Hindawi
Scientific Programming
Volume 2021, Article ID 5514284, 9 pages
https://doi.org/10.1155/2021/5514284

Review Article

Hardware and Software Solutions for Energy-Efficient Computing in Scientific Programming

Daniele D'Agostino,[1] **Ivan Merelli**,[2] **Marco Aldinucci**,[3] **and Daniele Cesini**[4]

[1]*CNR-IEIIT, Genoa, Italy*
[2]*CNR-ITB, Segrate (MI), Italy*
[3]*University of Turin, Turin, Italy*
[4]*CNAF-Italian Institute for Nuclear Physics, Bologna, Italy*

Correspondence should be addressed to Daniele D'Agostino; daniele.dagostino@ge.imati.cnr.it

Received 25 January 2021; Accepted 28 May 2021; Published 9 June 2021

Academic Editor: Cristian Mateos

Energy consumption is one of the major issues in today's computer science, and an increasing number of scientific communities are interested in evaluating the tradeoff between time-to-solution and energy-to-solution. Despite, in the last two decades, computing which revolved around centralized computing infrastructures, such as supercomputing and data centers, the wide adoption of the Internet of Things (IoT) paradigm is currently inverting this trend due to the huge amount of data it generates, pushing computing power back to places where the data are generated—the so-called fog/edge computing. This shift towards a decentralized model requires an equivalent change in the software engineering paradigms, development environments, hardware tools, languages, and computation models for scientific programming because the local computational capabilities are typically limited and require a careful evaluation of power consumption. This paper aims to present how these concepts can be actually implemented in scientific software by presenting the state of the art of powerful, less power-hungry processors from one side and energy-aware tools and techniques from the other one.

1. Introduction

Information and communication technologies (ICT) play a fundamental role in supporting human activities for the global economic, social, and environmentally sustainable developments [1]. However, energy consumption is one of the most relevant issues for present computing platforms, and this trend is expected to continue in the foreseeable future. This implies that the electricity bill increasingly dominates costs related to the running of applications and the consequent environmental pollution [2].

This situation is evident for high-performance computing (HPC) infrastructures, where the sum of the energy bills over a supercomputer's lifetime is comparable to the acquisition cost and represents one of the most relevant elements of the total cost of ownership [3]. This is because energy is used not only for computation but also for cooling, communication, storage, and display [4].

The focus of performance-at-any-cost computer operations has led to the emergence of supercomputers that consume vast amounts of electrical power and produce so much heat in that extended cooling facilities must be constructed to ensure proper performance. The consequence is that, in the context of deploying an exascale system, the simple scaling of current technologies would result in a supercomputer with a power consumption of 100 MW, while a limit of 20 MW has been estimated as the maximum acceptable limit [5]. The attention to the flop-per-watt

performance has been demonstrated by the introduction, in 2007, of the Green500 List [6] that ranks the top 500 supercomputers by energy efficiency [7].

The same problem also arises in general-purpose data centers: in the US, such infrastructures consumed about 70 billion kWh in 2014, representing 1.8% of total US electricity consumption, as reported in [8]. Some projections estimate for 2020 an electricity demand that varies by about 135 billion kWh, depending on the adoption rate of efficiency measures [9].

This scenario must be combined because in the past two decades, computing has been focused around centralized (and possibly complex [10]) infrastructures, but the wider diffusion of cyber-physical systems (CPSs) is currently inverting this trend, pushing computing power back to where data are generated. In both cases, the energy consumption of telecommunication networks is very relevant [11]. A striking example of the trend is the Internet of Things (IoT) paradigm, by which millions of devices generate a huge amount of data that are pre-elaborated locally before being integrated remotely in a data analytics context. Nevertheless, also considering science, the diffusion of powerful data acquisition devices boosted the diffusion of pre-elaboration computational architectures, such as in bioinformatics [12, 13].

While HPC is a well-specific market sector, the so-called "embedded HPC" is an emerging topic [14] to develop and employ microservers/highly parallel embedded computing systems in the CPS. Therefore, the adoption of energy-efficient systems represents a crucial aspect considering the characteristics of fog/edge computing environments [15].

We can formulate the problem as the need to assess a satisfactory tradeoff between time-to-solution and energy-to-solution. This problem has been faced with different approaches, which can be summarised as follows: vendors work on power-efficient processor architectures and software developers on how to use them. However, to reach exascale computing, an effective solution is possible only by properly managing all layers of the system, from the software stack to the cooling system [16] passing by less power-hungry CPUs. This can be achieved by reducing the energy consumed in the total system via both power-efficient software and hardware integrated solutions [17, 18].

Energy efficiency is a key design challenge for modern computing systems for many years. Even more now, the Big Data paradigm requires addressing both issues related to the efficient processing of such an enormous amount of data and how to achieve this goal in a green way, i.e., considering issues related to sustainability and environmental concerns [19].

Therefore, many papers proposing novel techniques for managing power aspects and presenting real-world experiences, together with surveys and overviews, have been published. A critical analysis on how to greening the whole life cycle of big data systems is presented in [20]. On a more technical perspective, Czarnul et al. [21] focused on the available methods and tools allowing proper configuration, management, and simulation of HPC systems for energy-aware processing. An overview of application performance analysis tools, including the energetic profiling of an application and auto-tuning tools for energy saving, has been presented in [22]. The usage of low-power System-on-Chip (SoC) architectures for scientific (and industrial) applications is discussed in [23], intending to assess the tradeoff among time-to-solution, energy-to-solution, and economic aspects for both scientific and commercial purposes they can achieve in comparison to traditional server-grade architectures adopted in present infrastructures.

