ETH zürich

ETH Library

Cooling-aware node-level task allocation for next-generation green HPC systems

Conference Paper

Author(s):
Beneventi, Francesco; Bartolini, Andrea; Cavazzoni, Carlo; Benini, Luca

Publication date:
2016

Permanent link:
https://doi.org/10.3929/ethz-b-000125003

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
https://doi.org/10.1109/HPCSim.2016.7568402

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

This is the post peer-review accepted manuscript of:

F. Beneventi, A. Bartolini, C. Cavazzoni and L. Benini, "Cooling-aware node-level task allocation for next-generation green HPC systems," 2016 International Conference on High Performance Computing & Simulation (HPCS), Innsbruck, 2016, pp. 690-696. doi: 10.1109/HPCSim.2016.7568402

The published version is available online at: https://ieeexplore.ieee.org/abstract/document/7568402

© 2018 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works

Cooling-Aware Node-level Task Allocation for Next-Generation Green HPC Systems

Francesco Beneventi[†], Andrea Bartolini[†§], Carlo Cavazzoni[‡] and Luca Benini[†§]
[†]Department of Electrical, Electronic and Information Engineering (DEI), University of Bologna, Italy
{francesco.beneventi, a.bartolini, luca.benini}@unibo.it
[§]Integrated Systems Laboratory, ETH Zurich, Switzerland {barandre, lbenini}@iis.ee.ethz.ch
[‡]Cineca, Italy {c.cavazzoni}@cineca.it

***Abstract*—Energy-efficiency is of primary interest in future HPC systems as their computational growth is limited by the supercomputer peak power consumption. A significant part of the power consumed by a supercomputer machine is caused by the cooling infrastructure. Todays thermal design is based on coarse grain models which consider the silicon die of the processing elements as an isothermal surface. Similarly feedback control loops uses the same assumption to modulate the cooling effort with the goal of reducing cooling cost and maintaining the silicon temperature in a safe working range. Recent processors development has brought into the market CPUs that integrate a large number of complex cores. Differently from massively parallel CPUs for which the area and power consumption of each core is very limited, the cores of these processors can consume tens of watts and thus, under heterogeneous workloads, creating significant thermal gradients. In this paper we first characterize the power and thermal characteristics of new server-class Intel Xeon computing node based on Haswell v3 architecture considering both the computational and the cooling components. We show that these systems are characterized by significant on-die thermal gradients and that the current O.S. task allocation strategy is not capable of taking advantage of that, leading to max CPU temperature and extra cooling activity. To solve this issue we propose a novel task allocation strategy that reduces the cooling power while matching the HPC performance requirements.**

I. Introduction and Related Work

Even if the end of Dennard's scaling [7] has marked the end of clock scaling, the pace dictated by Moore's law has now made possible to integrate in the same die several billions of transistors [9]. On top of heterogeneous architectures which use the available transistors to extend the general purpose processors with extra functionality and HW-accelerated computing kernels (i.e. cryptography, graphic and video processing units, GPUs, VPUs, big data accelerators, etc), multi-core and many-core architectures replicate the same core several times on the same die to keep satisfying the users performance requirements.

Todays multi-core and many-cores platforms have become a mass product permeating several market segments: spanning from the ultra-low power mobile domain, to the high performance computing domain [5], [9], [13], [14], [17]. Even if multi-core architectures are significantly more energy-efficient than single-core ones, the transistors count achievable by todays technology together with the peak performance required by todays datacenters and supercomputers [8] have made todays systems thermally and power limited. Differently from mobile architectures which are constrained by the battery power and large idle periods, in the high performance computing domain the processors are thermally constrained. Indeed they are required to sustain the peak performance which leads to long high power consumption phases and costly cooling solutions.

Even if the cooling cost of these machines can be reduced by adopting advanced cooling strategies, such as liquid cooling and free-cooling, due to physical constraints and high investment costs a large slice of servers and supercomputers are still be based on air-cooling. In an air-cooled blade, cold air is forced to flow through the processors heatsink by mean of a set of rotating fans present inside each server node. Authors in [16] show that fan power can account for up to 23% of typical server power and scales super-linearly with node utilization.

Several works in literature have proposed techniques to control the rotating speed of the fans to achieve a higher energy efficiency. This can be done either by reducing the fan speed and balancing the power gain in the fans with the power loss due to the increased leakage power caused by the higher silicon temperature [4], [10], [18] or by balancing the CPU temperature by mean of task migration and dynamic thermal management [1], [6].

