

Analyzing energy/performance trade-offs with power capping for parallel applications on modern multi and many core processors

Adam Krzywaniak*, Jerzy Proficz†, Pawel Czarnul*

* Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology
Narutowicza 11/12, 80-233 Poland

Email: adam.krzywaniak@pg.edu.pl, pczarnul@eti.pg.edu.pl

† Centre of Informatics — Tricity Academic Supercomputer & network (CI TASK)
Gdansk University of Technology
Narutowicza 11/12, 80-233 Poland
Email: j.proficz@task.gda.pl

Abstract—In the paper we present extensive results from analyzing energy/performance trade-offs with power capping observed on four different modern CPUs, for three different parallel applications such as 2D heat distribution, numerical integration and Fast Fourier Transform. The CPU tested represent both multi-core type CPUs such as Intel® Xeon® E5, desktop and mobile i7 as well as many-core Intel® Xeon Phi™ x200 but also server, desktop and mobile solutions used widely nowadays. We show that using enforced power caps we can find points of lower than default energy consumption but mostly for desktop and mobile solutions at the cost of increased execution time. We show with particular numbers how energy consumed, power consumption and execution time change for the point of minimum energy used versus the default configuration with no power limit, for each application and each tested CPU.

I. INTRODUCTION

NOWADAYS the consumption of electric energy by the Information and Communication Technology (ICT) sector reaches extreme values, it is estimated as 269 TWh per year and 2% of global CO₂ emissions. An average data center, having 2 600 m² server rooms, causes almost 2 MW IT load [1]. Thus the energy conservation is very important for such environments as well as for mobile/IoT computations where the battery lifetime can be significantly extended by performing various procedures such as power level capping or calculation offload [2].

Considering the microprocessor devices: Central Processor Units (CPUs) and their applications, usually the actual power level used by such a device is proportional to its current workload. However, many modern CPUs are able to control their maximum power level via special API, e.g. RAPL [3]. Thus, in many cases for such CPUs, energy consumption

Supported partially by the Polish Ministry of Science and Higher Education. The experiments were partially performed using high-performance computing infrastructure provided by Centre of Informatics — Tricity Academic Supercomputer & network (CI TASK) at Gdansk University of Technology.

depends not only on the workload, but also on the actual power cap, set by the managing software or directly by the developer.

For CPUs, it is important to distinguish between the power level and energy consumption — the factor is execution time, as it is presented in the following sections, sometimes the same problem is solved using a lower amount of energy (measured in J or kWh), despite the higher (average) power level (measured in W) observed in the device. There is no simple conversion between these two values, but in many cases, there is a spot where limited power causes lower energy consumption. In this paper we are going to analyze such minima — these can be exploited to trade off between energy consumption and execution time.

Our original contribution covers: (i) the presentation and analysis of execution time, power and energy consumption measurements for different power level caps of various CPU types, (ii) evaluation of trade-off between execution time and energy consumption for three representative HPC applications.

The next section describes the related works, the detailed goal of tests is presented, afterwards performed experiments are described, including the testbed applications, systems as well as the obtained results. And finally conclusions and future work are covered.

II. RELATED WORK

In the context of high performance computing (HPC), energy consumption and energy efficiency are among the most important challenges. It is important in particular for execution that is energy efficient for various levels of utilization [4]. The authors of [5] investigate software methods aimed at improving energy efficiency in parallel computing. In particular it focuses on load imbalance, mixed precision in floating-point operations. Power consumption of compute components is characterized. Energy efficiency metrics are introduced including dynamic energy improvement for n processors. The taxonomy of methods considered in this work includes power-aware

scheduling and resource management, parallelization oriented including balancing, communication focused, approximation methods with part of computations executed with lower energy usage and slight loss of accuracy. In papers [6] and [7] we modeled energy consumption of parallel applications focusing on various communication routines establishing both functions and coefficients valid for various cluster systems.

In terms of measurement of power consumption and energy usage, several tools and techniques have been used and reported. IgProf which is an open source performance profiler is available for x86 and x86-64 as well as ARMv7 and ARMv8 platforms. The authors of [8] added a module for statistical sampling energy profiling. Measurements have been taken using the RAPL interface. The STREAM benchmark has been used to gather results that demonstrate expected correlation between execution time and energy consumption.

RAPL (Running Average Power Limit) provides counters to take measurements of energy consumption of CPUs, integrated GPUs and memory as well as to set corresponding power limits allowing to manage energy efficiency of a system. Paper [9] focuses on measurement and power limiting for main memory for server platforms. It has been shown that power limits can be enforced with minimization of performance impact of the approach. SPEC CPU2000 sub-benchmarks were run for various power limits.

Paper [10] validates DRAM related results from RAPL for desktop and server environments with DDR3 and DDR4 types of memories. RAPL results were compared to actual measurements with matching in general within roughly 20%. Tests were performed with a variety of benchmarks including sleep, HPL Linpack, gcc PAPI, SmallptGPU2 ray-tracer, Kerbal Space Program. RAPL has also been validated in works [3] and [11]. In the latter, the authors concluded that RAPL power estimation is more accurate on IvyBridge than on SandyBridge generation of CPUs. In work [12] and paper [13] the authors investigate power consumption of various components including instruction decoders in x86-64 processor which was done through microbenchmarks. The authors have concluded that decoders consume between 3% and 10% of the total processor package power.