However, an issue is represented by the fact that nearly all the existing surveys focus on only one of the two main strategies, i.e.,

(i) The development and usage of new energy-efficient CPUs and SoCs

(ii) The use of software tools and frameworks for reducing the power consumption of software using an existing CPU

Moreover, as recognized by most of these papers, this is a rapidly evolving research field where new results are continuously presented. For example, at the time of writing, the following five European research projects and initiatives are ongoing:

(i) Mont-Blanc 2020, European scalable, modular, and power-efficient HPC processor

(ii) HiPEAC, High Performance and Embedded Architecture and Compilation

(iii) LEGaTO, Low-Energy Toolset for Heterogeneous Computing

(iv) SDK4ED, Software Development toolKit for Energy optimization and technical Debt elimination

(v) TeamPlay, Time, Energy and security Analysis for Multi/Many-core heterogeneous PLAtforms

This is because the European Commission has been aware since at least 2010 that the ICT sector is responsible for carbon emissions which are rapidly growing and should be kept to a minimum and therefore is supporting the development of more energy-efficient computing technologies.

Therefore, this work's main goal is to present the most relevant available solutions for users interested in improving the energy consumption of scientific software focusing on computation. This is achieved by investigating the availability and performance of current hardware devices and software tools for scientific applications.

This means that the aspects related to energy efficiency in communications are not considered here. Interested readers can rely on [24, 25].

The structure of the paper is as follows: Section 2 presents hardware techniques and solutions for achieving energy-savvy processing, Section 3 discusses tools and methodologies for supporting developers in producing energy-aware software, while the last section concludes the paper.

2. Energy-Efficient Architectures

2.1. General-Purpose Techniques. Firstly, let us review the techniques that exploit hardware characteristics to reduce energy consumption. Most of the present architectures, in fact, implement energy-saving techniques. They are based on the use of low-level electronic characteristics to run no faster than necessary at a voltage no higher than acceptable. They are

(i) Dynamic frequency scaling (DFS)
(ii) Dynamic voltage scaling (DVS)
(iii) Dynamic voltage and frequency scaling (DVFS)
(iv) Near-threshold voltage (NTV)
(v) Dynamic power management (DPM)

Dynamic frequency (DFS) or voltage (DVS) scaling allows to modulate the power consumption processor and memory [26], scaling the clock frequency of one or both subsystems according to the execution of memory- or compute-bound application kernels [27].

For example, voltage reduction has to be considered for the heterogeneous accelerators equipping current systems also because the efficient reduction of the total power can be achieved with different voltage reduction levels for each available chip [28].

Very often, voltage and frequency ranges are fully interdependent, i.e., a change in clock frequency does imply changes in the supply voltage, and vice versa: in these cases, the technique is called dynamic voltage and frequency scaling (DFVS) [29]. Specific hardware mechanisms can implement DVFS with minimal software and operating system involvement or through enabling software.

For example, DVFS is implemented in the Linux kernel with the CPUfreq subsystem [30, 31]. The original implementation of kernel 2.6 has been designed to be used when no real-time tasks are executed. However, it is possible to relax this constraint [32].

More recently, other projects focused on near-threshold voltage (NTV) computing [33], making the processors work at even lower voltages. Since this may lead to computation errors, appropriate checks and recomputation have to be added to algorithms in this case.

On the contrary, the Intel Turbo Boost technology opportunistically allows the processor to run faster than the nominal frequency if the CPU is operating below the defined power and temperature limits to speed up compute-intensive applications [34]. In detail, as explained in [35], "the thermal design power (TDP) represents the maximum amount of power the cooling system in a computer requires to dissipate. This is the power budget under which the system needs to operate. Nevertheless, this is not the same as the maximum power the processor can consume. The processor can consume more than the TDP for a short time without it being thermally significant." More details on this and the hardware power controller called Running Average Power Limit (RAPL) introduced with the Sandy Bridge architecture are provided in [36]. A similar solution, the NVIDIA Management Library (NVML), has been provided for NVIDIA GPUs [37, 38].

The Advanced Configuration and Power Interface specification has been developed since 1996 to provide the possibility to manage these aspects via software, e.g., at the operative system level. For example, ACPI defines up to 16 active states, named P0–P15, associated with a set of power/performance/latency characteristics [39]. In P0, the process runs at the maximum power and frequency level, while these values are decreased from P1 till maximum supported Pi [40].

2.2. Commercial-Off-the-Shelf Low-Power Devices. The energy-efficient architectures range from many-core architectures, such as the Graphics Processing Unit (GPU) to System on Chip (SoC), to Systems-on-Chip (SoCs). GPUs feature a high performance-per-watt ratio. At the time of writing this paper, the most powerful GPU devices, AMD MI100 and NVIDIA A100, presented, respectively, a peak performance of 38.33 gigaflops per watt (GFlops/W) and 24.25 GFlops/W considering 64 bit floating-point operations, with a power consumption of, respectively, 300 and 260 watt. It is, therefore, clear that GPUs aim at one side at energy efficiency, but they require careful programming and optimization to provide high computing performance.

The increasingly adopted class of low-power processors, often called System-on-Chip (SoC), originally designed for the embedded and mobile market, represents an attractive solution for scientific and industrial applications given their increasing computing performance coupled with relatively low cost and low electrical power demand.

SoC hardware platforms typically embed in the same die low-power multicore processors possibly combined with a GPU and all the circuitry needed for several I/O devices. For the case of off-the-shelf SoCs, various limitations may arise, such as 32 bit-only architectures, small CPU caches, small RAM sizes, high latency interconnections, and unavailability of ECC memory.

However, some solutions are progressively reducing the performance gap with high-end processors, with the added value of keeping a competitive edge on costs, reducing their carbon footprint, and preserving the environment. For these reasons, in this paper, we disregard devices such as Arduino or Raspberry Pi devices that, even if considered for compute-intensive applications [41], are mainly used for equipping IoT systems [42, 43] without significant, local preprocessing of data.