Authors in [4], [10] show that the fan power can be reduced by iteratively modulating the fan speed based on die temperature and a node power measurements. Indeed by reducing the fan power iteratively with small steps, until the silicon temperature raises and the overall power decreases, it is possible to account for leakage power while finding the energy minimum. This technique can lead to an energy saving of the 5%. Authors in [18] use only load measurements and a pre-calibrated open-loop controller based on a set of look-up tables to find the optimal fan speed based on the server input load. With this technique the authors were able to reduce the fan power, achieving up to the 9% of energy-efficiency gain. Differently Lee et al. [11] use the signal coming from an external thermocouple, positioned on the heat spreader, to implement an optimal PID controller scheme. The results show that up to 14% of a servers fan cooling power can be saved if the fan control permits a small overshoots in the thermal response.

In case of multi-core and multi-socket systems the fan power can be reduced by migrating tasks in between different

cores to take advantage of the thermal capacitance of the heat dissipation materials, reducing the overall die temperature by migrating jobs in between the hot and cold cores [1], [6]. Even if these techniques can achieve up to the 78% of cooling power reduction they require to constantly migrate workloads in between cores. This can be detrimental for the performance of HPC applications which is often based on message passing, for which best design practices suggest to bind the MPI tasks to the different cores. Moreover these analysis have been conducted on simulators which are based on general assumptions about the heat propagation in real digital designs and thus they cannot model the real behaviour of a complex supercomputing systems. Authors of [3], [12] proposes a combination of machine learning and constraint programming to extract the thermal interaction in between cores of a many-core system and translating them in rules for mapping tasks to cores. These rules are embedded as constraint in an optimization problem which find the best core in which running each task. Even if the proposed technique is shown to reduce cores slowdown induced by thermal protection mechanisms it is evaluated only in simulations and is not clear how it would perform in a real supercomputing systems.

By looking at todays Top500 list (November 2015), which ranks the worldwide supercomputers by their peak double-precision floating point operations per second (GFLOPs) we can notice that 85% of the supercomputers today uses Intel Xeon Class processors and that 26% uses recent Intel Xeon Haswell processors [9]. In the previous top500 edition (June 2015) the supercomputers based on Intel Haswell processors were at 19% (5% in November 2014) showing a growth of almost 2x in the last six months. Previous generation Intel's CPU, namely Sandy Bridge were used by 46% of November 2014 supercomputers. This suggests that in the near future Intel Xeon processors based on Haswell architecture could power the half of worldwide supercomputers. Todays fastest Intel Xeon Haswell CPU integrates 18 cores on the same die and achieves a theoretical peak performance of almost 700 GFLOPs. This extraordinary peak performance is possible only by an increased complexity in the core logic which leads to large area and power density. The elevate number of core integrated in the same die and the large area and power consumption of each core make this device significantly different from previous multi-core and many-core systems [15], suggesting the presence of strong thermal effects and not idealities which, if modeled, can be used by dynamic thermal management policies to further increase the energy-efficiency of a large share of future green supercomputers.

In this paper we show the characterization of the thermal and power effects on the Intel Xeon E5-2699 v3 processors and server node. We first show that this device is affected by strong on-chip thermal gradient during normal operation and balanced workload (up to 10^oC), which can increase to 24^oC under unbalanced workloads showing practical opportunities for DTM techniques. Secondly we measured the impact of the fan power on the overall power budget which can cause an additional 20% power loss during peak computational phases. Finally we show the effectiveness of a job allocation techniques which minimize the package temperature reducing the fan speed and power. The proposed technique does not affect the computational performance and is capable of achieving the 4% of energy saving during load peak. To the best of authors knowledge this work is the first analysis which characterizes the thermal heterogeneity present in large server-class multi-cores based on large Out-of-Order cores, showing the potentials and the challenges of DTM techniques for this class of devices.

II. Architecture and Modeling

In this section we describe the target Intel Xeon E5-2699 v3 based server platform, the power and thermal characterization results. As early introduced this is the first study which quantifies the impact of thermal variation on top-class multi-cores for next generation green supercomputing platforms.

This work is based on a 2U Intel "Wildcat Pass" server platform. It was configured with two Intel Xeon E5-2699 v3 (Haswell) and 128GB of DDR4 RAM and it is air cooled. The server chassis is a 2U full-width type.

