Parallel Computing: Technology Trends
I. Foster et al. (Eds.)

doi:10.3233/APC200093

Evaluation of DVFS and Uncore Frequency Tuning Under Power Capping on Intel Broadwell Architecture

Lubomir RIHA [a,1], Ondrej VYSOCKY [a] and Andrea BARTOLINI [b]
[a] *IT4Innovations national supercomputing center,*
VŠB – Technical University of Ostrava, Ostrava, Czech Republic
[b] *University of Bologna, DEI, via Risorgimento 2, 40136 Bologna, Italy*

Abstract. In this paper we present an evaluation of the Intel Xeon Broadwell platform in the CINECA Galileo supercomputer when DVFS and UnCore Frequency (UCF) tuning is performed under the active power capping using RAPL powercap registers. This work is an extension of our previous work done under the H2020 READEX project which focused on a dynamic tuning of DVFS and UCF for complex HPC applications, but with no powercap limit enforced. Power capping is an essential technique that allows system administrators to maintain the power budget of an entire system or data center using either out-of-band management system or runtime systems such as GEOPM.

In this paper we use two boundary workloads, Compute Bound Workload (CBW) and Memory Bound Workload (MBW) to show the behavior of the platform under power capping and potential for both energy and runtime savings when compared to the default CPU behavior. We show that DVFS and UCF tuning behave differently under the limited power budget. Our results show that if CPU has a limited power budget the proper tuning can provide both improved energy consumption as well as reduced runtime and that it is important to tune both DVFS and UCF.

For MBW we can save up 22 % for both runtime and energy when compared to default behavior under powercap. For CBW we can improve both performance, up to 9.4 %, and energy consumption, up to 14.9 %.

1. Introduction

Energy and power consumption become limiting parameters of new peta- and exa-scale HPC clusters. Due to that accelerators are more common hardware used to provide the performance of the system [1]. Nevertheless it is not only hardware but also software and runtime systems that must be improved to reduce energy and power clusters' hungriness to stay below the 20 MW limit that is being considered as a peak power for an HPC system [2,3].

Energy savings given by software tuning come from better utilization of the hardware resources. It is up to the developers to improve performance of their application,

[1] Corresponding Author: IT4Innovations National Supercomputing Center, VSB-Technical University of Ostrava, 17. listopadu 15/2172, 708 00 Ostrava - Poruba, Czech Republic; E-mail: lubomir.riha@vsb.cz

or apply one of many approaches that limit the resources to the level, that the application does not waste the resources. Typically CPU core frequency is being reduced (also known as Dynamic Voltage and Frequency Scaling, DVFS) for this purpose. In several researches the DVFS is usually set to one specific frequency. Fraternali et al. [4] study the impact of DVFS and HW/SW variability in heterogeneous workloads. Bonati et al. [5] focuses on evaluation this trade-off in a multi-node multi-accelerator context. Calore et al. [6] also evaluate the effect of DVFS on modern HPC processors and accelerators. This approach is efficient in case of single-purpose kernels, however it does not work well when a complex application is tuned.

This work is an extension of our previous effort done under the H2020 READEX project [7,8] which was focused not only on a tuning of CPU core frequency but also its uncore frequency. CPU uncore frequency (UCF) refers to frequency of subsystems in the physical processor package that are shared by multiple processor cores e.g., L3 cache or on-chip ring interconnect. READEX has developed an open-source runtime system called READEX Runtime Library (RRL) that performs dynamic tuning of hardware parameters during a complex parallel applications run, based on Score-P [9] regions instrumentation. RRL uses a configuration file created during the analysis of an application and applies the optimal settings for different parts of the code. RRL supports both DVFS and UCF tuning and also a concurrency throttling, however with no power cap limit enforced.