A hybrid hardware/software power capping system called PUPiL was evaluated in [14] for maximizing performance under power capping. The solution was compared to RAPL and e.g. a software-based DVFS control system and a software based decision system. PUPiL showed response time similar to hardware approaches and generally better performance than RAPL under power constraints.

The authors of [15] notice the phenomenon called PERC (Performance-Equivalent Resource Configurations) according to which applications with various configurations of resources show similar performance at various power consumption and use it for their PowerCap algorithm that selects a configuration that follows power limits. The authors claim that the algorithm requires 50% less reconfiguration and 12% more power compared to the DVFS approach.

In paper [16] authors propose an algorithm for scheduling

execution of independent jobs on a system with integrated CPU-GPU with consideration of power caps. The authors have shown that throughput has been improved by between 9% and 46% over default schedules.

As an example, in [17] the author has performed detailed analysis of power consumption of Intel[®] i7-4820K. It should be noted, similarly to our findings for our testbed applications, that the power consumption of an application computing prime numbers reaches roughly 40W at the highest considered frequency at the TDP of the CPU equal to 130W.

The authors of [18] present empirical assessment of vendor provided power capping on a Cray XC40 system and comparison of performance with p-state control. They concluded generally better performance of the latter for many benchmarks in HPC.

III. MOTIVATIONS AND CONCEPT OF RESEARCH

Based on the aforementioned related works, we intend to perform detailed research, for a representative set of HPC applications, into energy/performance trade-offs for modern multi- and many- core CPUs using software imposed power caps.

Specifically, we are looking into such a configuration, for each application and each CPU, for which the total energy consumed during computations is minimized, compared to the default configuration without power consumption caps for a CPU i.e. full computational power. For such energy minimized configurations, we are looking into energy/performance trade-offs. It is especially interesting to analyze and observe it for various modern CPUs that differ, in terms of the target market, design and numbers of cores:

- server: Intel[®] Xeon Phi[™] x200 (many-core CPU), Intel[®] Xeon[®] E5 (multi-core CPU) present in many workstations and cluster nodes,
- desktop: Intel[®] Core[™] i7 desktop present in many home and office computers,
- mobile: Intel[®] Core[™] i7 mobile present in many laptops and notebooks.

The software power caps as well as energy consumption measurements are implemented using RAPL driver available in modern Intel[®] CPUs. Due to technical limitations in measuring the impact of our power caps on the whole server we read the energy consumption using RAPL from the Package (CPU + DRAM) and acknowledge it representative and valuable result.

In terms of applications, we use three parallel applications, that differ in the computing paradigms and compute/synchronization overhead ratios:

- geometric single program multiple data stream: heat distribution,
- master-slave: numerical integration and FFT.

This continues our work [19] of analysis of representative parallel applications with consideration of energy usage.

IV. EXPERIMENTS

A. Testbed applications

For the testing purposes we selected three representative problems found in high performance computing (HPC) environment, and accordingly, implemented three applications, which are executed concurrently, and are horizontally scalable, i.e. speeding up with the increase of the core number; however they utilize shared memory for data exchange and synchronization, thus in this case they cannot be distributed between more compute nodes.

The application were implemented in C language v. C99, using OpenMP [20] for processing parallelization. They were compiled by the GCC v. 4.8 with maximal provided optimization (parameter -O3). They use the default OpenMP configuration (omp directive: schedule(auto)) regarding the thread number and computation partitioning, in the execution environment we did not tune up these settings. The applications use the floating point double precision for calculations (C language type: double).

The first application performs a numerical integration of a given function in a specified range. The specified partition is split between working threads and each thread calculates the sum of its range. The intermediate results are stored separately for each thread, although OpenMP is responsible for their reduction (omp directive: reduction(+:)). For testing purposes we defined the function as $f(x) = \frac{1}{1+x}$. The arguments of the application allow to specify the range and the calculation's precision as a number of subpartitions to be integrated.

The second application is a simplified version of the 2D heat distribution simulation (based on the conception proposed in [21]) over the closed square area, divide into $N \times N$ parts and containing a set of working heaters. For test purposes we set $N = 1000$ and introduced one heater located in the area corner. The input parameters cover a speed of heat propagation and a number of iteration to be simulated. The solution uses three memory buffers: (i) a constant heater map in the room, (ii) an input buffer with the current heat distribution, (iii) an output buffer with the heat distribution after performing current step. The buffers (i) and (ii) are swapped after each step of simulation: the output buffer in step i becomes the input buffer of step $i + 1$. The temperature of each square in the room can be calculated independently, thus potentially the above problem can be parallelized (omp directive: for) among threads as well as the threads do not interfere each other, each one has its own area to perform the simulation.

The third application is a parallel implementation of Fast Fourier Transform (FFT). It uses Radix-2 algorithm with Decimation-in-Time parallelization strategy [22]. At the beginning the sequence of N transformed samples (the input data) is parallelly shuffled, then the $\log_2 N$ iterations are executed, where the parallel computations over complex numbers are performed: each thread has its own range of the data to process (omp directive: for). The result is placed in the array replacing the input data. For the benchmark purposes the input data is automatically generated.

B. Testbed systems

As a testbed environment we used 4 different systems. Two of them contained server dedicated processors with multi-core (Xeon[®] E5 v4) and many-core (Xeon Phi[™] x200) architectures. Another two systems was based on Intel[®] Core[™] i7 processors, one dedicated for desktop and one dedicated for mobile personal computers. Parameters of aforementioned systems are presented in Table I.