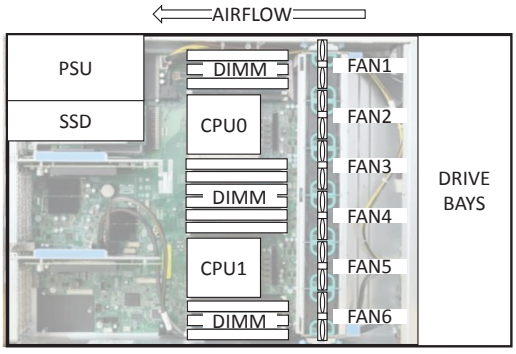

Fig. 1. 2U Intel "Wildcat Pass" server platform

From Fig.1 we can see that the two CPU sockets are placed in front of the six fans that pull the cold air from the front of the blade and push it toward the CPU's heatsinks and DRAM. As a result the two CPU are equally cooled by the airflow. The fans provide a variable airflow regulated by the on-board fan controller. In this chassis two separated controllers are implemented, one per CPU, each independently controlling the speeds of three of the six fan as described in Fig.2. The inputs of the controllers are the package temperatures T_{pkg0-1} which can be directly read from the server telemetry system. The main task of the controller is to maintain the CPU temperatures T_{pkg0-1} below a safe physical limit fixed to $T_{treshH} = 66\ ^oC$. In particular the controller is in the active state, i.e. it enables its output, when the input temperatures T_{pkg0-1} are respectively higher than a lower temperature threshold $T_{treshL} = 56\ ^oC$. Otherwise the fan speed is fixed at its minimum value (3600 RPM c.a.). The server platform features a set of sensors which can be queried on-line through the IPMI interface. They are used to monitor each fan speed (Fan RPM), the

power consumed by the two power supply units (PSU) and the package temperature T_{pkg0-1} with a sampling time of 2s.

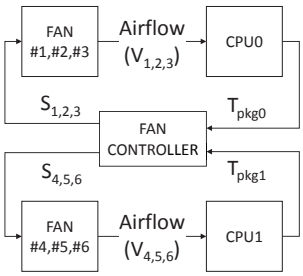

Fig. 2. Fan controller

The tested Intel server integrates two Intel Xeon E5-2699 v3 which is the top-class server processor based on "Haswell v3" architecture and featuring 18 cores, 36 HW threads, 2.3GHz of nominal frequency and 3.6 GHz of maximum frequency with a TDP of 145Watts. The Intel server processors belonging to the "Haswell v3" architecture introduce a series of architectural novelty in terms of energy efficiency. The instruction parallelism is increased thanks to the introduction of the AVX2, a new ISA extension based on 256-bit wide integer SIMD instructions. Together with the new FMA3 (Fused Multiply-Accumulate) instructions, this new architecture is able to bring the whole peak performances to 16 FLOPS/cycle in double precision. However, due to the increased processing density, the AVX HW units are more demanding in terms of power consumption than the other core units. In addition to turbo mode, to enforce the TDP constraints in the case of AVX-based workloads, the Haswell CPU provides a mechanism to lower the core frequency when a certain number of AVX instructions are executed over a period. The AVX frequencies are selected according to the number of active cores and TDP limits.

A second major improvement of the "v3" server processors is an enhanced DVFS infrastructure that features on-die and per-core voltage regulators. This reduces the power consumption by reducing the absolute current flowing in the CPU and enables more fine grained and aggressive thermal and power management. As previous Intel architectures, the selected device has built-in performance counters (PMU) which can be queried by the software to obtain architectural metrics (IPC, CPU load and current clock frequency) and physical parameters such as (per core temperature and per CPU power consumption). To monitor these parameters online we used a similar approach to the one used in [2] with a sampling period of 2s. In the next section we use this monitoring infrastructure to characterize the performance and power trade-off of the target architecture.

A. Power and performance characterization

In this section we report the results of a set of tests aiming to highlight the corner-case behaviors of the target server blade and Haswell processor in terms of power consumption and performance.

In the first test we isolate the different blade power contributions by running the same PowerVirus[1] on all the 36 cores (two sockets) while measuring all the different available sensors. Fig.3 shows the cores load (Load) and temperatures (Core Temperatures), the average fan speed (Fan Speed) and PSU total power (Power). From figure we can notice that, after

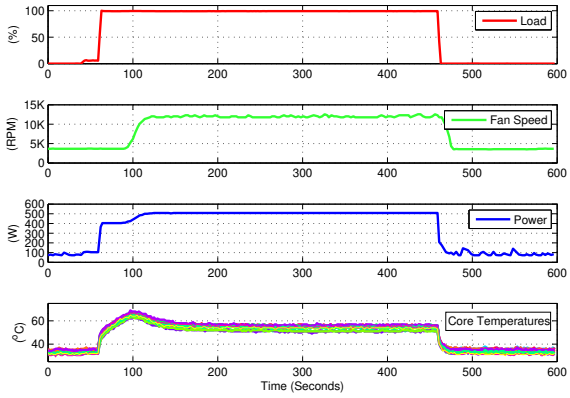

Fig. 3. Power step test results.

the PowerVirus starts, all the cores see a load step (60s), the power increases as well and the temperature of the CPUs too.

