-core CPU, with 16 threads, at maximum frequency for both core and uncore. Fig. 3 shows the normalized execution time for different configurations and workloads considered in this work, with the QoS constraint set at 2x degradation.

C. Server Processor Power Model

We consider two main contributors to the overall power consumption of the CPU: 1) the core region composed of the cores and L1/L2 caches, and 2) uncore components, which include LLC, memory controller, and IO subsystem. For power measurements, we use the running average power limit (RAPL) interface. We also leverage CPUPOWER and LIKWID [26] utilities to set the core and uncore frequency, respectively.

TABLE I
C-STATES POWER CONSUMPTION OF XEON E5 V4 FOR ALL 8 CORES

	Latency (s)	Power (W) @2.6GHz	Power (W) @2.9GHz	Power (W) @3.2GHz
POLL	0	27	32	40
C1	2	14	15	17
C1E	10	9	9	9

1) Core Power: Current servers can benefit from different idle power states (C-states). Our target Intel processor is equipped with POLL, C1, C3, and C6 states [23]. POLL state represents the default working state of a core without any latency to resume execution. A higher C-state level shows a deeper sleep state with less power consumption but larger resuming latency. TABLE I shows the power measurements of the C-states for our target server.

Finally, for each benchmark, the dynamic power consumption is measured as a function of frequency for different configurations. In our case, to satisfy the defined QoS requirement, we consider three frequency levels: 2.6, 2.9, and 3.2 GHz.

2) Uncore Power: The static and dynamic LLC power model was extracted by measuring for a 25 MB capacity which is 2 W in the worst-case. We also empirically measured the memory controller and IO subsystem power consumption overhead of an Intel Xeon v4 CPU. This power consumption is split in two parts: (i) a constant component which accounts for the static, and (ii) a component proportional to the operating condition and uncore frequency. The constant part constitutes a 9 W overhead in all operating points, while the proportional one provides an 8 W variation from the minimum to maximum uncore frequency (i.e., 1.2 GHz-2.8 GHz).

V. PROBLEM DESCRIPTION

In this work, we address the design of a thermosyphon for CPU packages placed within a server and racks based on the range of workloads. As discussed in Section IV, different workloads require different computational power for a particular performance requirement. This power consumption also varies with the number of threads, cores, and frequency. Since the total package power consumption ranges from 40.5 W to 79.3 W among all configurations and applications, the goal is:

- design a thermosyphon for the worst-case power consumption and hot spot temperature,
- select the best configuration, i.e., frequency, number of cores, and number of threads, to satisfy the required QoS while minimizing power consumption,
- map workloads within a CPU to minimize the number and magnitude of hot spots,
- set the cooling system parameters including water flow rate and water temperature such that all thermosyphons within a rack can operate optimally while the cooling power is minimized.

A thermosyphon is equipped with a flow-meter and valve that allows adjusting the coolant flow rate at runtime (as shown in Fig. 4) to satisfy the thermal constraints. To adjust water temperature, one water cooling system (chiller) per rack is

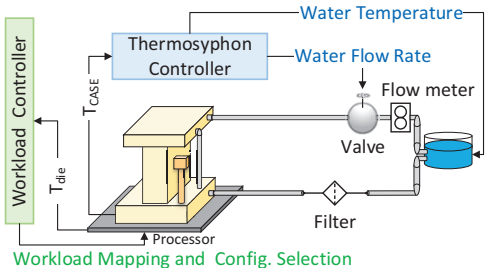
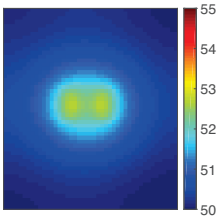
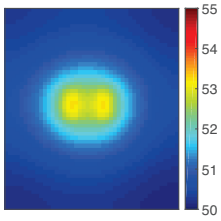


Fig. 4. Proposed approach.



(a) Package thermal map: Design 1



(b) Package thermal map: Design 2

Design		θ_{max} (°C)	θ_{avg} (°C)	$\nabla\theta_{max}$ (°C/mm)
Package	#1	52.7	50.3	0.33
	#2	53.5	50.6	0.43
Die	#1	73.2	62.1	6.8
	#2	79.4	66.2	7.1

(c) Temperature comparison

Fig. 5. Package and die temperature for different orientations of thermosyphon on processor.

used. Therefore, all thermosyphons should work with the same water temperature. This limitation necessitates careful workload allocation to CPUs, as well as thermal-aware workload mapping on the cores to provide balanced temperature for all CPU packages.