C. Results

Obtained results are presented for each testbed system separately. Therefore, for each of the four systems we present three figures individual for each of testbed applications and one common figure. In the individual figures for each power limit preset (bottom axis) we present execution time of the tested application (left axis) as well as total energy consumed (right axis). The common figure presents average power (left axis) for each test run against the power limit (bottom axis) that was preset for each of three tested applications.

Figure 1 presents results obtained using the testbed system with server Xeon E5 v4 for Fast Fourier Transform, simulation of heat distribution and numerical integration respectively. The most important observation is that the testbed applications used in experiments are not able to reach the TDP of server Xeon processor. In each case for the experiment with maximum power limit we use less than 50% of available power. However, when the limit is set below 50% of TDP and close to the reference power consumption the average power consumption starts to respect the enforced limit. For this system the benefits of lowering the power consumption can be observed only for one testbed application (FFT) for which we can find the minimum of energy consumed while running calculations with different power limits. However, the minimum is still saving less than 3% of energy comparing to reference run with no limits. The two other applications have the most energy efficient point at their default settings of the power limit.

Figure 2 presents the results obtained using testbed system with another server processor, Xeon Phi x200, again for Fast Fourier Transform, simulation of heat distribution and numerical integration respectively. Although both server processors present far different architectures (multi-core vs. many-core), the results of the experiments are quite similar. The main common feature of experimental results is that again our testbed applications are using less than 50% of available power. The system respects the preset limit in each case until the minimum value (85W) is reached. For two of tested applications (heat distribution and FFT) we observed the minimum of energy consumed for the value of power limit 135W. However, the energy benefits are not significant again (1-3% energy saved). For numerical integration the lowest energy consumption was obtained again when the power limit had its default value.

Figure 3 presents results of the same testbed applications for the first of non-server testbed systems with mobile PC dedicated Intel[®] Core[™] i7 processor. Results for the last system with another Intel[®] Core[™] i7 processor, dedicated

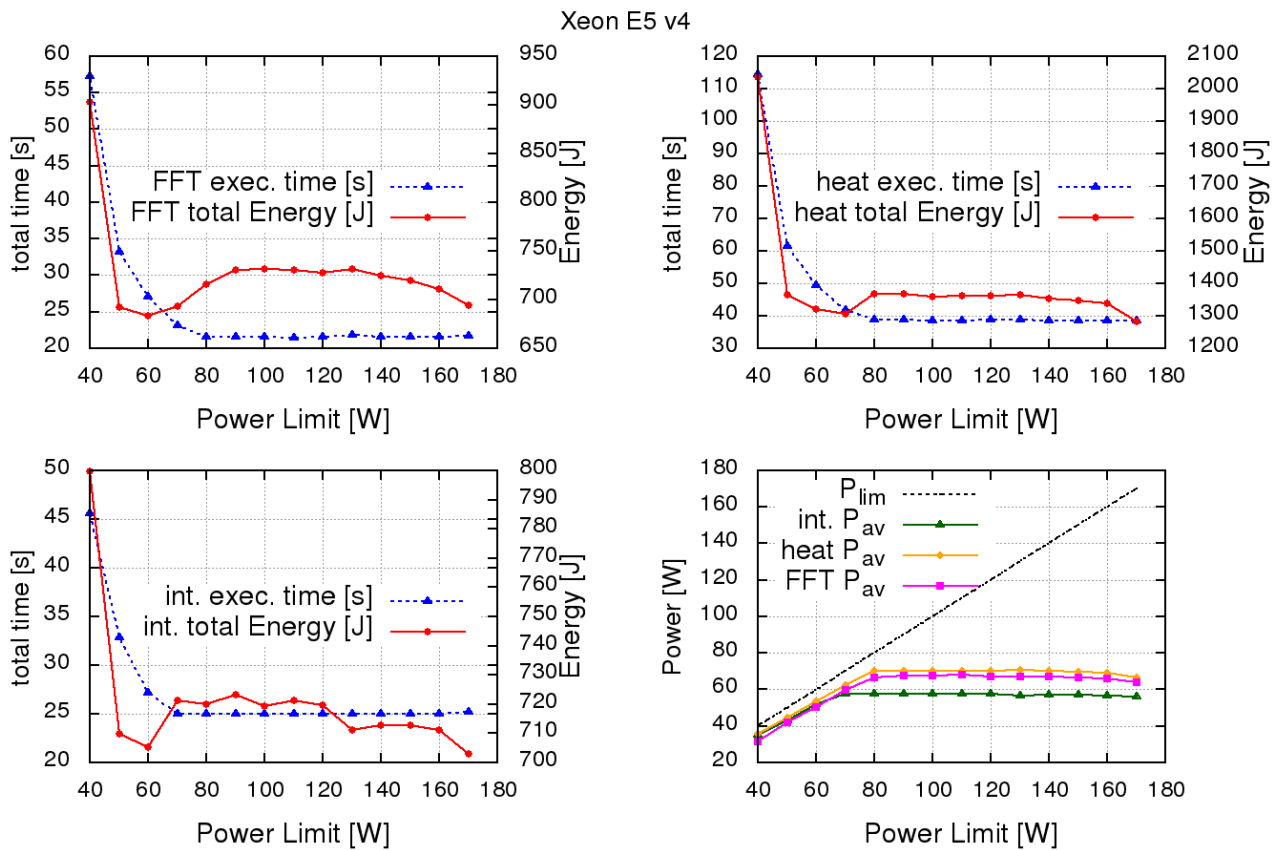


Fig. 1. Tests results for Xeon® E5 v4.