Fugaku represents the most important example of the adoption of SoCs for HPC—the first supercomputer in the TOP500 list of November 2020 and the most recent at the time of writing this paper—which is equipped with Fujitsu's 48-core A64FX SoC, providing a comparable performance-per-watt value with respect to GPU-based systems [44].

In the corresponding Green500 List, we can see that Fugaku appears in position 10 with a value of 15.418 GFlops/W, while NVIDIA DGX SuperPOD, the most energy-savvy system which is equipped with NVIDIA A100 GPUs,

provides 26.195 GFlops/W but is ranked only at position 170 in the TOP500. A more interesting comparison is between Fugaku and Selene, again a supercomputer equipped with A100 GPUs: this last appears in position 5 in both lists, with a value of 23.983 GFlops/W but providing only 63,460 TFlops/s with respect to 442,010 TFlops/s provided by Fugaku.

As for most HPC architectures, the question remains this [45]: do the raw numbers related to performance per second and watt correspond to achievable performance figures for most of the scientific applications and, in particular, for the application I am interested into?

This was the goal of the Computing On SOC Architecture (COSA) project [46, 47], an initiative funded by the Italian Institute for Nuclear Physics (INFN) between 2015 and 2018. In particular, the COSA project focused on assessing the energy consumption behavior of a wide set of state-of-the-art architectures using benchmarks and software widely used in many scientific applications.

In particular, an in-depth comparison of the performance of x86-based SoCs (i.e., Pentium N3700 and J4205, Avoton C2750, Xeon D1540, and Atom C3958) and low-power GPUs (i.e., Jetson TK1 and TX1) for state-of-the-art high-end solutions (i.e., Xeon E5-2683 and Tesla K20) is discussed in [23] with two benchmarks, represented by the widely used, computationally intensive N-body algorithm and the use of a deep learning approach applied to a classification problem, together with the real-world application taken from the field of molecular biology.

Although comparing high-end commercial/HPC servers with motherboards based on low-power SoC taken from the mobile and embedded world can be considered unfair, the results assess that the use of low-power architectures represents a feasible choice in terms of tradeoff among time-to-solution, energy-to-solution, and economic aspects.

The authors also discuss the economic aspects in [15, 48] by showing how a proper placement of the computational services considering edge and fog's composition cloud infrastructures is the key factor for achieving the best tradeoff between costs, performance, and power consumption.

Regarding the usage of SoCs based on ARM instruction set architectures (ISAs) or FPGAs, a quantitative evaluation is presented, for example, in [49], again using the N-body algorithm. Both these devices have been exploited in the ExaNoDe project to build a prototype of computing element for exascale [50].

However, it is to note that the porting of the code on these architectures is a bit more complex because the development and tuning tools have not yet reached the maturity level, ease of use, and does not provide the wide set of functionalities as those provided for free by Intel or NVIDIA [51].

2.3. HPC Low-Power Devices. If we move from off-the-shelf products to the design of new solutions for joining high performance and energy efficiency, one of the most important references is represented by the Mont-Blanc project, started in 2011. Its goal is to foster the development of a low-power European processor for Exascale, with a target of 50 GFlops/W at the processor level. This project is part of the European Processor Initiative, a Framework Partnership Agreement to develop the European skills in the design and exploitation of such processors.

Also, this project, together with ExaNoDe [52], is part of a wider group of EU-funded projects (e.g., ExaNeSt [53] focused on interconnection and storage and Ecoscale [54] focused on the heterogeneous architecture and, in particular, on the use of FPGAs), pursuing a strategic vision for economical, low-power approaches.

Also, the Mont-Blanc projects consider the use of ARM instruction set architectures (ISAs), such as the ThunderX processor family [55], and quantitative evaluations about different energy-performance tradeoffs achievable when designing an architecture based on mobile market technologies have been presented [56].

Heterogeneity seems to represent the most promising way, e.g., by integrating CPUs (X86 or ARM), GPUs, and FPGA in a single platform [57]. Also, the great efforts in developing unified programming models and API supporting all these heterogeneous hardware architectures such as OpenCL, SYCL, and oneAPI [58] demonstrate this trend.

3. Tools for Energy-Efficient Computing

In the previous section, we saw that power and energy consumption had become the driving metrics for computing hardware design and the most interesting CPUs. However, the advances in hardware efficiency must be followed by energy-aware algorithms, appropriate choice and allocation of specific hardware to applications, and adequate management techniques.

One of the most complete and interesting introductions to the problem was presented by Prof. Gallaghers [59] in summer school "ICT-Energy: Energy consumption in future ICT device" organized in 2016 within the context of the ICT-Energy European project [60].

The key concept is that energy is consumed by hardware, but this occurs under the control of software. Normal high-level languages (e.g., C++ and Java) hide the hardware characteristics, but the key aspect is that there could be many differences in the same high-level code (e.g., C++) machine instruction programs with different energy consumption figures. To this extent, an interesting tool is represented by Compiler Explorer [61], an open-source web application for interactive compiler code generation observation based on Node.js [62]. It shows the assembly output of the compiled code with different compilers and compiler versions to extract valuable information as, for example, for evaluating the power consumption.

Therefore, energy saving has to start at the software level to be propagated to the hardware level. Techniques for saving energy with power-aware hardware management or power capping [63] described in the previous section can represent a valuable complement. However, a key aspect, neglected by nearly all programmers, is their active engagement to inspect where a program wastes energy and, therefore, experiment with different designs. This is

obviously coupled with the fact that results have to be produced within an acceptable deadline [64], an aspect often disregarded approaching the energy efficiency problem.

3.1. Profiling Tools. The first step for achieving energy-efficient behavior is to investigate software behavior using information gathered as a program executes (i.e., profiling it) or simulating this through a performance model.

