Energy Concerns with HPC Systems and Applications

Roblex Nana
Mines Paris - PSL,
Centre de Recherche en Informatique (CRI)
Fontainebleau, France
roblex.nana_tchakoute@minesparis.psl.eu

Claude Tadonki
Mines Paris - PSL,
Centre de Recherche en Informatique (CRI)
Fontainebleau, France
claude.tadonki@minesparis.psl.eu

Petr Dokladal
Mines Paris - PSL,
Centre de Morphologie Mathématique (CMM)
Fontainebeau, France
petr.dokladal@minesparis.psl.eu

Youssef Mesri
Mines Paris - PSL,
Centre de Mise en Forme de Matériaux (CEMEF)
Sophia Antipolis, France
youssef.mesri@minesparis.psl.eu

ABSTRACT

For various reasons including those related to climate changes, *energy* has become a critical concern in all relevant activities and technical designs. For the specific case of computer activities, the problem is exacerbated with the emergence and pervasiveness of the so called *intelligent devices*. From the application side, we point out the special topic of *Artificial Intelligence*, who clearly needs an efficient computing support in order to succeed in its purpose of being an *ubiquitous assistant*. There are mainly two contexts where *energy* is one of the top priority concerns: *embedded computing* and *supercomputing*. For the former, power consumption is critical because the amount of energy that is available for the devices is limited. For the latter, the heat dissipated is a serious source of failure and the financial cost related to energy is likely to be a significant part of the maintenance budget. On a single computer, the problem is commonly considered through the electrical power consumption. This paper, written in the form of a survey, we depict the landscape of energy concerns in computer activities, both from the hardware and the software standpoints.

KEYWORDS

Energy profiling, power measurement, HPC systems, Embedded systems, Optimization

Contents

Abstract 1
Contents 1
1 Introduction 1
2 Related Review Works 2
3 Overview of energy metrics in HPC 3
3.1 Standard energy consumption metrics 3
3.2 Energy metrics in supercomputers 3
3.3 From energy to Carbon footprint 4
4 Hardware architecture from the energy standpoint 5
4.1 Accelerators 5
4.2 Embedded systems: Microcontrollers 5
4.3 General Purpose Processors 6
5 Energy management tools 7
5.1 Energy tools for GPUs 7
5.2 Energy tools for CPUs 7
5.3 Energy tools for microcontrollers 8
5.4 Energy tools for Modern HPC systems 8
6 About cooling Systems 11
6.1 Cooling technologies for HPC systems 11
6.2 News trends in liquid cooling design 12
7 Energy Optimization techniques 12
7.1 Static energy optimization approaches 12
7.2 Dynamic energy optimization approaches 14
7.3 Hybrid energy optimization approaches 14
8 Energy of AI processing 14
8.1 Motivation 14
8.2 Studies on CO_2 concerns with AI 15
8.3 Energy profiling tools for AI applications 15
8.4 Energy optimization of AI applications 16
9 Prospective viewpoint 17
10 Conclusion 17
References 18

1 INTRODUCTION

Manufacturers of high performance computing (HPC) systems are striving to provide more and more potential computing power by acting on the related hardware aspects like: *number of cores, vectors units, 3-operands units, accelerators*, to name the main ones. Performance is a high priority in servers and supercomputers beside storage capacity. In order to leverage the potential power of HPC systems, efforts are made reach better implementations through cutting-edge programming and code optimisation techniques [126]. The reality is that these performance-guided activities do not explicitly consider the energy efficiency. Energy saving has become one of the main challenges for the new generations of servers and supercomputers.

Nowadays, the design of HPC systems considers *power efficiency*: 21.1 MW for the 1.102 EFlop/s Frontier, 29.9 MW for the 442 PFlop/s Fugaku, and 2.94 MW for the 151.9 PFlop/s Lumi to consider the top 3 machines of the most recent TOP500 list [37]. The associated electricity bill increasingly dominates the overall costs related all the activities of HPC systems.

The problem is generally formulated as the need to reach a good trade-off between *time-to-solution* and *energy-to-solution*. Different approaches have emerged to solve this problem, which can be summarised as follows: vendors work on power-efficient processors and software developers on how to use them at the best [30]. However, an effective solution is possible only by properly managing all layers of the system, from the software stack to the cooling system [28]. Thus, we need power efficient software's as well as hardware integrated solutions and optimized devices.

The HPC market is growing significantly as the topic itself is becoming popular and the need for computing speed a genuine fact. The so-called "embedded HPC" is a new and emerging topic, which consists on the development and use of highly parallel micro-servers/embedded devices as mainframe computing systems [23]. These systems are increasingly used particularly in the field of artificial intelligence (AI) to support both *data collection* and *model inference*. The advantage of using embedded devices is their energy efficiency for a competitive computing performance compared to traditional CPUs. For machine/deep learning inference, a new generation of Coral Dev Bord micro-controllers can outperform traditional Intel Skylake server processors by more than 20x times on both time performance and energy efficiency[48]. This illustrate the energy efficiency and the processing speed of embedded systems over CPUs for some specifics applications. Thus, these low-power systems are widely considered as good candidates when energy is the central concern.