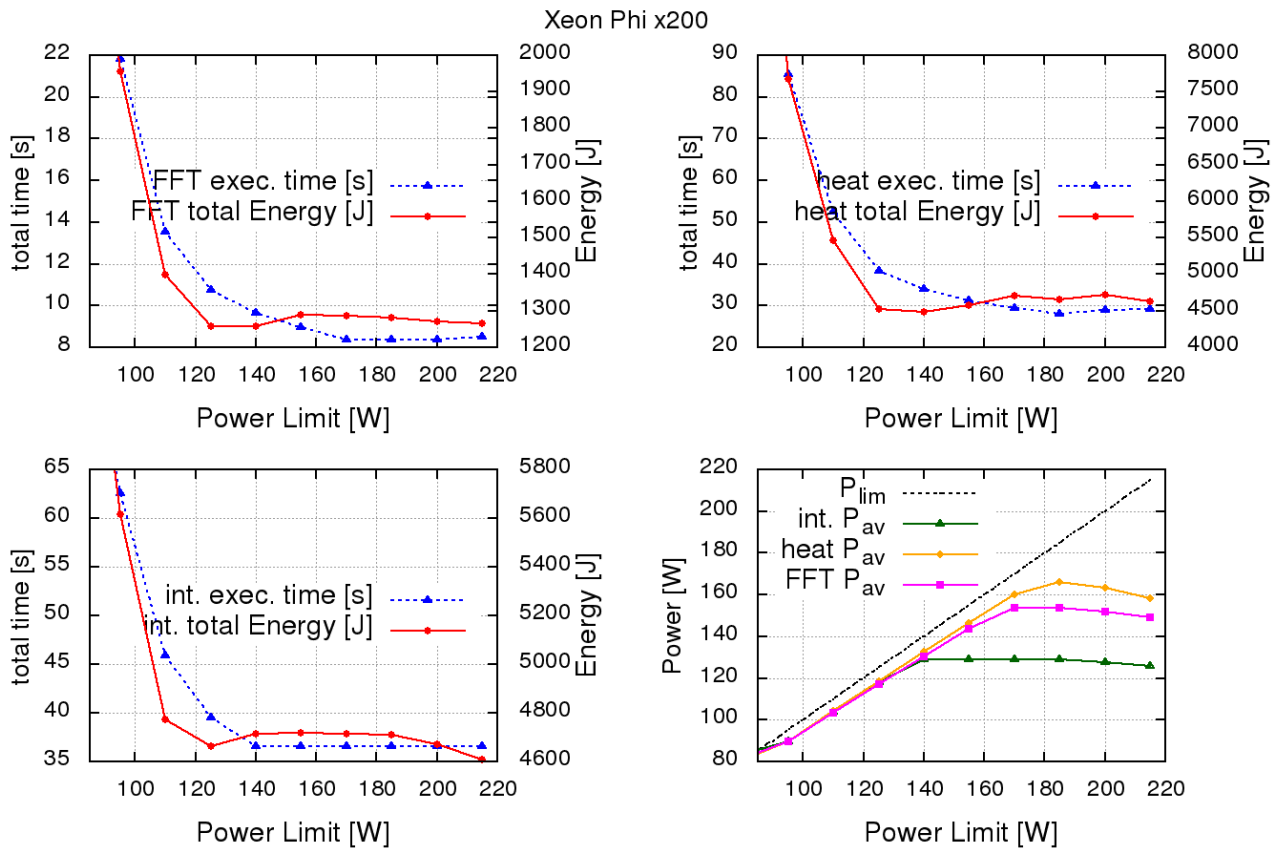


Fig. 2. Tests results for Xeon Phi™ x200.