One of the most used tools for profiling is the Performance API (PAPI) analysis library [65]. PAPI is platform independent and provides developers with an interface and methodology for gathering performance-related data made available by hardware. The basic principle is to allow developers to see the relation between the software performance and processor events. As regards the power consumption, PAPI has been extended to measure and report energy and power values also on complex architectures [66].

Also, the PowerPack framework [67] provides a set of tools for analyzing the energetic performance. Unlike PAPI, the measurements are gathered on a separate machine in order to limit probe effects.

The scalable performance measurement infrastructure for parallel codes (Score-P) [68] has been extended for collecting information from technologies such as the aforementioned Intel RAPL.

Extrae is a tool relying on PAPI that allows collecting its countermetrics (including power and thermal data) for parallel programs [37]. Paraver effectively supports the analysis of such information, a visual data browser developed at the Barcelona Supercomputing Center as the previous one [69].

The Energy-Aware COmputing Framework (EACOF) has been designed to allow developers to profile their code for energy consumption [70]. In particular, it allows profiling codes in order to know exactly where energy is being used. Moreover, it allows applications to adapt at runtime based on current energy consumption. As an example application, the authors proposed a video player that may intelligently adapt based on energy consumption readings to ensure a video will complete before the battery runs out. The framework is available on GitHub [71], but no updates have been published since 2015.

In general, many tools such as these two have been presented in the literature. It is worth citing EProf [72], having the main feature to support fine-grained attributions of energy consumption to a particular function/software segment. However, in most cases, they are not actively maintained at the end of the projects where they have been developed, and software becomes difficult—if not impossible—to find and run.

A similar fate occurred for the Multiple Metrics Modeling Infrastructure (MuMMI) [73] project, focused on integrating existing tools such as PAPI and PowerPack for facilitating measurement, modeling, and prediction of software for multicore systems.

3.2. Dynamic Tuning. Some tools aim to achieve energy-saving figures automatically. In detail, many of them have been proposed, e.g., [74, 75], but, as stated before, not actively maintained. Here, we present just two of them because they are not part of wider and integrated solutions, which are discussed below.

The Global Extensible Open Power Manager (GEOPM) is a framework for exploring power and energy optimizations targeting high-performance computing [76]. One of the most interesting features is the possibility to dynamically coordinate hardware settings across all compute nodes used by an application in response to the application's behavior and requests from the resource manager. For example, it is possible to optimize MPI applications to improve energy efficiency or reduce the effects of work imbalance, system jitter, and manufacturing variation through built-in or user-defined control algorithms. The framework is available on GitHub [77].

The COUNTDOWN Slack library [78] allows identifying and automatically reducing power consumption during communication and synchronization primitives [79]. The library faces the problem of power wasting in communication and synchronization operations because of the adopted blocking mechanisms [80]: for example, nearly all MPI implementations use a busy-waiting mechanism. This library, on the contrary, is able to run a processor in a low-power mode, resulting in lower power consumption with limited or no impact on the execution time [81].

3.3. Integrated Solutions. The Runtime Exploitation of Application Dynamism for Energy-efficient eXascale computing (READEX) project has been funded by the European Union's Horizon 2020 research program between 2015 and 2018 to develop a tool-aided methodology for dynamic autotuning for performance and energy efficiency [82]. The tool suite was released in 2018, and it is available via GitHub [83].

The methodology is based on instrumenting an application with Score-P. This can be performed in an automatic way by compiling it with Score-P. Then, the dynamism of the application is detected and analyzed in order to identify the significant regions that will be managed with the project tuning methodology at runtime.

The key advantage of this suite is that it can be exploited by any developer even if she/he is unaware of the READEX methodology, with the result of increasing the energy efficiency of her/his application. It has been estimated that the application of the READEX tool suite to a nearly complex application can take several days [84], mainly for compiling the application with Score-P.

The Low-Energy Toolset for Heterogeneous Computing (LEGaTO) project has been funded by the European Union's Horizon 2020 research program between 2017 and 2020 to design and develop a software toolchain for energy-efficiency computing on heterogeneous hardware, i.e., a system equipped with CPUs, GPUs, and FPGA [57, 85].

The toolchain was released in 2020, and it is available via GitHub [86]. It is composed by several software components integrated to achieve a consistent programming environment across heterogeneous hardware platforms.

The hearth of the toolchain is represented by OmpSs [87], an extension to OpenMP developed at the Barcelona Supercomputing Center for supporting the asynchronous parallelism on heterogeneous resources as multicore CPUs, GPUs, and FPGAs.

An application in the OmpSs programming model is composed of one or more tasks with possible data dependency flow among some of them. The runtime environment analyses the resulting graph and produces a correct and possibly concurrent order of task execution. Several compiler and runtime systems (e.g., Nanos6, XiTAO [88], and Mercurium) support the process and manage all the energy efficiency, security, and fault-tolerance aspects [89].

Three use cases have been defined in healthcare, IoT for Smart Homes and Cities, and machine learning because they have different requirements in terms of energy efficiency, fault tolerance, and security. Results have been published in Deliverable 5.4 [90].

The Software Development toolKit for Energy optimization and technical Debt elimination (SDK4ED) project has been funded by the European Union's Horizon 2020 research program between 2018 and 2020 to minimize the cost, the development time, and the complexity of low-energy software development processes by designing a methodological approach and a software toolchain [91].

The SDK4ED platform [92] consists of five toolboxes: Technical Debt Management, Energy Optimization, Dependability Optimization, Forecaster, and Decision Support. They are implemented following the microservice paradigm as Docker images containing the specific web service.

Focusing on the Energy toolbox, it analyses projects available in an online repository (e.g., GitHub) on the machine running the Docker container with regard to its energy efficiency. This means it finds the energy hotspots, estimates the energy consumption through static or dynamic analysis [93, 94], and inspects possible solutions by suggesting specific code refactoring. This is a valuable approach, in particular, for software reusing [95].