VI. Thermosyphon Design Optimization

In this work, we study the thermosyphon design optimization aspects from a system-level perspective including the orientation of inlet-outlet micro-channels on the evaporator, refrigerant type and its filling ratio, and water inlet temperature and its flow rate.

A. Thermosyphon Orientation

The thermosyphon orientation influences the position of evaporator inlet and outlet. Since the CPU die and package are not symmetric, the thermosyphon orientation also affects the number of micro-channels at evaporator side (assuming a constant channel width). Hence, for the same workload, different hot spots can appear depending on different orientations. Fig. 5 shows two different designs of the thermosyphon for our target CPU when all cores are equally loaded. In the first design, the evaporator inlet and outlet are located, on the east and the west, respectively, while in the second design, the evaporator inlet and outlet are located on the north and the south, respectively. Also, Fig. 5 indicates that, for a fully utilized CPU, if less frequent hot spots with lower maximum temperature are required, the first design provides better results. Moreover, although the die is centered in the package, it creates larger hot spots on its left, since, as shown in Fig. 2c, there is a dead area producing no power on the right side of the die. Consequently, due to the fact that evaporator inlet is always cooler than its outlet, an eastward flow of the refrigerant results in more homogeneous thermal profile across the chip, with smaller hot spots. Therefore, we choose the first design for the thermosyphon orientation.

B. Refrigerant and Filling Ratio

Refrigerant physical and thermal properties can considerably affect the thermosyphon efficiency in terms of heat removal. Since the type of refrigerant should be determined at design time, we consider the maximum workload (i.e., the worst case) and the T_{CASE_MAX} ($85°C$), which is the maximum temperature from the center of the heat spreader, as the thermal constraint for our design. Once the refrigerant is known, thermosyphon should be charged at a particular filling ratio, as this changes the thermosyphons' efficiency in heat removal. For our design and target workload, we fill the thermosyphon with R236fa and a filling ratio of 55%.

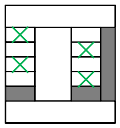
C. Water Temperature and Flow Rate

While water temperature can be adjusted even at runtime, due to large response time, run-time adaptation of such a parameter may not be practical for workloads with critical deadlines. Water flow rate, however, can be adjusted dynamically due to its faster response time. The water flow rate and inlet water temperature affect the amount of electrical power consumed by the temperature control system. Hence, for an optimized thermosyphon design, the lowest flow rate and the highest inlet water temperature for which T_{CASE} remains below T_{CASE_MAX} for the worst case workload, should be the one to be used. For our design and target workload, we consider a water flow rate of 7 Kg/h at $30°C$.

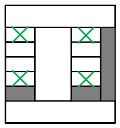
VII. QoS-Aware Configuration Selection and Thermal-Aware Workload Mapping

When meeting the QoS constraint requires the use of all CPU cores, there is no choice left for the workload scheduler to optimize average and maximum temperature. However, as shown in Section IV, depending on the application type and the user-defined QoS requirement, fewer cores than the maximum number of CPU cores ($N_{c,cpu}$) can be used to achieve larger power savings. Thus, when the number of required cores (N_c) is less than $N_{c,cpu}$, workload threads can be mapped optimally onto the cores to minimize average temperature and/or number and values of hot spots according to the thermosyphon behavior.

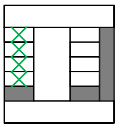
Fig. 6a-c shows three completely different workload mappings using four cores of an 8-core processor. In particular, we consider two different scenarios. In the first scenario, we assume that all idle cores are set to POLL state, while in the second scenario, we consider C1 state. Fig. 6d shows the corresponding hot spot, the maximum spatial gradient, and average temperature of the die for these mappings.



(a) Scenario #1



(b) Scenario #2



(c) Scenario #3

Die Temperature	POLL			C1		
	1	2	3	1	2	3
θ_{max} (°C)	68.2	65.0	77.6	57.1	64.2	73.3
θ_{avg} (°C)	55.8	54.5	62.0	52.1	53.7	60.5
$\nabla\theta_{max}$ (°C/mm)	1.8	2.0	6.5	1.5	2.2	6.8

(d) Comparison

Fig. 6. Die thermal profile vs. package thermal profile when using thermosyphon with non-optimized design and workload mapping strategy.