Power capping is an essential technique that allows system administrators to maintain the power budget of an entire system or data center using either out-of-band management system or runtime systems such as GEOPM [10]. This runtime system in addition to capping CPU's package power consumption also may tune the CPU's core frequencies, but it does not control uncore frequency of the chip. This paper shows that adding a support for UCF tuning will have significant impact on both performance and energy consumption. In [11] Zhang et.al. presents an approach for maximizing the performance under powercap by tuning the DVFS, number of cores, hyper-threads and potentially number of sockets, however also in this research the UCF tuning is not presented.

The proposed method is implemented into our open-source library MERIC [12], that has been also developed under the READEX project.

2. Methodology

2.1. Experiments description

We have conducted a set of experiments that is defined in Table 1. This set covers all the possibilities: (1) pure CPU firmware automatic tuning of all parameters - EXP0; (2) combination of user and firmware tuning - EXP1 to EXP6 and (3) pure user tuning of all three parameters - EXP7. The goal is to find out in which cases user tuning can help and when it can harm the performance or energy consumption.

For each of the experiments we have run a compute bound workload (CBW) and memory bound workload (MBW) to evaluate the behavior of the Intel RAPL power capping system [13] in two situations. When high core frequency is more important the uncore frequency can be reduced without major performance penalty, which is the case of the CBW. The exactly opposite situation is for the MBW. As a compute bound

Experiment number	Powercapping	DVFS (core freq. tuning)	Uncore freq. tuning	Description
0	-	-	-	default CPU behavior (powersave scaling governor)
1	x	-	-	default CPU behavior under powercap
2	-	x	-	default CPU behavior under DVFS tuning
3	-	-	x	default CPU behavior under uncore freq. tuning
4	-	x	x	READEX tuning approach - DVFS & uncore freq.
5	x	x	-	DVFS tuning under powercap; uncore freq. unset
6	x	-	x	uncore freq. under powercap; DVFS unset
7	x	x	x	DVFS and uncore freq. tuning under powercap

Table 1. A set of experiments performed on the platform to determine its behavior.

region we have selected a loop of tangents (TAN) operation and memory bound region is represented by a loop with a matrix vector multiplication (DGEMV).

2.2. Hardware Platform Description and Measurement Setup

The evaluation was done on the Broadwell partition of the Galileo supercomputer installed in CINECA [14]. The servers in this partition are dual socket machines equipped with two 18-core Intel Xeon E5-2697v4 processor [15] running at 2.3 GHz nominal frequency. The turbo frequency when all 18 cores are utilized is 2.7 GHz. This was verified by our measurements. The TDP of the processor is 145 W. Further details are shown in Table 2 including the ranges of all tunable parameters and their granularity.

	nominal value	minimal value	maximal value	minimal step
CPU core frequency (DVFS)	2.3 GHz	1.2 GHz	2.8 GHz turbo	100 MHz
CPU uncore frequency	-	1.2 GHz	2.8 GHz	100 MHz
Power capping	145 W[2]	33 W	145 W	0.125 W

Table 2. Key tunable parameters of the 18-core Intel Xeon E5-2697v4 processor and their respective ranges and steps.

All test were performed in a way that the workload was executed on socket 1, while socket 0 was not utilized. This way we were able significantly reduce the effect of the system noise on the measurements. Also all measurements were repeated ten times and outliers were eliminated using interquartile range rule[3].

3. Results

3.1. DVFS and UCF Tuning without Powercap

It this section we will evaluate the behavior of the platform without enforcing the power cap.

[2] TDP value of the E5-2697v4 processor.

[3] see: https://www.mathwords.com/o/outlier.htm

The key behavior of the platform when running CBW is: the DVFS has key impact on performance/runtime; the uncore frequency has no effect on performance/runtime, it can only affect the power and therefore energy consumption.

On the other hand the key behavior of the MBW is: the uncore frequency has major impact on performance/runtime; the CPU core frequency has no effect on performance/runtime. It can only affect the power and therefore energy consumption.