TABLE I
POWER BREAKDOWN

	PSU power (W)	2xCPU power (W)	2xDRAM (W)
Idle	76.4	36.7	8.2
Full load (fan@minimum speed)	404.7	289.8	24.5
Full load (fan@maximum speed)	508.4	289.8	24.5

It can be noticed that the thermal transient seen by each core is characterized by two time constants. Indeed the temperature increases of almost 10^oC in few seconds and then takes almost 30s to increase of a similar quantity. As a consequence of the cores temperature increase the fans increase their rotation speed (95s) as well, leading to a power overhead and efficiency loss which is visible in the Power plot of Fig.3. Table I quantifies the power consumption of the different components in the server node. From it we can notice that the CPU and DRAM account for the 59% of the total power in the Idle state and almost the same at full load (61%). In the same circumstance the extra fan speed, needed to cool down the CPUs, causes a 20% increase in power consumption. If we consider an ideal cooling circuit (with no extra fan power) the CPU and DRAM power would increase their impact on the total power of the 78%. This however is far from a fully energy proportional system.

These results suggest that dynamic thermal management (DTM) policies aiming at reducing the fan speed can lead up

[1]Cpuburn power virus by Robert Redelmeier: it takes advantage of the superscalar architecture to maximize the CPU power consumption. The binary used in this work is "burnP6" installed from default repository.

to the 20% of gain in energy-efficiency. Moreover the dynamic power management (DPM) techniques, which increase the energy-efficiency of the CPUs, impact only for the 60% on total node power in air-cooled future servers. This percentage increases up to the 80% in ideal free-cooled/water-cooled future server nodes.

B. Thermal characterization and hardware heterogeneity

In this section we report the results of a set of tests aiming to highlight the thermal behaviors and characteristics of the target server blade and Haswell processor. As we will show there are significant thermal heterogeneity sources which can be exploited by DTM policies.

1) Test 1: steady-state and fan control active: As early introduced, our test system is equipped with an Intel E5-2699 CPU which features 18 cores over a chip area of 662mm^2. The first test aims to measure the cores homogeneity from the thermal dissipation perspective and the maximum achievable thermal gradient. The basic idea is to apply the same workload (PowerVirus) to each core, one by one for a total of 36 experiment steps. At each step we measure the steady-state temperatures for all the cores and also the corresponding speed of the fans (1-6). The final results of this test are collected in the tables showed in Fig. 4. The left subplot shows on the x-axis the Core-Id, on the y-axis the experiment step (i-th experiment step = i-th core active) and the color code represents the core steady-state temperature. Core-Ids 0-18 and Core-Ids 19-36 share the same die. The right subplot shows on the x-axis the Fan-Id and with color code the fan speed. In this experiment all cores have turbo mode enabled to obtain the highest power consumption. However during the tests we verified that the active cores are running at the same real frequency (3.6GHz).

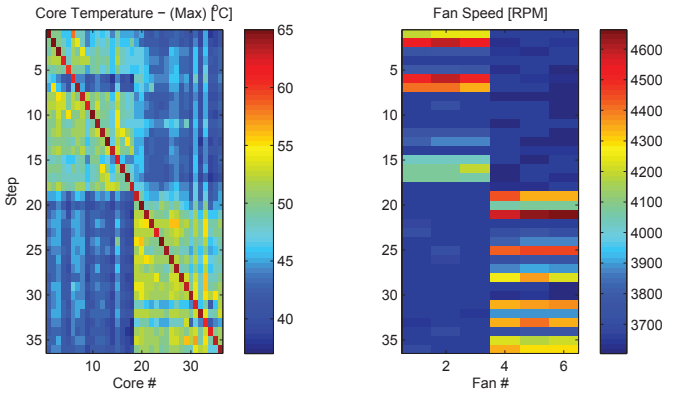

Fig. 4. Intra-die and inter-die thermal variation

The first information which we can extract from the left plot is the presence of a significant thermal gradient within the same die. For all the experiment steps, the active core reaches the steady-state temperature amounting to $60-65^oC$. The minimum temperature between inactive cores of the same CPU is between 38 and 48^oC. Compared to its idle temperature, the active core increases its own steady-state temperature of almost 20^oC. The maximum intra-die thermal gradient is 24^oC (experimental step 8), while the average gradient (among the experiment steps and considering only active CPUs) is 18.2^oC. If we consider the inactive CPU only (CPU1, from experiment step 19-36) we see an average thermal gradient equal to 6.2^oC.

In addition to the strong presence of thermal gradients, from the same figure we can see that, according to the active cores position, a subset of neighboring cores are hotter then others. We can appreciate a temperature difference of almost 10^oC between the active core and the first thermal neighbors. This temperature gradient increases of additional $8-10^oC$ if we consider the coldest core in the same die. From the same plot we can notice that there is no inter-die thermal interaction. The maximum inter-die gradient is 27^oC and the average gradient is 24^oC, that is 5.8^oC higher than the intra-die average gradient. From the DTM perspective this means that the core temperature depends primarily by its own power consumption and by a subset of thermal neighbors, suggesting opportunities for distributed DTM policies.