When the CPU is in POLL state, the static power of idle cores is comparable to the dynamic power consumption of active ones. In this case, scenario #2, which is a conventional thermal-aware workload balancing strategy (i.e., loading the CPU with the same workload starting from the corners), results in lower hot spots and average die temperature than the other two scenarios. While scenario #3 leads to the highest hot spot and average temperature, scenario #1 attains higher temperature but close to that of scenario #2. The reason lies in the fact that high power density between the active cores, does not let the cores exchange heat properly.

When deeper C-states, such as C1, are used, scenario #1 is a better choice since there is not more than one hot spot (active core) on the same horizontal line. Therefore, evaporator micro-channels are more efficient in removing heat from the chip. Nonetheless, this is not the case when idle cores are set at POLL state, because idle cores still consume large amount of static power. Therefore, depending on the C-states used for idle cores (determined by the maximum latency tolerable for the application) different optimal mappings can be attained to alleviate thermal hot spots. The same discussion is also valid when 5 cores are used. Nonetheless, when using more than 5 cores, a more straightforward approach should be adopted. In this case, threads should be mapped to the cores starting from the corners, then mapped to the rest recalling that always fewer active cores on the same horizontal line are desirable.

Following the discussion above, an optimal mapping that minimizes the number and magnitude of hot spots can be obtained for each particular configuration (number of cores to be used). Algorithm 1 presents the proposed configuration selection and workload mapping. We consider a set of applications $\boldsymbol{A}$ whose threads should be mapped to the multicore CPU with $N_{c,cpu}$ cores. Each application requires a minimum QoS q_i and can tolerate up to d_i seconds delay for idle cores on the CPU. The goal is to find the number of cores (N_c), threads per core (N_t), and frequency (f) for which the power consumption is minimized and q_i is satisfied. The power consumption and the QoS resulting from each configuration j are known and stored in $\boldsymbol{P_i}$ and $\boldsymbol{Q_i}$ vectors, respectively, by $\mathbf{P}(N_c^j, N_t^k, f^l)$ and $\mathbf{Q}(N_c^j, N_t^k, f^l)$ obtained from profiling the application. We sort $\boldsymbol{P_i}$ in an ascending order and we select the first configuration for which $\boldsymbol{Q_i}$ is larger than q_i. Finally, knowing the area ($\boldsymbol{S}$) and the power consumption of each component, the heat generated by different components is estimated. Afterwards, based on the delay that each application can tolerate and the per-component estimated heat flux ($\boldsymbol{H_i}$), we follow the mapping strategy discussed earlier to minimize the value and number of hot spots. Finally, during runtime, we increase water flow rate only if a thermal emergency (i.e., $T_{CASE} \geq T_{CASE\_MAX}$) occurs and lowering the frequency violates QoS requirement.

Algorithm 1: Configuration selection and thread mapping

Input : $\boldsymbol{A} = \{A_1, \ldots, A_n\}$, $\boldsymbol{D} = \{d_1, \ldots, d_n\}$, $N_{c,cpu}$, $\boldsymbol{N_c} = \{1, \ldots, N_{c,cpu}\}$, $\boldsymbol{N_t} = \{1, 2\}$, $\boldsymbol{f} = \{f_{min}, \ldots, f_{max}\}$, $\boldsymbol{S} = \{s_{core}, s_{llc}, s_{uncore}, s_{memcnt}\}$, $\boldsymbol{QoS} = \{q_1, \ldots, q_n\}$

Output: $CPU_i \xleftarrow{map} A_i@C^{opt}(N_c, N_t, f)$; // Mapping A_i with optimal configuration to server i

forall $i \in \boldsymbol{A}$ **do**
  forall $j \in \boldsymbol{N_c}$, $k \in \boldsymbol{N_t}$, $l \in \boldsymbol{f}$ **do**
    $\boldsymbol{P_i} \leftarrow \mathbf{P}(\boldsymbol{N_c^j}, \boldsymbol{N_t^k}, \boldsymbol{f^l})$
    $\boldsymbol{Q_i} \leftarrow \mathbf{Q}(N_c^j, N_t^k, f^l)$
  $\boldsymbol{P_{sort}} \leftarrow \mathbf{Sort}^{asc}(\boldsymbol{P_i})$
  $C^{opt} \leftarrow$ Find the first configuration in $\boldsymbol{P_{sort}}$ $s.t.$ $\boldsymbol{Q_i} > q_i$
  $\boldsymbol{H_i} \leftarrow \mathbf{H}(\boldsymbol{P_{sort}^{opt}}, S)$
  $CPU_i \leftarrow \mathbf{Map}(\boldsymbol{H_i}, d_i, A_i@C^{opt}(N_c, N_t, f))$

VIII. Experimental Results and Discussion

We use the simulation framework of [8]. In order to provide a fair comparison, we compare our thermosyphon design and thermal-aware workload mapping with the design of [8] equipped with a configuration selection strategy [27] and two different workload mapping policies: [9] and [7]. The latter, is aimed at inter-layer liquid-cooled MPSoCs.