The Figure 1 presents runtime and energy consumption for both CBW and MBW in EXP2 and EXP4 configuration. The key observations for the compute bound workload are:

- The energy consumption (full red line) significantly increases from 305 J to 358 J (by 17.4 %) when core frequency increases from 2.2 GHz to 2.3 GHz (the nominal frequency) while runtime decreases by only 4.2 %.
- At this point the CPU switches to the highest available uncore frequency, which is confirmed by the test that runs at maximum uncore frequency (red dashed line).
- One can further reduce the energy consumption by reducing the UCF to minimum value, see the red doted line. In this case the energy consumption is reduced from 358 J to 284 J (by 26 %) for nominal frequency (2.3 GHz).
- The same behavior remains when CPU runs at turbo frequency (2.7 GHz), in this case the energy consumption is reduced from 354 J to 299 J (by 18.4 %) when UCF is set minimum. For CBW this is the optimal point from both energy and performance point of view.

The key observations from Figure 1 for memory bound workload are:

- By default CPU runs at high uncore frequency in the entire range of CPU core frequencies.
- Reducing the uncore frequency to low values increases both runtime and energy consumption.

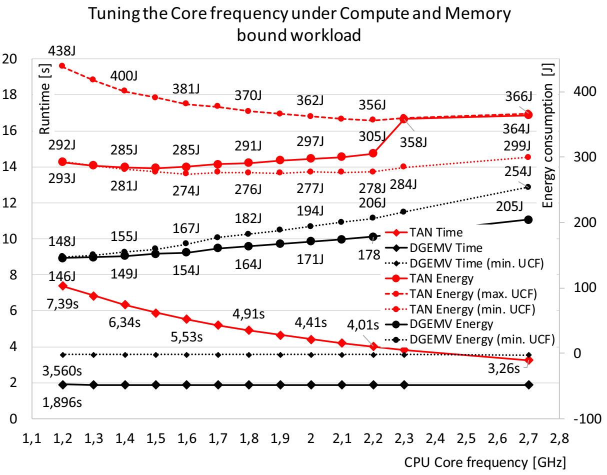

Figure 1. The behavior of the platform for the DVFS tuning for compute bound and memory bound workloads. The solid lines show default behavior without UCF tuning, the dashed lines show the behavior for maximum UCF and the doted lines show the behavior for minimum UCF.

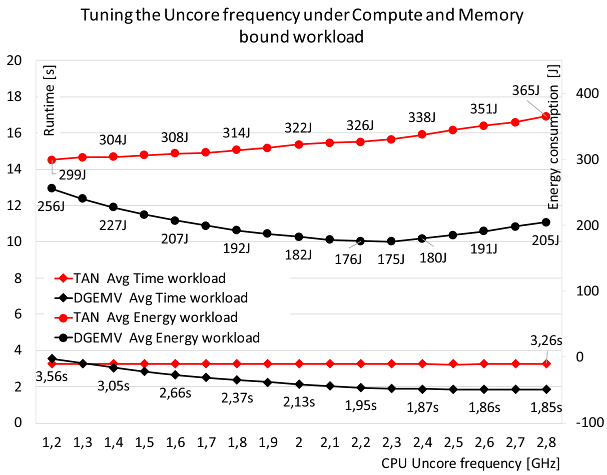

Figure 2. The behavior of the platform under the uncore frequency tuning for compute bound and memory bound workload.

Figure 2 shows the behavior of the platform for UCF tuning, the EXP3. We can see that, as expected, for CBW the uncore frequency has no effect on performance (runtime remains the same in the entire range, while energy consumption grows with higher UCF.

On the other hand, for the MBW the optimal performance requires high UCF. From energy point of view the optimal frequency is 2.3 GHz. If one increase the UCF to 2.8 GHz the gain is only 2.1 % higher performance at a cost of additional 14.6 % of energy.

3.2. DVFS and UCF Tuning under Powercap for Memory Bound Workload

Figure 3 shows the behavior of the platform when running memory bound workload under three different power cap levels: 100 W, 80 W and 60 W.

The default behavior of the CPU without powercap is represented by the EXP0 results: 1.88 sec runtime; 197 J energy consumption. In terms of runtime, this represents the maximum achievable performance.