Finally from the right plot of Fig.4 we can see that, even if in all the experiment step the cores have executed the same benchmark and at the same operating point, some cores have activated the fan while other not. This suggests that not only the cores show significant thermal variability inside the die but their thermal impact on the package temperature (T_{pkg0-1}) depends on the core position. This effect can be exploited by thermal-aware job allocation policies to allocate power intensive jobs in the cores which impact less the package temperature. It must be noted that the similarity in the steady-state temperature of the active core was an effect of the fan control policy.

2) Test 2: steady-state and fan control not-active: The second test tries to better quantify the thermal heterogeneity present between similar cores in the same die and decoupling it from the fan speed noise. For this purpose we exploited the RAPL power capping feature to fix a power budget of 49W to each CPU. This value is empirically chosen to avoid the increase of the fan speed while running the PowerVirus in each core. In this way, observed thermal variations are due to different cooling efficiency for the cores rather than different power. In this test we used only the CPU0 and we run the PowerVirus sequentially, on one core at a time while the others are kept idle, for a total of 18 step experiment. Then, for each step, we measured the steady-state temperature of the active core. Every step runs for 10 minutes and between two consecutive tests the system was kept idle for another 10 minutes. The temperature values resulting from these tests are the average (over a time interval of one minute) of the temperature samples measured after the thermal transient has been settled (steady-state).

Fig.5 collects the results of this test. It shows the steady-state temperature reached by the core in each of the 18 experiment steps. On the x-axis there is the active Core-Ids while the y-axis reports its own steady-state temperature. Active Core-Ids are ranked in a descent order. From the plot we can notice that the core 6 is the hottest one while core

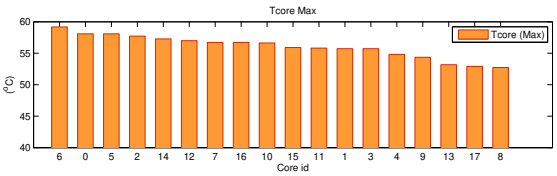

Fig. 5. Tcore max (same job in all the cores but at reduced frequency to avoid fan activation)

8 is the coldest, showing a difference of almost 7^oC. It is clear that under thermal limitations an optimal DTM policy will reduce the performance of core 6 in the measure of the 10% more than of the core 8. This will translate in a visible thermally induced performance heterogeneity inside future green supercomputers.

As already seen from Fig. 4, not all the cores affect similarly the package temperature and thus not all the cores require the same amount of cooling. To evaluate this, differently from Fig.5, in Fig.6 we have ranked the cores according to their impact on the package temperature (Tpkg). On the figure we show on the y-axis, with different bars, the core maximum temperature and the corresponding package temperature. From the plot we can notice that self-heating and cooling cost are two similar objective metrics but the resulting core rank is different. Indeed the ranking obtained ordering the cores according to their cooling cost is different from the ranking obtained ordering them considering their sensitivity to self-heating.

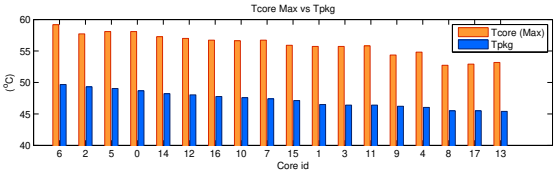

Fig. 6. Tcore max vs Tpkg (same job in all the cores but at reduced frequency to avoid fan activation)

As a matter of fact, upcoming high-performance computing nodes based on large multi-cores CPUs are affected by a significant thermal heterogeneity and cooling heterogeneity. DTM can exploit this kind of heterogeneities to reduce the active-cooling costs and power consumption that impacts for the 20% of the total system power. In the next section we will validate these considerations by implementing directly on the Intel Xeon E5-2699 v3 HW a thermal-aware job allocation policy.

III. Test case: thermal allocation for fan control

In this section we take advantage of the previous characterization results to implement on the Intel Xeon E5-2699 v3 processor a thermal-aware job dispatcher which takes advantage of the intrinsic thermal heterogeneity to reduce the cooling cost, i.e. the fan speed. With this test we aim to underline the importance and the feasibility of DTM policies which exploits the thermal heterogeneity on real next-generation green supercomputer hardware.