The project ended at the end of 2020. Therefore, at the time of writing, not all the details and the code are available.

The Time, Energy, and security Analysis for Multi/Many-core heterogeneous PLAtforms (TeamPlay) project has been funded by the European Union's Horizon 2020 research program since 2018 to design and develop new techniques for producing highly parallel software for low-energy systems, such as IoT devices and CPS [96].

The idea is to develop a set of tools for allowing programmers to reason about time, energy, and security at the program source level. The idea is to design new language constructs to manage these extrafunctional properties as first-class citizens of the source code and express contracts in the source code that are machine-checkable by an underlying proof system.

The project is ongoing; therefore, at the time of writing, little information and software components were available.

4. Conclusions

Energy consumption is increasingly becoming one of the most relevant issues concerning the computing platforms for scientific applications and workloads.

As stated in [97], the huge level of energy consumption of ICT systems is probably due to the fact that nobody really cared for a long time, but today, things are changing because of economic reasons and also because our way of thinking has changed.

In this paper, we presented state-of-the-art solutions, both hardware and software, and methodological approaches for pursuing energy efficiency in scientific software to provide interested readers an updated introduction to the topic. The conclusion we can derive is that there are an increasing number of projects focusing on these topics, and some interesting SoC-based solutions are available. From the software side, instead, the situation is not satisfactory because tools are sometimes difficult to be found, not integrated, and, very often, disappear after the end of the project that developed them. What is actually needed is the definition of a common methodology and a coordination effort of groups acting in this field comparable with that of the Virtual Institute-High-Productivity Supercomputing (VI-HPS) [98], having in mind the tradeoff among time-to-solution, energy-to-solution, and usability of the proposed tools.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

[1] J. Wu, S. Guo, H. Huang, W. Liu, and Y. Xiang, "Information and communications technologies for sustainable development goals: state-of-the-art, needs and perspectives," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2389–2406, 2018.

[2] C. Magazzino, D. Porrini, G. Fusco, and N. Schneider, "Investigating the link among ict, electricity consumption, air pollution, and economic growth in eu countries," *Energy Sources, Part B: Economics, Planning, and Policy*, pp. 1–23, 2021.

[3] M. Heikkurinen, S. Cohen, F. Karagiannis, K. Iqbal, S. Andreozzi, and M. Michelotto, "Answering the cost assessment scaling challenge: modelling the annual cost of European computing services for research," *Journal of Grid Computing*, vol. 13, no. 1, pp. 71–94, 2015.

[4] G. Fagas, L. Gammaitoni, J. P. Gallagher, and D. Paul, *ICT-Energy Concepts for Energy Efficiency and Sustainability*, BoD–Books on Demand, Norderstedt, Germany, 2017, https://www.intechopen.com/books/ict-energy-concepts-for-energy-efficiency-and-sustainability.

[5] P. Kogge, S. Borkar, D. Campbell et al., "Exascale computing study: technology challenges in achieving exascale systems,"

Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Techinal Representative, vol. 15, no. 1, 2008.

[6] 2021, https://www.top500.org/green500/.

[7] T. R. Scogland, C. P. Steffen, T. Wilde et al., "A power-measurement methodology for large-scale, high-performance computing," in *Proceedings of the 5th ACM/SPEC international Conference on Performance Engineering*, pp. 149–159, Dublin, Ireland, March 2014.

[8] A. Shehabi, S. Smith, D. Sartor et al., *United States Data Center Energy Usage Report*, Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA, USA, 2016.

[9] A. Shehabi, S. Smith, E. Masanet, and J. G. Koomey, "Data center growth in the United States: decoupling the demand for services from electricity use," *Environmental Research Letters*, vol. 13, no. 12, p. 124030, 2018.

[10] D. D'Agostino, A. Clematis, A. Galizia et al., "The DRIHM project: a flexible approach to integrate HPC, grid and cloud resources for hydro-meteorological research," in *Proceedings of the SC'14: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 536–546, IEEE, New Orleans, LA, USA, November 2014.

[11] J. Wu, S. Rangan, and H. Zhang, *Green Communications: Theoretical Fundamentals, Algorithms, and Applications*, CRC Press, Boca Raton, FL, USA, 2016.

[12] F. Chiappori, I. Merelli, L. Milanesi, and A. Marabotti, "Static and dynamic interactions between GALK enzyme and known inhibitors: guidelines to design new drugs for galactosemic patients," *European Journal of Medicinal Chemistry*, vol. 63, pp. 423–434, 2013.

[13] D. Corrada, F. Viti, I. Merelli, C. Battaglia, and L. Milanesi, "myMIR: a genome-wide microrna targets identification and annotation tool," *Briefings in Bioinformatics*, vol. 12, no. 6, pp. 588–600, 2011.

[14] J. M. P. Cardoso, J. G. F. Coutinho, and P. C. Diniz, "High-performance embedded computing," in *Embedded Computing for High Performance*, J. M. Cardoso, J. G. F. Coutinho, and P. C. Diniz, Eds., pp. 17–56, Morgan Kaufmann, Boston, MA, USA, 2017, http://www.sciencedirect.com/science/article/pii/B9780128041895000028.

[15] D. D'Agostino, L. Morganti, E. Corni, D. Cesini, and I. Merelli, "Combining edge and cloud computing for low-power, cost-effective metagenomics analysis," *Future Generation Computer Systems*, vol. 90, pp. 79–85, 2019.

[16] C. Conficoni, A. Bartolini, A. Tilli, L. Benini, and G. Tecchiolli, "Energy-aware cooling for hot-water cooled supercomputers," in *Proceedings of the 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1353–1358, IEEE, Grenoble, France, March 2015.