For all three powercap levels EXP1 results presents the default behavior of the CPU under the powercap. These values are the baselines for all further experiments and are as follows: for 100 W it is 1.88 sec and 188.2 J; for 80 W it is 1.92 sec and 153.2 J; and for 60 W it is 2.47 sec and 147.8 J.

In the previous section where no power limit was set we have observed that for memory bound workload, tuning the DVFS does not affect the performance, but has a significant impact on energy consumption. The results of EXP5 for 100 W powercap level still hold this behavior. The runtime remains 1.88 sec while energy consumption is reduced from 188.2 J to 148.6 J when CPU core frequency is reduced from turbo frequency (2.7 GHz) to its minimal value 1.2 GHz. However, the expected behavior is no longer true for the 80 W powercap level. In this case the energy consumption remains similar and it is only slightly reduced from 153.2 J to 146.0 J. The runtime is also only slightly reduced from 1.92 sec to 1.89 sec. The most visible effect on both performance and energy consumption has the DVFS tuning under the 60 W powercap. In this case

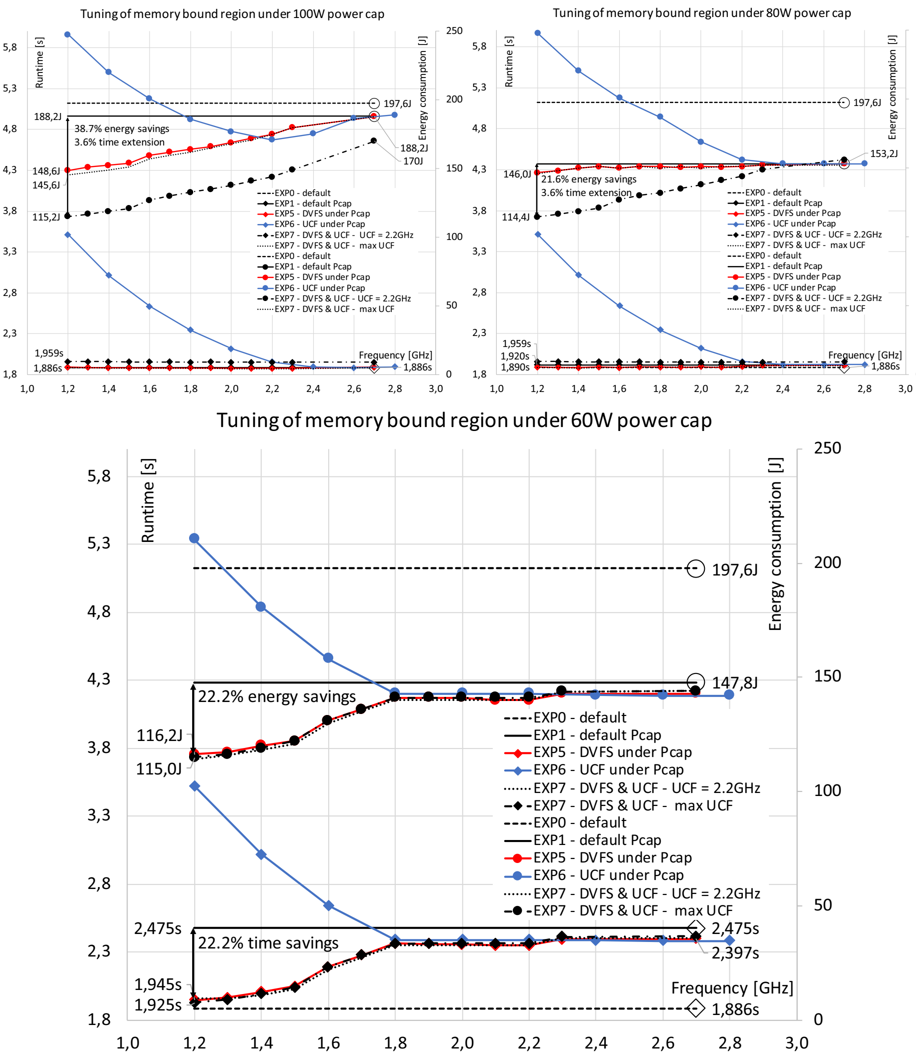

Figure 3. The behavior of the platform running memory bound workload (GEMV) for (EXP1) default CPU behavior under powercap, (EXP5) DVFS tuning under powercap, (EXP6) UCF tuning under powercap and (EXP7) DVFS and UCF tuning under powercap. All tests are done for 60, 80 and 100 W powercap levels.

both runtime and energy are reduced by 22.2 % when core frequency is set to its minimal value (1.2 GHz). We explain this behavior as follows: by limiting the performance and as a consequence the power consumption of CPU cores, the uncore part of the chip responsible for communication with memory gets higher power budget and it can run on higher frequency and achieve higher performance. Therefore CPU executes the MBW more efficiently. Under such very limited power budget (60 W) this makes the significant difference against the default CPU behavior.

Results of EXP6 shows that tuning the UCF has a significant impact on performance and it should be kept as high as possible. In terms of energy consumption the optimal setting is 2.2 GHz (it is the most visible in EXP6 results for 100 W powercap). However the key observation is that tuning *ONLY* the UCF for MBW has small impact as by default CPU keeps it high enough.

Finally, the results of EXP7 show that adding the UCF tuning to DVFS tuning has a significant impact on energy consumption for higher power cap (100 W and 80 W). For 100 W powercap CPU saves up to 38.7 % of energy while increases the runtime by 3.6 % only. For 80 W powercap CPU saves 21.6 % of energy with the same time penalty (3.6 %). For 60 W powercap both energy consumption and runtime are almost identical to the DVFS tuning only (EXP5).