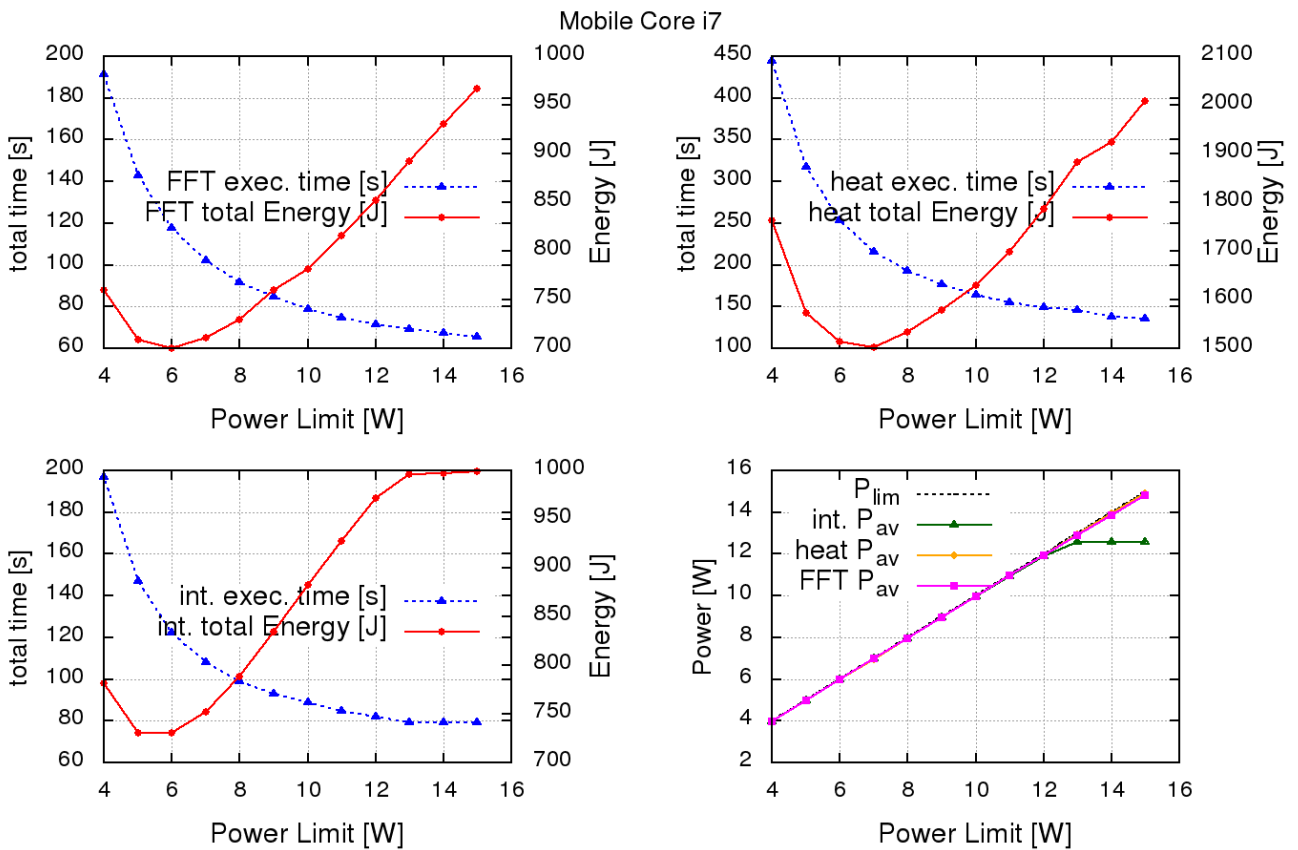


Fig. 3. Tests results for mobile Core™ i7.

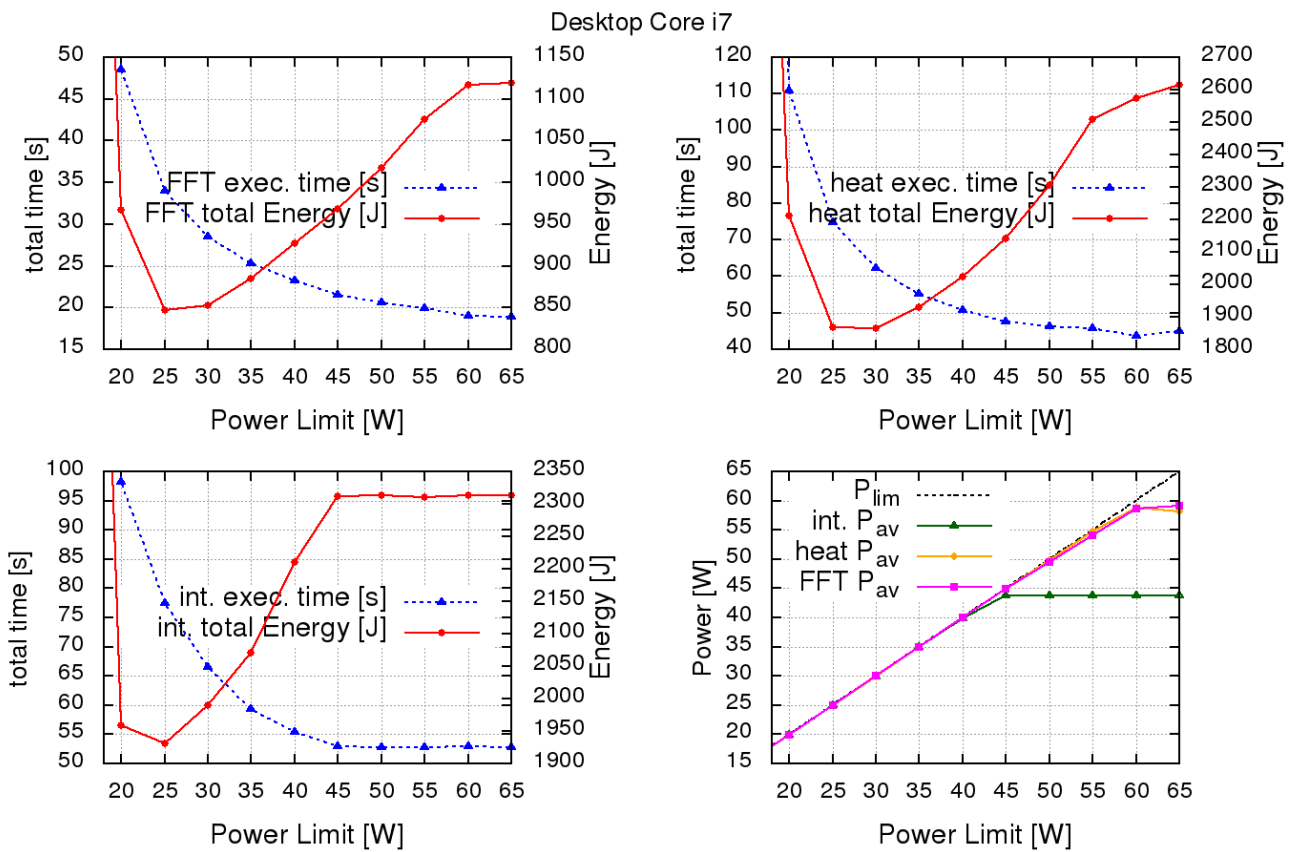


Fig. 4. Tests results for desktop Core™ i7.