We started by analyzing the job entering in a real supercomputer, in a production environment, to evaluate their thermal and power heterogeneity. For this purpose we analyzed the traces recorded from the job scheduler running on the EURORA supercomputer [2]. Eurora is employed by several users for different applications ranging from weather forecasting and big data analysis to heavy scientific workloads, thus it represents a good sample for our purposes. We calculated the average temperature and power of each job executed in a time window of 3 months and the results are displayed in Fig.7. Each point of the scatter plot represents a single job that ran on a single or multiple nodes. In the figure we can notice several clusters of points which distributes along virtual lines. These are jobs spanning an increasing number of nodes. If we check the distribution of the temperature values (on the left) we notice that the jobs temperatures are collected into two main heterogeneous groups, the first at high temperatures (hot jobs) and the second at relatively cold temperatures (cold jobs). Moreover, looking at the power distribution on the bottom, we see that the majority of the EURORA jobs ran on a single node.

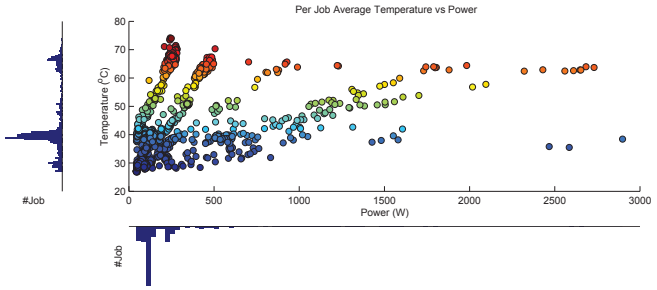

Fig. 7. Average (per job) power and temperature for the EURORA system

This analysis highlights the presence of jobs with heterogeneous power requirements on the supercomputer workload. Moreover naturally the jobs tend to cluster in hot and cold jobs. This property of supercomputer workload matches perfectly the thermal heterogeneity of the tested Intel Xeon E5-2699 v3 which is representative of future high performance computing infrastructure. In the following subsections we leverage these properties together to deploy a simple but yet effective allocation policy which reduces the fan power while preserving the computational performance.

A. Cooling-Aware Job Allocation Policy

Starting from the the cooling system properties of the tested device, where the fan speed is regulated by a controller based on the package temperature signal, we analyze the impact of each single core temperature to the package temperature. From it we can deduce a simple model that maps the core temperatures to the package temperatures. This model, in turn, can be used proactively to map a given job to a certain core by looking at:

- job average load
- core thermal influence to package temperature.

As example, jobs having an high load (hot jobs) can be mapped to cores having a low influence on the package temperatures (cold cores) and, in turn reducing to the fan activity. In the following text we made the assumption that hot jobs are composed by hot tasks, as well as cold jobs are composed by cold tasks.

This model can be easily learned using the technique showed in Section II-B2. The model is learned offline and is a list of core ranked from the core which impacts less the package temperature to the one which impacts more. The model obtained for the Intel Xeon E5-2699 v3 platform is showed in figure Fig.6. The online allocation algorithm (1) first ranks the job to be executed in a descend order according to their average load (Hot job first) and then, based on the platform model, (2) allocates the first job of the job list into the first core of the core list.

To evaluate the proposed algorithm we have created a series of synthetic tasks, in a number equals to the number of the cores of a CPU. Each task has a different load which is generated randomly with a normal distribution in the range of 30% and 100% core load. This set of task is generated only once and used for all experiments in order to compare the resulting behaviour. Successively the scheduler routine allocates each task to the cores of the real Intel Xeon CPU following different policies. To compare the effectiveness of the proposed approach we consider different task allocation policies:

- OS: This is the ordinary case, where the tasks are allocated to the CPU by the default Linux task scheduler which performs load balancing.
- Random: this policy binds each task to a specific core in a random order.
- Cooling Aware: in this case, the core affinity of each task is established considering the Cooling Aware model explained previously in this section.

For each test case and allocation policy we measure the total blade power consumption which includes the fan power and the whole server board. This allows to consider also the potential increase of the CPU's leakage power. Indeed since we are going to lower the speed of the fans, the average CPU temperature increase may induce more leakage power. Moreover, we monitor the fan speed to evaluate the direct impact of the policy in terms of RPM reduction. These results are obtained using only one CPU (CPU0) and with turbo enabled.