3.3. DVFS and UCF Tuning under Powercap for Compute Bound Workload

Figure 4 shows the behavior of the platform when running CBW under three different power cap levels: 100 W, 80 W and 60 W. The default behavior of the CPU without powercap is represented by the EXP0 results: 3.27 sec runtime; 363.4 J energy consumption. In terms of runtime, this represents the maximum achievable performance.

For all three powercap levels EXP1 results presents the default behavior of the CPU under the powercap. These values are the baselines for all further experiments and are as follows: for 100 W it is 3.45 sec and 344.4 J; for 80 W it is 3.90 sec and 311.8 J; and for 60 W it is 4.94 sec and 296,0 J.

When compared to MBW results we can see that CPU requires more power to execute CBW. By reducing the powercap from 140 W (TDP level) to 100 W the performance is reduced by 5.2 %. The 80 W powercap reduces performance by 16.2 % and the 60 W power reduce performance by 33.8 %.

For a compute bound workload DVFS tuning is a key knob to control the performance for all powercap levels. Any energy savings gained by the DVFS tuning are paid by significant performance penalty. However if energy savings are needed this knob has the highest impact for higher power budgets. If power budget gets lower the UCF tuning gains on importance.

The key findings comes from EXP6 for tuning the UCF frequency. For 100 W powercap level by reducing the UCF to 2.2 GHz or bellow we improve the performance by 4.5 % over the default level, from 3.45 s to 3.29 s. If one further reduces the the UCF to its minimum value, 1.2 GHz the performance remains the same but energy consumption is improved by 14.9 % against the default powercap behavior (EXP1). For 80 W powercap level since the CPU is already struggling with the limited power budget the performance increase is visible in the entire range of UCF going from max. to min. value. The same holds for energy consumption. Both the best performance and the lowest energy consumption is achieved at 1.2 GHz (minimal) UCF frequency. In this case the performance is improved by 8.4 % and energy consumption by 8.5 % against the default behavior under powercap (EXP1). Also, the performance is only 8.6 % lower against non powercapped CPU (EXP0), without UCF tuning this penalty was 16.2 %. For 60 W powercap the CPU behavior is similar to 80 W powercap. The best performance is achieved at minimal uncore frequency, and for this case the performance is increased by 9.4 % and energy consumption is reduced by 9.1 %. Against the default CPU behavior without powercap (EXP0) the performance penalty is reduced from 33.8 % to 27.0 %.