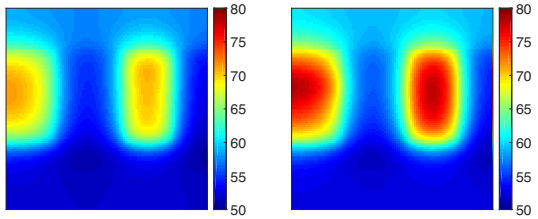
A. Thermal Hot spots and Spatial Gradients

TABLE II shows the thermal hot spots and spatial gradients, on average, achieved by our proposed approach against the state of the art, for different QoS requirements. When no QoS degradation is allowed, all approaches run the workload with f_{max} and maximum number of available cores and threads. Consequently, the improvement in hot spot and spatial gradient reduction comes only from the thermosyphon design. This comparison, highlights the importance of a workload-aware thermosyphon design. In particular, when 2x or 3x degradation from the reference QoS is allowed, our proposed thermal-aware workload mapping outperforms that of [9] and further diminishes thermal hot spots and spatial gradients. The main improvement is obtained if 3x degradation is allowed, as our workload mapping strategy is able to map the workload based on the C-states for idle cores.

The key idea of the workload scheduling policy proposed by [7] is to map the workload first to the cores closer to the liquid inlet. However, such a policy is not suitable when a two-phase thermosyphon is used. Although our discussion

TABLE II
THERMAL HOT SPOT AND SPATIAL GRADIENTS FOR DIFFERENT QOS REQUIREMENTS

Approach	QoS	Die		Package	
		θ_{max}	$\nabla\theta_{max}$	θ_{max}	$\nabla\theta_{max}$
Proposed	*1x*	78.3	0.90	52.1	0.24
	2x	72.2	1.03	49.0	0.24
	3x	68.4	1.25	46.3	0.28
[8]+ [27]+ [9]	*1x*	83.0	0.95	52.5	0.27
	2x	79.5	1.33	51.4	0.30
	3x	77.8	1.60	49.1	0.36
[8]+ [27]+ [7]	*1x*	83.0	0.95	52.5	0.27
	2x	80.5	1.8	50.4	0.32
	3x	79.1	2.3	49.1	0.43



(a) (b)

Fig. 7. Thermal map of the die obtained from a) proposed approach b) state of the art.

in Section VI-A reveals the importance of inlet and outlet location w.r.t. the die floorplan, since the die and micro-channels are separated by means of the package and the heat spreader, the amount of heat removal at the inlet compared to the outlet does not motivate mapping the workload starting from those closer to the inlet. In particular, the 3^{rd} scenario of Fig. 6 clearly discourages such a mapping strategy. Hence, [7], on average, provides the worst results.

Finally, Fig. 7 depicts the die thermal map obtained through our design and workload mapping policy versus state-of-the-art approaches. This figure shows one sample thermal map obtained under 2x QoS degradation. While this hot spot is $78.2°C$ for the state of the art, this work achieves $71.5°C$.

B. Cooling Power

To achieve the same hot-spot temperature without our proposed thermal-aware workload mapping for the same water flow rate, a water temperature of $20°C$ is required. Moreover, the temperature difference between the inlet ($T_{in,w}$) and outlet ($T_{out,w}$) water for our approach is $6°C$, and $11°C$ without our approach, which increases the chiller burden. This implies that the proposed approach reduces cooling power up to 45%, assuming that the electrical power (W) required to change the temperature of L litre water ΔT K is:

$$P = \dot{V} \times \rho \times C_w \times \Delta T, \quad (1)$$

where $\dot{V}$ is volumetric flow rate (Litre/s), ρ represents density (Kg/Litre), and C_w shows water specific heat ($J.Kg^{-1}.K^{-1}$).

Moreover, this previous discussion is pessimistic as we consider that only the chiller is in charge of cooling down the $T_{out,w}$ and the outside air temperature cannot be used. Hence, in real scenarios, the chiller would need to consume much less power to cool down the water (even close to zero).