In Fig.8 we report the variables measured during the test: the fan speed for the active CPU, the total power consumption of the board taken from the PSU sensors and the temperature of the CPU package. In this sets of results we report only the results for one socket as the other one was giving similar results. From them we can notice that our policy decreases the overall power and fan speed. To have a more robust evaluation of the performance of the designed policy we performed a set

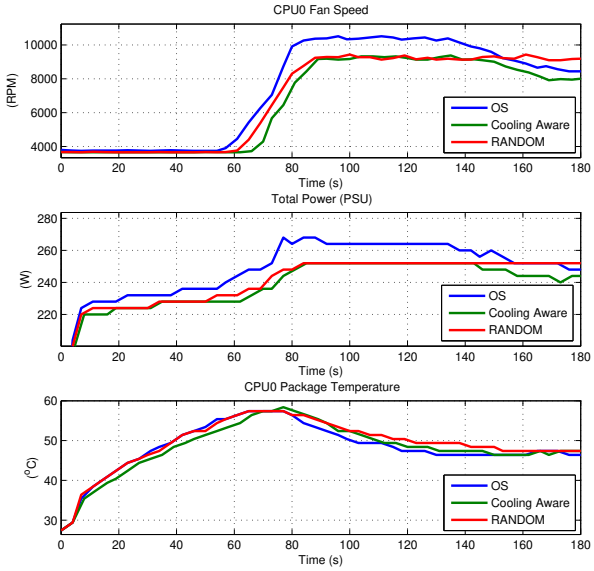

Fig. 8. Comparison of the different policies during the tests execution

of different run for each policy and we collected aggregated results which are resumed in Fig.9. In this plot we correlate the energy consumption (PSU) of the server board with the average speed of the fans. The values are normalized w.r.t. the maximum value. We can notice that the Cooling-Aware scheduler, in the average, performs better than the "Random" and the "OS" approach which shows the worsts results.

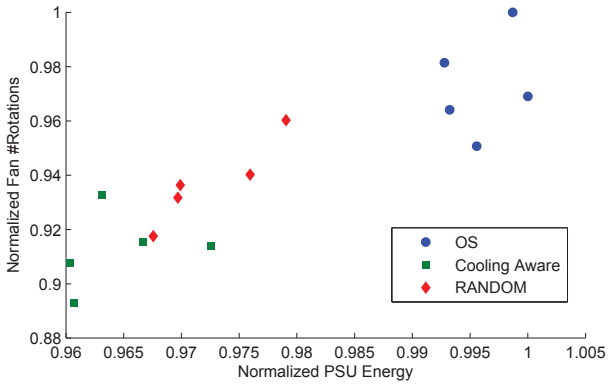

Fig. 9. Normalized fan activity versus average PSU energy for the different policies

The proposed cooling-aware job allocation policy can lead to an overall increase in the energy-efficiency of up to 4%. The results of this test are a confirmation that future server architecture can take advantage of their thermal heterogeneity to improve the overall node energy-efficiency by reducing the cooling cost.

B. Future Works

In future works we aim to validate this results by considering real-workloads, the coupling of the thermal-aware

allocation strategy with standard power management policies and a multi-blade and rack system. In our vision to become applicable to a real HPC environment several additional components still need to be developed: (i) a job power predictor which allows to estimate if a job which is going to execute in the machine is going to be hot and cold. This will enable the batch job scheduler to decide where to allocate the job; (ii) modelling the thermal heterogeneity at the rack level. Indeed air-cooled blades receives a different quantity of inlet cold air accordingly to the their height with respect to the floor; (iii) a more open fan-speed control policy which allows a co-design of the optimal cooling-aware power management and allocation policy.

IV. Conclusion

This work evaluates the thermal and power characteristics of server node based on the Intel Xeon E5-2699 v3 CPU. This CPU is peculiar because it integrates on the same silicon die a large number (18) of powerful processors. We proof with empirical data that these devices are affected by strong on-chip thermal gradient during normal operation and balanced workload (up to 10^oC), which can increase to 24^oC under unbalanced workloads showing practical opportunities for DTM techniques. We show that fan power accounts for the 20% of the node power and we show that job allocation strategies which minimizes the package temperature are capable of reducing the fan speed and power improving the system energy-efficiency. The proposed technique is capable of saving the 4% of energy during peak load without affecting the computational performance. To the best of author's knowledge, this work represents the first analysis which characterizes the thermal heterogeneity present in a large server-class multi-cores CPU based on "fat" cores, showing the potentials and challenges of DTM techniques for this class of devices.

V. Acknowledgments

This work was supported, in parts, by the FP7 ERC Advance project MULTITHERMAN (g.a. 291125), by the EU H2020 FETHPC project ANTAREX (g.a. 67623) and by the YINS RTD project (no. 20NA21 150939), evaluated by the Swiss NSF and funded by Nano-Tera.ch with Swiss Confederation financing

References

[1] R. Ayoub, S. Sharifi, and T. S. Rosing. Gentlecool: Cooling aware proactive workload scheduling in multi-machine systems. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 295–298. European Design and Automation Association, 2010.