Finally, the results of EXP7 show again that adding the UCF tuning to DVFS tuning has a significant impact on energy consumption. It is the most visible for the 100 W and 80 W powercap experiments. But under all powercap levels the minimum energy con-

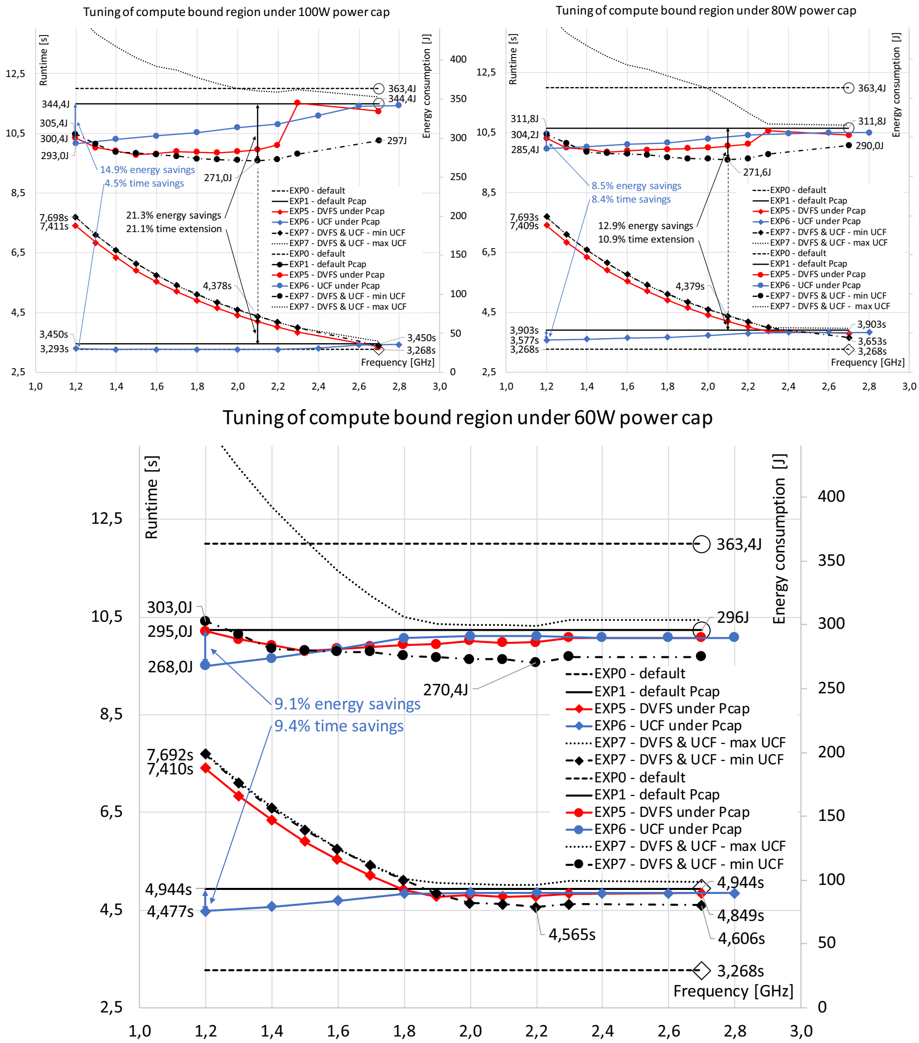

Figure 4. The behavior of the platform running compute bound workload (TAN) for (EXP1) default CPU behavior under powercap, (EXP5) DVFS tuning under powercap, (EXP6) UCF tuning under powercap and (EXP7) DVFS and UCF tuning under powercap. All tests are done for 60, 80 and 100 W powercap levels.

sumption, within a very small margin, approximately 271 J is achieved. This is achieved for minimal uncore frequency and 2.1 GHz core frequency.