[17] G. Fagas, J. P. Gallagher, G. Luca, and D. J. Paul, "Energy challenges for ICT," in *ICT—Energy Concepts for Energy Efficiency and Sustainability*IntechOpen, London, UK, 2017.

[18] M. Capra, R. Peloso, G. Masera, M. R. Roch, and M. Martina, "Edge computing: a survey on the hardware requirements in the internet of things world," *Future Internet*, vol. 11, no. 4, p. 100, 2019.

[19] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: greening big data," *IEEE Systems Journal*, vol. 10, no. 3, pp. 873–887, 2016.

[20] J. Wu, S. Guo, J. Li, and D. Zeng, "Big data meet green challenges: big data toward green applications," *IEEE Systems Journal*, vol. 10, no. 3, pp. 888–900, 2016.

[21] P. Czarnul, J. Proficz, and A. Krzywaniak, "Energy-aware high-performance computing: survey of state-of-the-art tools, techniques, and environments," *Scientific Programming*, vol. 2019, p. 8348791, 2019.

[22] O. Vysocky, L. Riha, and A. Bartolini, "Overview of application instrumentation for performance analysis and tuning," in *Parallel Processing and Applied Mathematics*, R. Wyrzykowski, Ed., Springer International Publishing, Cham, Switzerland, pp. 159–168, 2020.

[23] D. D'Agostino, A. Quarati, A. Clematis et al., "Soc-based computing infrastructures for scientific applications and commercial services: performance and economic evaluations," *Future Generation Computer Systems*, vol. 96, pp. 11–22, 2019.

[24] Y. Li, T. Jiang, K. Luo, and S. Mao, "Green heterogeneous cloud radio access networks: potential techniques, performance trade-offs, and challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 33–39, 2017.

[25] X. Cao, L. Liu, Y. Cheng, and X. Shen, "Towards energy-efficient wireless networking in the big data era: a survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 303–332, 2018.

[26] J. Murray, P. Wettin, P. P. Pande, and B. Shirazi, "Dynamic voltage and frequency scaling," in *Sustainable Wireless Network-on-Chip Architectures*, J. Murray, Ed., Morgan Kaufmann, Boston, MA, USA, pp. 79–105, 2016.

[27] D. Horak, L. Riha, R. Sojka, J. Kruzik, and M. Beseda, "Energy consumption optimization of the total-FETI solver and BLAS routines by changing the CPU frequency," in *Proceedings of the 2016 International Conference on High Performance Computing Simulation (HPCS)*, pp. 1031-1032, Innsbruck, Austria, July 2016.

[28] G. Papadimitriou, A. Chatzidimitriou, D. Gizopoulos et al., "Exceeding conservative limits: a consolidated analysis on modern hardware margins," *IEEE Transactions on Device and Materials Reliability*, vol. 20, 2020.

[29] 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.

[30] 2021 https://www.kernel.org/doc/html/v4.14/admin-guide/pm/cpufreq.html.

[31] V. Spiliopoulos, S. Kaxiras, and G. Keramidas, "Green governors: a framework for continuously adaptive DVFS," in *Proceedings of the 2011 International Green Computing Conference and Workshops*, pp. 1–8, IEEE, Orlando, FL, USA, July 2011.

[32] C. Scordino, L. Abeni, and J. Lelli, "Real-time and energy efficiency in Linux," *ACM SIGAPP Applied Computing Review*, vol. 18, no. 4, pp. 18–30, 2019.

[33] S. Catalán, J. R. Herrero, E. S. Quintana-Ortí, and R. Rodríguez-Sánchez, "Energy balance between voltage-frequency scaling and resilience for linear algebra routines on low-power multicore architectures," *Parallel Computing*, vol. 73, pp. 28–39, 2018.

[34] D. Lo and C. Kozyrakis, "Dynamic management of turbo-mode in modern multi-core chips," in *Proceedings of the 2014 IEEE 20th International Symposium on High Performance Computer Architecture (HPCA)*, pp. 603–613, IEEE, Orlando, FL, USA, February 2014.

[35] S. Pandruvada, *Running Average Power Limit," 01 STAFF*, Intel Open Source.org, Santa Clara, CA, USA, 2014, https://01.org/blogs/2014/running-average-power-limit--rapl.

[36] E. Rotem, A. Naveh, A. Ananthakrishnan, E. Weissmann, and D. Rajwan, "Power-management architecture of the intel

microarchitecture code-named sandy bridge," *IEEE Micro*, vol. 32, no. 2, pp. 20–27, 2012.

[37] F. Mantovani and E. Calore, "Performance and power analysis of HPC workloads on heterogenous multi-node clusters," *Journal of Low Power Electronics and Applications*, vol. 8, no. 2, p. 13, 2018.

[38] K. Kasichayanula, D. Terpstra, P. Luszczek et al., "Power aware computing on GPUs," in *Proceedings of the 2012 Symposium on Application Accelerators in High Performance Computing*, pp. 64–73, IEEE, Argonne, IL, USA, July 2012.

[39] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, "Energy management for commercial servers," *Computer*, vol. 36, no. 12, pp. 39–48, 2003.

[40] I. Ratković, N. Bežanić, C. S. Ünsal, A. Cristal, and V. Milutinović, "An overview of architecture-level power-and energy-efficient design techniques," *Advances in Computers*, vol. 98, pp. 1–57, 2015.

[41] P. M. M. Pereira, P. Domingues, N. M. M. Rodrigues, G. Falcao, and S. M. M. Faria, "Assessing the performance and energy usage of multi-CPUs, multi-core and many-core systems: the MMP image encoder case study," *International Journal of Distributed and Parallel Systems*, vol. 7, no. 5, pp. 1–20, 2016.

[42] D. R. Patnaik Patnaikuni, "A comparative study of arduino, raspberry pi and esp8266 as iot development board," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 5, 2017.