TABLE I
TESTBED SYSTEMS USED IN EXPERIMENTS.

System	Processor	Base Frequency	Physical Cores	Logical Cores	Architecture	Cache	RAM
Xeon E5 v4	Intel® Xeon® E5-2620 v4	2.10 GHz	2 x 8	32	Broadwell	2 x 20 MB	128 GB
Xeon Phi x200	Intel® Xeon Phi™ 7210	1.30 GHz	64	256	Knights Landing	32 MB	256 GB
Mobile Core i7	Intel® Core™ i7-5500U	2.40 GHz	2	4	Broadwell	4 MB	16 GB
Desktop Core i7	Intel® Core™ i7-7700	3.60 GHz	4	8	Kaby Lake	8 MB	16 GB

for desktop PC are presented in the last figure, Figure 4. In both mobile and desktop systems the proposed applications seem to generate reasonable load and compared to the server testbed systems much more of available power is used. The level of power consumption is the highest for heat distribution simulation and the lowest for numerical integration. Both systems respect the preset limit well.

For the last two testbed systems we finally observed significant energy consumption benefits caused by limiting the power. The most efficient cases allow to save 25-28% of energy using the Mobile Core i7 system and 16-29% of energy using the Desktop Core i7 system. Of course, as we expected, with gain on energy savings we increase execution time. However, the time loss is much higher than the energy savings. For Mobile Core i7 system execution time increased by 59-86% and for the Desktop Core i7 system the time increase was in range of 38-80%. However, while considering only minimal energy points the time loss might suggest that power limiting is unreasonable, other low energy points with much better performance can be found. If we consider execution time against power limit we can observe that time grows non-linearly with a linear decrease of power limit and the energy consumption has several points besides the minimum in a region below e.g. 20% of energy saved. In such a situation we can search for much better performance with a power limit slightly higher and energy savings level similar to the best possible result. An exemplary illustration of the proposed approach could be seen in the results of FFT tests in Figure 4 (top left) where the minimum of energy was obtained for the power limit 25W but for the 30W limit we are able to obtain as good energy savings as in the best case (around 24%) but the time loss drops from 79% to 50%.

D. Conclusions

The results of experiments with limiting the power we proposed and executed on selected testbed systems can lead to several conclusions. First of all, the RAPL driver is able to limit the average power consumption for each of testbed systems and the systems respect the enforced power limit when set between minimal and maximal value. In the experiments we focused on lowering the power consumption and measuring the performance (execution time) and energy consumption during test application runs. We selected the most energy efficient power limit settings and compared the results with the

reference values with non-limited (reference) test runs. Table II collects the aforementioned data and correlates them with testbed systems and testbed applications.

The data collected together allows not only for answering the concerns that was a goal of this article but it is possible to compare the performance of the systems and energy efficiency between each other as well. For the testbed applications selected by us for the experiments the best performing system for 2 out of 3 applications was Xeon Phi x200. On the other hand the most energy efficient system was also a server dedicated processor but Xeon E5 v4. Both server systems showed that for such testbed applications the power consumption limiting gives no or insignificant results of energy saving.

The other pair of testbed systems based on Mobile and Desktop Core i7 processors proved that power consumption limiting can result in significant energy savings but, what is expected, we have to take the loss of performance into account. For the most energy efficient settings which offer between 16% and 29% of energy savings the performance loss is between 38% and 86%. One more conclusion when looking at the power utilisation for the Mobile and Desktop Core i7 systems is that when the testbed application is able to make use of more available power when no limits are set, the better are results of lowering the energy consumption. We can assume that if we had another testbed application that would be able to exploit more of the TDP of our server testbed systems we could probably observe better energy saving results.

V. FINAL REMARKS AND FUTURE WORK

The paper presented the experiments measuring the electrical energy consumption under a set of power caps for three representative HPC applications and four different processors. For some of CPU-application pairs the result analysis shows the existence of energy minima where the power capping provides significant savings — up to 28.8% for Desktop Core i7 executing the Heat Distribution simulation (see Table II for more details).

The future works are going to cover the following issues:

- analysis of the trade-off to find out potential points where values for measures incorporating execution time and energy used would be optimal for a specific application,
- benchmarking other applications, especially those that take more power from our testbed systems,

TABLE II
SUMMARY OF RESULTS PRESENTING MINIMAL ENERGY CASE FOR EACH EXPERIMENT.