However it is important to note, that by tuning both core and uncore CPU frequencies the performance gained by tuning the UCF only was not met. For 100 W UCF only tuning is 3.8 % faster, for 80 W it is 2.0 % faster, and for 60 W it is 1.9 % faster.

To summarize the numbers for EXP7: for 100 W by DVFS we can save 21.3 % energy at 21.1 % runtime penalty; for 80 W by DVFS we can save 12.9 % energy at 10.9 % runtime penalty; and for 60 W by DVFS we can save 8.6 % and at 7.7 % runtime penalty. All against the default CPU behavior under the same level of powercap (EXP1).

4. Conclusion - Summary of Observations and Best Practises

The Intel RAPL power capping system guarantees that the CPU keeps its energy consumption in a specified time window under a power boundary. We present how the system reduces both CPU core and uncore frequencies to reach this constraint. Since the system does not identify the kind of the workload running on the chip, it leads to the situation that core frequency is reduced while uncore frequency is still inefficiently too high for the given workload running or vice versa. Manually forcing a CPU configuration, DVFS or UCF, does not mean that the configuration will be applied if it infringes the power cap limit given to RAPL. However, manual reduction of one of the frequencies opens the availability to tune the other one to higher frequencies as it enables power bugdet shifting from one part of chip to the other one.

We have identified the optimal configuration of the CPU frequencies to reach the minimal energy consumption of the two workloads. When the powercap is applied, the CPU frequencies are reduced accordingly but not efficiently, due to that our manual frequency tuning leads to both time and energy savings.

To conclude, the results show that for MBW the proposed tuning can achieve:

- Under the power budget lower that 80 W settings the DVFS to minimum value boost the performance of the uncore part by 22 %.
- In addition to DVFS tuning the uncore frequency has low effect on the performance but a major one on energy consumption (between 21 % to 38 %).

The results show that for CBW the proposed tuning can achieve:

- To achieve the best possible performance it is key to reduce the UCF to minimum level. This way *BOTH* performance (up to 9.4 %) and energy consumption (up to 14.9 %) are improved.
- If further energy savings are required (up to 21 %) it can be achieved by DVFS tuning by lowering the core frequency. This comes at penalty in runtime (up to 21 %). This effect is more visible for higher powercap levels.

In the future work we would like to extend our measurements with benchmark, that can set vary arithmetic intensity on a fine grain (instruction) level and evaluate new CPU architectures code-named Skylake and Cascade Lake.

Acknowledgement

This work was supported by The Ministry of Education, Youth and Sports from the Large Infrastructures for Research, Experimental Development and Innovations project IT4Innovations National Supercomputing Center LM2015070.

The work has been performed under the Project HPC-EUROPA3 (INFRAIA-2016-1-730897), with the support of the EC Research Innovation Action under the H2020 Programme.

This work was supported by the Moravian-Silesian Region from the programme ”Support of science and research in the Moravian-Silesian Region 2017” (RRC/10-/2017).

This work was also partially supported by the SGC grant No. SP2019/59 ”Infrastructure research and development of HPC libraries and tools”, VŠB - Technical University of Ostrava, Czech Republic.

References

[1] E. Strohmaier, “Highlights of the 51st top500 list,” 2018.

[2] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Karp, S. Keckler, D. Klein, R. Lucas, M. Richards, A. Scarpelli, S. Scott, A. Snavely, T. Sterling, R. S. Williams, K. Yelick, K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, J. Hiller, S. Keckler, D. Klein, P. Kogge, R. S. Williams, and K. Yelick, “Exascale computing study: Technology challenges in achieving exascale systems,” 2008.

[3] V. Sarkar, W. Harrod, and A. E. Snavely, “Software challenges in extreme scale systems,” *Journal of Physics: Conference Series*, vol. 180, p. 012045, jul 2009.