IX. CONCLUSION

Our study reveals that micro-scale two-phase cooling is a promising technology that requires customized design w.r.t. the workload and processor floorplan. Furthermore, such a design must be accompanied by adequate thermal-aware workload mapping strategies which take into account its efficiency potential and limitations. Overall, our proposed thermosyphon design and workload mapping strategy showed on average, up to $10°C$ reduction in thermal hot spots, and 45% reduction in the maximum spatial thermal gradient on the die with at least 45% less cooling power consumption for the chiller, compared to state-of-the-art solutions.

REFERENCES

[1] J. Koomey, "Growth in data center electricity use 2005 to 2010," *A report by Analytical Press, at the request of The New York Times*, 2011.
[2] J. Y. Kim *et al.*, "Energy conservation effects of a multi-stage outdoor air enabled cooling system in a data center," *Energy and buildings*, 2017.
[3] A. Pahlevan *et al.*, "Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers," *TCAD*, 2018.
[4] M. Stansberry and J. Kudritzki, "Uptime institute 2012 data center industry survey," *white paper, Uptime Institute*, 2013.
[5] Cisco, "Cisco unified computing system site planning guide: Data center power and cooling," 2017, "https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/unified-computing/white_paper_c11-680202.pdf".
[6] M. A. Kadhim *et al.*, "Performance of a mixed mode air handling unit for direct liquid-cooled servers," in *SEMI-THERM*, 2017.
[7] M. M. Sabry *et al.*, "Energy-efficient multiobjective thermal control for liquid-cooled 3-d stacked architectures," *IEEE TCAD*, 2011.
[8] A. Seuret *et al.*, "Design of a two-phase gravity-driven micro-scale thermosyphon cooling system for high-performance computing data centers," in *ITHERM*, 2018.
[9] A. K. Coskun *et al.*, "Temperature aware task scheduling in mpsocs," in *DATE*, 2007.
[10] Y. Joshi and P. Kumar, *Energy efficient thermal management of data centers*. Springer Science & Business Media, 2012.
[11] N. Rolander *et al.*, "An approach to robust design of turbulent convective systems," *Journal of Mechanical Design*, 2006.
[12] A. H. Khalaj and S. K. Halgamuge, "A review on efficient thermal management of air-and liquid-cooled data centers: From chip to the cooling system," *Elsevier, Applied Energy*, 2017.
[13] S. Zimmermann *et al.*, "Aquasar: A hot water cooled data center with direct energy reuse," *Elsevier, Energy*, 2012.
[14] P. L. Leonard and A. Phillips, "The thermal bus opportunity-a quantum leap in data center cooling potential." *ASHRAE transactions*, 2005.
[15] C. L. Ong *et al.*, "Two-phase mini-thermosyphon for cooling of data-centers: Experiments, modeling and simulations," in *ASME InterPACK*, 2017.
[16] N. Lamaison *et al.*, "Two-phase mini-thermosyphon electronics cooling, Part 4: Application to 2u servers," in *IEEE ITHERM*, 2016.
[17] A. Banerjee *et al.*, "Cooling-aware and thermal-aware workload placement for green hpc data centers," in *Green Computing Conference*, 2010.
[18] C. S. Chan *et al.*, "Fan-speed-aware scheduling of data intensive jobs," in *ISLPED*, 2012.
[19] A. Iranfar *et al.*, "Thespot: Thermal stress-aware power and temperature management for multiprocessor systems-on-chip," *IEEE TCAD*, 2018.
[20] ——, "Thermal characterization of next-generation workloads on heterogeneous mpsocs," in *IEEE SAMOS*, 2017.
[21] A. Sridhar *et al.*, "3d-ice: Fast compact transient thermal modeling for 3d ics with inter-tier liquid cooling," in *ICCD*, 2010.
[22] https://en.wikichip.org/wiki/intel/microarchitectures/broadwell_ (client).
[23] Intel, "Intel xeon processor e5 v4 product family datasheet, volume one: Electrical," https://www.intel.com/content/dam/www/public/us/en/documents/datasheets/xeon-e5-v4-datasheet-vol-1.pdf, 2016.
[24] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, 2011.
[25] C. Delimitrou *et al.*, "Optimizing resource provisioning in shared cloud systems," Stanford University, Tech. Rep., 2014.
[26] http://tiny.cc/LIKWID.
[27] R. Cochran *et al.*, "Pack & cap: adaptive dvfs and thread packing under power caps," in *MICRO*, 2011.