[2] A. Bartolini, M. Cacciari, C. Cavazzoni, G. Tecchiolli, and L. Benini. Unveiling eurora-thermal and power characterization of the most energy-efficient supercomputer in the world. In *Proceedings of the conference on Design, Automation & Test in Europe*, page 277. European Design and Automation Association, 2014.

[3] A. Bartolini, M. Lombardi, M. Milano, and L. Benini. *Principles and Practice of Constraint Programming – CP 2011: 17th International Conference, CP 2011, Perugia, Italy, September 12-16, 2011. Proceedings*, chapter Neuron Constraints to Model Complex Real-World Problems, pages 115–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[4] J. Chen, R. Tan, G. Xing, and X. Wang. Ptec: A system for predictive thermal and energy control in data centers. In *Real-Time Systems Symposium (RTSS), 2014 IEEE*, pages 218–227. IEEE, 2014.

[5] F. Conti, D. Rossi, A. Pullini, I. Loi, and L. Benini. PULP: A Ultra-Low Power Parallel Accelerator for Energy-Efficient and Flexible Embedded Vision. *Journal of Signal Processing Systems*, pages 1–16, 2015.

[6] A. K. Coşkun, K. Whisnant, K. C. Gross, et al. Static and dynamic temperature-aware scheduling for multiprocessor SoCs. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 16(9):1127–1140, 2008.

[7] R. H. Dennard, V. Rideout, E. Bassous, and A. Leblanc. Design of ion-implanted MOSFET's with very small physical dimensions. *Solid-State Circuits, IEEE Journal of*, 9(5):256–268, 1974.

[8] J. J. Dongarra, H. W. Meuer, E. Strohmaier, et al. Top500 supercomputer sites. *Supercomputer*, 11:133–133, 1995.

[9] P. Hammarlund, R. Kumar, R. B. Osborne, R. Rajwar, R. Singhal, R. D'Sa, R. Chappell, S. Kaushik, S. Chennupaty, S. Jourdan, et al. Haswell: The fourth-generation Intel core processor. *IEEE Micro*, (2):6–20, 2014.

[10] W. Huang, M. Allen-Ware, J. B. Carter, E. Elnozahy, H. Hamann, T. Keller, A. Lefurgy, J. Li, K. Rajamani, and J. Rubio. TAPO: Thermal-aware power optimization techniques for servers and data centers. In *Green Computing Conference and Workshops (IGCC), 2011 International*, pages 1–8. IEEE, 2011.

[11] C. Lee and R. Chen. Optimal self-tuning pid controller based on low power consumption for a server fan cooling system. *Sensors*, 15(5):11685–11700, 2015.

[12] M. Lombardi, M. Milano, and A. Bartolini. Empirical decision model learning. *Artificial Intelligence*, pages –, 2016.

[13] S. Raghav, M. Ruggiero, D. Atienza, C. Pinto, A. Marongiu, and L. Benini. Scalable instruction set simulator for thousand-core architectures running on gpgpus. In *High Performance Computing and Simulation (HPCS), 2010 International Conference on*, pages 459–466, June 2010.

[14] D. Rossi, A. Pullini, I. Loi, M. Gautschi, F. K. Grkaynak, A. Bartolini, P. Flatresse, and L. Benini. A 60 gops/w, 1.8 v to 0.9 v body bias {ULP} cluster in 28 nm {UTBB} fd-soi technology. *Solid-State Electronics*, 117:170 – 184, 2016.

[15] M. Sadri, A. Bartolini, and L. Benini. Single-chip cloud computer thermal model. In *Thermal Investigations of ICs and Systems (THERMINIC), 2011 17th International Workshop on*, pages 1–6. IEEE, 2011.

[16] Z. Wang, C. Bash, N. Tolia, M. Marwah, X. Zhu, and P. Ranganathan. Optimal fan speed control for thermal management of servers. In *ASME 2009 InterPACK Conference collocated with the ASME 2009 Summer Heat Transfer Conference and the ASME 2009 3rd International Conference on Energy Sustainability*, pages 709–719. American Society of Mechanical Engineers, 2009.

[17] S.-H. Yang, S. Lee, J. Y. Lee, J. Cho, H.-J. Lee, D. Cho, J. Heo, S. Cho, Y. Shin, S. Yun, E. Kim, U. Cho, E. Pyo, M. H. Park, J. C. Son, C. Kim, J. Youn, Y. Chung, S. Park, and S. H. Hwang. A 32nm high-k metal gate application processor with GHz multi-core CPU. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 214–216, Feb 2012.

[18] M. Zapater, J. L. Ayala, J. M. Moya, K. Vaidyanathan, K. Gross, and A. K. Coskun. Leakage and temperature aware server control for improving energy efficiency in data centers. In *Proceedings of the Conference on Design, Automation and Test in Europe*, pages 266–269. EDA Consortium, 2013.