[4] F. Fraternali, A. Bartolini, C. Cavazzoni, and L. Benini, “Quantifying the Impact of Variability and Heterogeneity on the Energy Efficiency for a Next-Generation Ultra-Green Supercomputer,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 29, pp. 1575–1588, July 2018.

[5] C. Bonati, E. Calore, M. DElia, M. Mesiti, F. Negro, S. F. Schifano, G. Silvi, and R. Tripiccione, “Early Experience on Running OpenStaPLE on DAVIDE,” in *High Performance Computing* (R. Yokota, M. Weiland, J. Shalf, and S. Alam, eds.), Lecture Notes in Computer Science, pp. 387–401, Springer International Publishing, 2018.

[6] E. Calore, A. Gabbana, S. F. Schifano, and R. Tripiccione, “Evaluation of dvfs techniques on modern hpc processors and accelerators for energy-aware applications,” *Concurrency and Computation: Practice and Experience*, vol. 29, no. 12, p. e4143, 2017. e4143 cpe.4143.

[7] READEX, “Horizon 2020 READEX project.” https://www.readex.eu, 2018.

[8] J. Schuchart, M. Gerndt, P. G. Kjeldsberg, M. Lysaght, D. Horák, L. Říha, A. Gocht, M. Sourouri, M. Kumaraswamy, A. Chowdhury, M. Jahre, K. Diethelm, O. Bouizi, U. S. Mian, J. Kružík, R. Sojka, M. Beseda, V. Kannan, Z. Bendifallah, D. Hackenberg, and W. E. Nagel, “The readex formalism for automatic tuning for energy efficiency,” *Computing*, vol. 99, pp. 727–745, Aug 2017.

[9] A. Knüpfer, C. Rössel, D. a. Mey, S. Biersdorff, K. Diethelm, D. Eschweiler, M. Geimer, M. Gerndt, D. Lorenz, A. Malony, W. E. Nagel, Y. Oleynik, P. Philippen, P. Saviankou, D. Schmidl, S. Shende, R. Tschüter, M. Wagner, B. Wesarg, and F. Wolf, “Score-p: A joint performance measurement run-time infrastructure for periscope,scalasca, tau, and vampir,” in *Tools for High Performance Computing 2011* (H. Brunst, M. S. Müller, W. E. Nagel, and M. M. Resch, eds.), (Berlin, Heidelberg), pp. 79–91, Springer Berlin Heidelberg, 2012.

[10] J. Eastep, S. Sylvester, C. Cantalupo, B. Geltz, F. Ardanaz, A. Al-Rawi, K. Livingston, F. Keceli, M. Maiterth, and S. Jana, “Global extensible open power manager: A vehicle for hpc community collaboration on co-designed energy management solutions,” in *ISC*, (Cham), pp. 394–412, Springer International Publishing, 2017.

[11] 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*, ASPLOS ’16, (New York, NY, USA), pp. 545–559, ACM, 2016.

[12] O. Vysocky, M. Beseda, L. Riha, J. Zapletal, V. Nikl, M. Lysaght, and V. Kannan, “Evaluation of the HPC applications dynamic behavior in terms of energy consumption,” in *Proceedings of the Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, pp. 1–19, 2017. Paper 3, 2017. doi:10.4203/ccp.111.3.

[13] H. David, E. Gorbatov, U. R. Hanebutte, R. Khanna, and C. Le, “Rapl: Memory power estimation and capping,” in *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, pp. 189–194, Aug 2010.

[14] CINECA, “GALILEO User Guide.” https://wiki.u-gov.it/confluence/display/SCAIUS/UG3.3%3A+GALILEO+UserGuide, 2018.

[15] Intel corp., “Intel Xeon Processor E5-2697v4 Product Specification.” https://ark.intel.com/content/www/us/en/ark/products/91755/intel-xeon-processor-e5-2697-v4-45m-cache-2-30-ghz.html, 2019.