[43] P. B. Otte and D. Djukanovic, "A cost effective and reliable environment monitoring system for HPC applications," *CoRR*, vol. abs/1802, p. 00724, 2018.

[44] R. Okazaki, T. Tabata, S. Sakashita et al., "Supercomputer Fugaku Cpu A64fx realizing high performance, high-density packaging, and low power consumption," *Fujitsu Technical Review*, https://www.fujitsu.com/global/documents/about/resources/publications/technicalreview/2020-03/article03.pdf, 2020.

[45] E. Danovaro, A. Clematis, A. Galizia, G. Ripepi, A. Quarati, and D. D'Agostino, "Heterogeneous architectures for computational intensive applications: a cost-effectiveness analysis," *Journal of Computational and Applied Mathematics*, vol. 270, pp. 63–77, 2014.

[46] http://www.cosa-project.it/.

[47] D. Cesini, E. Corni, A. Falabella et al., "Power-efficient computing: experiences from the cosa project," *Scientific Programming*, vol. 2017, p. 7206595, 2017.

[48] I. Merelli, L. Morganti, E. Corni et al., "Low-power portable devices for metagenomics analysis: fog computing makes bioinformatics ready for the internet of things," *Future Generation Computer Systems*, vol. 88, pp. 467–478, 2018.

[49] D. Goz, G. Ieronymakis, V. Papaefstathiou et al., "Performance and energy footprint assessment of FPGAs and GPUs on HPC systems using astrophysics application," *Computation*, vol. 8, no. 2, p. 34, 2020.

[50] P. Y. Martinez, Y. Beilliard, M. Godard et al., "Exanode: combined integration of chiplets on active interposer with bare dice in a multi-chip-module for heterogeneous and scalable high performance compute nodes," *2020 IEEE Symposium on VLSI Technology*, pp. 1-2, 2020.

[51] A. Armejach, "Porting the mont-blanc 2020 applications to teh arm isa and SVE," Tech. Rep. D3.5, MONT-BLANC Project, Ile-de-France, France, 2020.

[52] A. Rigo, C. Pinto, K. Pouget et al., "Paving the way towards a highly energy-efficient and highly integrated compute node for the exascale revolution: the exanode approach," in *Proceedings of the 2017 Euromicro Conference on Digital System Design (DSD)*, pp. 486–493, IEEE, Vienna, Austria, September 2017.

[53] M. Katevenis, N. Chrysos, M. Marazakis et al., "The exanest project: interconnects, storage, and packaging for exascale systems," in *Proceedings of the 2016 Euromicro Conference on Digital System Design (DSD)*, pp. 60–67, IEEE, Limassol, Cyprus, September 2016.

[54] I. Mavroidis, I. Papaefstathiou, L. Lavagno et al., "Ecoscale: reconfigurable computing and runtime system for future exascale systems," in *Proceedings of the 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 696–701, IEEE, Dresden, Germany, March 2016.

[55] A. Armejach, M. Casas, and M. Moretó, "Design trade-offs for emerging HPC processors based on mobile market technology," *The Journal of Supercomputing*, vol. 75, no. 9, pp. 5717–5740, 2019.

[56] A. Adria, C. Marc, and M. Miquel, "Design trade-offs for emerging HPC processors based on mobile market technology," *The Journal of Supercomputing*, vol. 75, no. 9, pp. 5717–5740, 2019.

[57] B. Salami, K. Parasyris, A. Cristal et al., "Legato: low-energy, secure, and resilient toolset for heterogeneous computing," in *Proceedings of the 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 169–174, IEEE, Grenoble, France, March 2020.

[58] 2021, https://khr.io/tr.

[59] 2021, https://www.nipslab.org/files/summerschool-aalborg-jpg-part1.pdf.

[60] K. Eder and J. Gallagher, "Energy-aware software engineering," *ICT-Energy Concepts for Energy Efficiency and Sustainability*, pp. 103–127, 2017.

[61] 2021, https://repo.hca.bsc.es/epic/.

[62] M. Godbolt, "Optimizations in c++ compilers," *Queue*, vol. 17, no. 5, pp. 69–100, 2019.

[63] C. Jin, B. R. De Supinski, D. Abramson et al., "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.

[64] A. Quarati, A. Clematis, and D. D'Agostino, "Delivering cloud services with QOS requirements: business opportunities, architectural solutions and energy-saving aspects," *Future Generation Computer Systems*, vol. 55, pp. 403–427, 2016.

[65] D. Terpstra, H. Jagode, H. You, and J. Dongarra, "Collecting performance data with PAPI-C," in *Tools for High Performance Computing 2009*, pp. 157–173, Springer, Berlin, Germany, 2010.

[66] H. McCraw, J. Ralph, A. Danalis, and J. Dongarra, "Power monitoring with PAPI for extreme scale architectures and dataflow-based programming models," in *Proceedings of the 2014 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 385–391, IEEE, Madrid, Spain, September 2014.

[67] R. Ge, X. Feng, S. Song et al., "Powerpack: energy profiling and analysis of high-performance systems and applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 5, pp. 658–671, 2009.

[68] A. Knüpfer, C. Rössel, D. A. Mey et al., "Score-p: a joint performance measurement run-time infrastructure for periscope, scalasca, tau, and vampir," in *Tools for High Performance Computing 2011*, pp. 79–91, Springer, Berlin, Germany, 2012.

[69] A. Munera, S. Royuela, G. Llort et al., "Experiences on the characterization of parallel applications in embedded systems

with extrae/paraver," in *Proceedings of the 49th International Conference on Parallel Processing-ICPP*, pp. 1–11, Edmonton, Canada, August 2020.

[70] H. Field, G. Anderson, and K. Eder, "Eacof: a framework for providing energy transparency to enable energy-aware software development," in *Proceedings of the 29th Annual ACM Symposium on Applied Computing*, pp. 1194–1199, Gyeongju, Republic of Korea, March 2014.