		Fast Fourier Transform				Testbed application Heat Distribution				Numerical Integrate			
		E [J]	P_{lim} [W]	P_{av} [W]	t [s]	E [J]	P_{lim} [W]	P_{av} [W]	t [s]	E [J]	P_{lim} [W]	P_{av} [W]	t [s]
Xeon E5 v4	Reference	694.4	170.0	64.1	21.7	1282.1	170.0	66.3	38.7	703.1	170.0	55.8	25.2
	Best case	674.7	60.0	52.1	25.9	1282.1	170.0	66.3	38.7	703.1	170.0	55.8	25.2
	Difference	-2.8%	-64.7%	-18.6%	19.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
Xeon Phi x200	Reference	1266.0	215.0	149.1	8.5	4623.4	215.0	158.3	29.2	4,605.9	215.0	125.9	36.6
	Best case	1257.2	140.0	130.4	9.6	4482.6	140.0	132.4	33.9	4,605.9	215.0	125.9	36.6
	Difference	-0.7%	-34.9%	-12.5%	13.5%	-3.0%	-34.9%	-16.3%	15.8%	0.0%	0.0%	0.0%	0.0%
Mobile Core i7	Reference	966.5	15.0	14.8	65.3	2008.5	15.0	14.9	134.9	999.2	15.0	12.6	79.4
	Best case	700.5	6.0	6.0	117.4	1502.2	7.0	7.0	215.7	730.6	5.0	5.0	147.0
	Difference	-27.5%	-60.0%	-59.7%	79.9%	-25.2%	-53.3%	-53.2%	59.9%	-26.9%	-66.7%	-60.5%	85.1%
Desktop Core i7	Reference	1119.6	65.0	59.2	18.9	2616.1	65.0	58.2	44.9	2,313.9	65.0	43.8	52.9
	Best case	847.2	25.0	25.0	33.9	1863.8	30.0	29.9	62.2	1,931.4	25.0	25.0	77.4
	Difference	-24.3%	-61.5%	-57.8%	79.5%	-28.8%	-53.8%	-48.6%	38.5%	-16.5%	-61.5%	-43.0%	46.4%

- power-aware modeling of compute devices in frameworks for simulation of application runs in high performance computing environments such as MERPSYS [23],
- development of a tool for automatic detection of the optimal power settings for the aforementioned time-energy measures using historical data (e.g. via machine learning),
- proposing a new method for minimizing the electrical energy usage dynamically at runtime for various HPC/cloud workloads [24].

We assume that the expectations of the IT industry will generate a high demand for green computing methods used for exchanging time of computations into savings in the energy consumption (e.g. dedicated for off-pick hours of data centers). Thus, we hope that our work will stimulate even more research on the subject.

REFERENCES

- [1] M. Avgerinou, P. Bertoldi, and L. Castellazzi, "Trends in data centre energy consumption under the european code of conduct for data centre energy efficiency," *Energies*, vol. 10, no. 10, 2017. doi: 10.3390/en10101470. [Online]. Available: <http://www.mdpi.com/1996-1073/10/10/1470>
- [2] H. Krawczyk, M. Nykiel, and J. Proficz, "Mobile offloading framework: Solution for optimizing mobile applications using cloud computing," in *Computer Networks*, P. Gaj, A. Kwiecień, and P. Stera, Eds. Cham: Springer International Publishing, 2015. ISBN 978-3-319-19419-6 pp. 293–305.
- [3] K. N. Khan, M. Hirki, T. Niemi, J. K. Nurminen, and Z. Ou, "RapI in action: Experiences in using rapI for power measurements," *ACM Trans. Model. Perform. Eval. Comput. Syst.*, vol. 3, no. 2, pp. 9:1–9:26, Mar. 2018. doi: 10.1145/3177754. [Online]. Available: <http://doi.acm.org/10.1145/3177754>
- [4] B. Subramaniam and W. Feng, "Towards energy-proportional computing using subsystem-level power management," *CoRR*, vol. abs/1501.02724, 2015. [Online]. Available: <http://arxiv.org/abs/1501.02724>
- [5] C. Jin, B. R. de Supinski, D. Abramson, H. Poxon, L. DeRose, M. N. Dinh, M. Endrei, and E. R. Jessup, "A survey on software methods to improve the energy efficiency of parallel computing," *The International Journal of High Performance Computing Applications*, vol. 31, no. 6, pp. 517–549, 2017. doi: 10.1177/1094342016665471. [Online]. Available: <https://doi.org/10.1177/1094342016665471>
- [6] P. Czarnul, J. Kuchta, P. Rosciszewski, and J. Proficz, "Modeling energy consumption of parallel applications," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2016, pp. 855–864.
- [7] J. Proficz and P. Czarnul, "Performance and Power-Aware Modeling of MPI Applications for Cluster Computing," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9574, pp. 199–209. ISBN 9783319321516. [Online]. Available: http://link.springer.com/10.1007/978-3-319-32152-3_19
- [8] D. Abdurachmanov, P. Elmer, G. Eulisse, R. Knight, T. Niemi, J. K. Nurminen, F. Nyback, G. Pestana, Z. Ou, and K. Khan, "Techniques and tools for measuring energy efficiency of scientific software applications," *Journal of Physics: Conference Series*, vol. 608, no. 1, p. 012032, 2015. [Online]. Available: <http://stacks.iop.org/1742-6596/608/i=1/a=012032>
- [9] H. David, E. Gorbato, U. R. Hanebutte, R. Khanna, and C. Le, "RapI: Memory power estimation and capping," in *Proceedings of the 16th ACM/IEEE International Symposium on Low Power Electronics and Design*, ser. ISLPED '10. New York, NY, USA: ACM, 2010. doi: 10.1145/1840845.1840883. ISBN 978-1-4503-0146-6 pp. 189–194. [Online]. Available: <http://doi.acm.org/10.1145/1840845.1840883>
- [10] S. Desrochers, C. Paradis, and V. M. Weaver, "A validation of dram rapI power measurements," in *Proceedings of the Second International Symposium on Memory Systems*, ser. MEMSYS '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2989081.2989088. ISBN 978-1-4503-4305-3 pp. 455–470. [Online]. Available: <http://doi.acm.org/10.1145/2989081.2989088>
- [11] A. Mazouz, B. Pradelle, and W. Jalby, "Statistical validation methodology of cpu power probes," in *Revised Selected Papers, Part I, of the Euro-Par 2014 International Workshops on Parallel Processing - Volume 8805*. New York, NY, USA: Springer-Verlag New York, Inc., 2014. doi: 10.1007/978-3-319-14325-5_42. ISBN 978-3-319-14324-8 pp. 487–498. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-14325-5_42
- [12] M. Hirki, "Energy and performance profiling of scientific computing: tieteellisen laskennan energia- ja suorituskykyprofilointi," G2 Pro gradu, diplomityö, 2015. [Online]. Available: <http://urn.fi/URN:NBN:fi:aalto-201512165699>
- [13] M. Hirki, Z. Ou, K. N. Khan, J. K. Nurminen, and T. Niemi, "Empirical study of the power consumption of the x86-64 instruction decoder," in *USENIX Workshop on Cool Topics on Sustainable Data Centers (CoolDC 16)*. Santa Clara, CA: USENIX Association, 2016. [Online]. Available: <https://www.usenix.org/conference/cooldc16/workshop-program/presentation/hirki>
- [14] H. Zhang and H. Hoffmann, "Maximizing performance under a power cap: A comparison of hardware, software, and hybrid techniques," in *Proceedings of the Twenty-First International Conference on*

- Architectural Support for Programming Languages and Operating Systems*, ser. ASPLOS '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2872362.2872375. ISBN 978-1-4503-4091-5 pp. 545–559. [Online]. Available: <http://doi.acm.org/10.1145/2872362.2872375>
- [15] F. Sun, H. Li, Y. Han, G. Yan, and J. Ma, “Powercap: Leverage performance-equivalent resource configurations for power capping,” in *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*, Nov 2016. doi: 10.1109/IGCC.2016.7892618 pp. 1–8.
- [16] Q. Zhu, B. Wu, X. Shen, L. Shen, and Z. Wang, “Co-run scheduling with power cap on integrated cpu-gpu systems,” in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2017. doi: 10.1109/IPDPS.2017.124 pp. 967–977.
- [17] M. Travers, “Cpu power consumption experiments and results analysis of intel i7-4820k,” uSystems Research Group, School of Electrical and Electronic Engineering, Newcastle University, UK, Tech. Rep. NCL-EEE-MICRO-TR-2015-197, 2015, <http://async.org.uk/tech-reports/NCL-EEE-MICRO-TR-2015-197.pdf>.
- [18] K. Pedretti, S. L. Olivier, K. B. Ferreira, G. Shipman, and W. Shu, “Early experiences with node-level power capping on the cray xc40 platform,” in *Proceedings of the 3rd International Workshop on Energy Efficient Supercomputing*, ser. E2SC '15. New York, NY, USA: ACM, 2015. doi: 10.1145/2834800.2834801. ISBN 978-1-4503-3994-0 pp. 1:1–1:10. [Online]. Available: <http://doi.acm.org/10.1145/2834800.2834801>
- [19] A. Krzywaniak and P. Czarnul, “Parallelization of selected algorithms on multi-core cpus, a cluster and in a hybrid cpu+xeon phi environment,” in *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology - ISAT 2017 - Part I, Szklarska Poreba, Poland, September 17-19, 2017*, ser. Advances in Intelligent Systems and Computing, L. Borzemski, J. Swiatek, and Z. Wilimowska, Eds., vol. 655. Springer, 2017. doi: 10.1007/978-3-319-67220-5_27. ISBN 978-3-319-67219-9 pp. 292–301. [Online]. Available: https://doi.org/10.1007/978-3-319-67220-5_27
- [20] “OpenMP home,” URL: <https://www.openmp.org/>, accessed: 2018-05-11.
- [21] J. Sanders and E. Kandrot, *CUDA by Example: An Introduction to General-Purpose GPU Programming*, 1st ed. Addison-Wesley Professional, 2010. ISBN 0131387685, 9780131387683
- [22] M. Balducci, A. Choudary, and J. Hamaker, “Comparative analysis of FFT algorithms in sequential and parallel form,” Tech. Rep., 1996.
- [23] P. Czarnul, J. Kuchta, M. Matuszek, J. Proficz, P. Rosciszewski, M. Wojcik, and J. Szymanski, “Merpsys: An environment for simulation of parallel application execution on large scale hpc systems,” *Simulation Modelling Practice and Theory*, vol. 77, pp. 124 – 140, 2017. doi: <https://doi.org/10.1016/j.simpat.2017.05.009>. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1569190X17300916>
- [24] P. Orzechowski, J. Proficz, H. Krawczyk, and J. Szymanski, “Categorization of cloud workload types with clustering,” in *Proceedings of the International Conference on Signal, Networks, Computing, and Systems*, D. K. Lobiyal, D. P. Mohapatra, A. Nagar, and M. N. Sahoo, Eds. New Delhi: Springer India, 2017. ISBN 978-81-322-3592-7 pp. 303–313.