[71] 2021 https://github.com/eacof/eacof.

[72] S. Schubert, D. Kostic, W. Zwaenepoel, and K. G. Shin, "Profiling software for energy consumption," in *Proceedings of the 2012 IEEE International Conference on Green Computing and Communications*, pp. 515–522, IEEE, Besancon, France, November 2012.

[73] X. Wu, C. Lively, V. Taylor et al., "Mummi: multiple metrics modeling infrastructure," in *Proceedings of the 2013 14th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*, pp. 289–295, IEEE, Honolulu, HI, USA, July 2013.

[74] B. Rountree, D. K. Lowenthal, B. R. De Supinski et al., "Adagio: making DVS practical for complex HPC applications," in *Proceedings of the 23rd International Conference on Supercomputing, ICS '09*, Yorktown Heights, NY, USA, June 2009.

[75] A. Marathe, P. E. Bailey, D. K. Lowenthal, B. Rountree, M. Schulz, and B. R. De Supinski, "A run-time system for power-constrained HPC applications," in *Lecture Notes in Computer Science*, vol. 9137, pp. 394–408, Springer, Berlin, Germany, 2015.

[76] J. Eastep, S. Sylvester, C. Cantalupo et al., "Global extensible open power manager: a vehicle for HPC community collaboration on co-designed energy management solutions," in *High Performance Computing*, J. M. Kunkel, Ed., Springer International Publishing, Cham, Switzerland, pp. 394–412, 2017.

[77] 2021, https://github.com/geopm/geopm.

[78] 2021, https://github.com/EEESlab/countdown.

[79] D. Cesarini, A. Bartolini, A. Borghesi, C. Cavazzoni, M. Luisier, and L. Benini, "Countdown slack: a run-time library to reduce energy footprint in large-scale MPI applications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 11, pp. 2696–2709, 2020.

[80] M. Torquati, D. De Sensi, G. Mencagli, M. Aldinucci, and M. Danelutto, "Power-aware pipelining with automatic concurrency control," *Concurrency and Computation: Practice and Experience*, vol. 31, no. 5, p. e4652, 2019.

[81] D. Cesarini, A. Bartolini, P. Bonfà et al., "Countdown: a run-time library for performance-neutral energy saving in MPI applications," *IEEE Transactions on Computers*, vol. 70, 2020.

[82] J. Schuchart, M. Gerndt, P. G. Kjeldsberg et al., "The readex formalism for automatic tuning for energy efficiency," *Computing*, vol. 99, no. 8, pp. 727–745, 2017.

[83] 2021 https://github.com/readex-eu.

[84] L. Riha, *D5.3: Evaluation of the READEX Tool Suite Using the READEX Test-Suite*, READEX Project, Naples, FL, USA, 2018, https://www.readex.eu/wp-content/uploads/2018/11/D5.3.pdf.

[85] D. Gizopoulos, G. Papadimitriou, A. Chatzidimitriou et al., "Modern hardware margins: CPUs, GPUs, FPGAs recent system-level studies," in *Proceedings of the 2019 IEEE 25th International Symposium on On-Line Testing and Robust System Design (IOLTS)*, pp. 129–134, IEEE, Rhodes, Greece, July 2019.

[86] 2021, https://github.com/legato-project.

[87] A. Duran, E. Ayguadé, R. M. Badia et al., "Ompss: a proposal for programming heterogeneous multi-core architectures," *Parallel Processing Letters*, vol. 21, no. 2, pp. 173–193, 2011.

[88] M. Pericàs, "Elastic places: an adaptive resource manager for scalable and portable performance," *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 2, 2018.

[89] K. Givaki, B. Salami, R. Hojabr et al., "On the resilience of deep learning for reduced-voltage FPGAs," in *Proceedings of the 2020 28th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*, pp. 110–117, IEEE, Västerås, Sweden, March 2020.

[90] 2021 https://legato-project.eu/sites/default/files/uploaded/d5.4.pdf.

[91] L. Papadopoulos, C. Marantos, G. Digkas et al., "Interrelations between software quality metrics, performance and energy consumption in embedded applications," in *Proceedings of the 21st International Workshop on Software and Compilers for Embedded Systems, SCOPES '18*, Sankt Goar, Germany, May 2018.

[92] 2021, https://gitlab.seis.iti.gr/sdk4ed-wiki/wiki-home/wikis/home.

[93] M. Axling, *D3.2 Suitable Monitor Indicators for Energy Consumption*, SDK4ED ProjectCentre of Research & Technology – Hellas (CERTH), Marousi, Greece, 2019, https://drive.google.com/file/d/1zkX71EfL2ybfWTzPNUPKTHhb145-DzuH/view.

[94] D. Tsoukalas, *D3.4 Forecasting Methods for TD/Energy/Dependability*, SDK4ED ProjectCentre of Research & Technology – Hellas (CERTH), Marousi, Greece, 2019, https://drive.google.com/file/d/1DVhM9JvSD3LsSVXlE9SfBrBXWobHMSXT/view.

[95] N. Nikolaidis, G. Digkas, A. Ampatzoglou, and A. Chatzigeorgiou, "Reusing code from stackoverflow: the effect on technical debt," in *Proceedings of the 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA'19)*, IEEE, Kallithea, Greece, August 2019.

[96] A. M. Coutinho Demetrios, D. De Sensi, A. F. Lorenzon et al., "Performance and energy trade-offs for parallel applications on heterogeneous multi-processing systems," *Energies*, vol. 13, no. 9, p. 2409, 2020.

[97] S. D'Elia, "Powering up: energy and computing," *HiPEAC Info*, vol. 59, 2020.

[98] 2021, https://www.vi-hps.org/about/about.html.