There are several contributions in the literature on energy in HPC and embedded systems. These works range from the definition of metrics [12, 105] to the optimization of energy [72, 110, 128] through the development of profiling and energy management tools[52, 120]. This work is carried out on the hardware and software side of the systems as well as on the algorithmic level, targeting different types of system *(embedded, CPU, GPU, FPGA and hybrid)*.

The focus on *energy* in the context of computer systems is also related to *carbon footprint*, which is a more general concern currently in the spotlight. Indeed, *energy* can be turned into *carbon emission* by multiplying it with the *carbon intensity* of the energy supply[103]. If the power consumption of most hardware components is well known or can be measured accurately, it not the case with carbon emission, which has to (roughly) estimated by specific means or using the aforementioned conversion.

In this paper, we survey a taxonomy of energy concerns in computers systems. For each type of system *(general purpose computers, accelerators, embedded systems/micro-controllers and modern supercomputers)*, we present stat-of-the-art (SOTA) architectures with a focus on power management tools. Another contribution of this work is a survey of SOTA energy/power optimization techniques with an emphasis on AI applications and a prospective analysis on all studied systems.

The remainder of this paper is structured as follows: Section 2 presents a review of existing surveys that address the topic of energy management in HPC and embedded systems. Section 3 is about a quantitative overview of energy/power aspect in computer, with a focus on the main energy/power and carbon footprint metrics. Section 4 show an overview of most recent energy aware hardware architecture for HPC and AI workloads. Section 5 discuss about a taxonomy of energy concern in embedded systems, accelerators, general purpose computers and modern supercomputers, with a special focus on energy management and optimization tools. Section 6 discuss about cooling system technologies from the energy consumption standpoint. Section 7 present, comments and discusses some energy/power optimization techniques from the literature. Section 8 present a literature review for energy concern of AI applications in computers system. Section 9 give a short prospective analysis of this survey together with some technical recommendations. Finally, Section 10 concludes the paper.

2 RELATED REVIEW WORKS

Beloglazov et al. [13] discuss about the sources and issues of high power/energy consumption, and provide a taxonomy key aspects related to energy-efficient design of computing systems, covering different levels including *hardware, operating system, virtualization* and *data center*. The main aim of their taxonomy is to guide future design and development activities.

A survey by Kocot et al. [72] investigates energy-aware scheduling methods used in modern HPC systems starting with the problem definition and then tackling various goals associated to this challenge, including a bi-objective approach that considers power and energy constraints. The work considers the standard types of HPC system (multicore CPU and GPU) together with related energy-saving mechanisms based on dynamic voltage/frequency scaling (DVFS), power capping, and other functionalities. The work uses a collection of carefully selected algorithms, classified by the programming paradigm (e.g. machine learning or fuzzy logic).

Czarnul et al. [29] provides a state of the art on energy-aware high-performance computing (*tools, techniques and environments*). They identify and classify the main approaches by *system/device types*, *optimization metrics*, and *energy/power control methods*. The work describes energy management tools (benchmarking, prediction, and simulation) and optimization approaches for standard devices (CPU/GPU/Hybrid) under various configurations (clusters, grids, and clouds). The authors point out the need for the unification of energy management interfaces for different architectures. Their conclusion states that we need to develop energy-aware methods for heterogeneous environments; indicates the optimization goals worth investigating based on minimizing the product of *energy* and *computing time*; and expresses the need for the validation of energy management tools.

An overview on energy-saving efforts is provided by Maiterth et al. [84], where they focused on energy/power-aware job scheduling and resource allocation as a major step towards more efficient systems. The paper considers nine large HPC centers located over three continents and the answers to eight questions from by their respective staff. Practical management procedures including *power capping, job killing*, and *virtual machines* are described. Moreover, the focus of the study is more engineering oriented as it does not provide any formal or theoretical aspect related to energy-aware scheduling.

A survey by Chaudhry et al. [25] addresses thermal-aware scheduling and associated techniques for green data centers. Their study focuses on the thermal and cooling aspects of tasks scheduling, where a balanced heat distribution among the racks of the server is the main objective. They indicate some metrics to evaluate thermal awareness in green data centers. In addition, they provide a thermal modeling together with effective solutions to prevent from hard-to-cool phenomena such as hot spots. They proposed two approaches: *reactive*, where the problem is fixed upon occurrence; and *proactive*, where the goal is to prevent the problem from occurring (e.g. using the thermal model of the server room followed by a proper tasks assignment on the compute nodes).

A technical report by Ramesh et al. [110] presents a taxonomy of power/energy concerns in embedded systems design. The proposed taxonomy is derived from a systematic review of the literature, where a categorization of the topics of interest is constructed. The authors considered a collection of 95 papers related to energy management from ACM, IEEE Xplore, and SpringerLink databases. Their study focuses on energy dissipation and power optimization from the standpoint of hardware devices and that of support tools for energy profiling and optimization.

Many review works about energy concerns are generally specific to computer infrastructures (i.e., data-centers, embedded systems, supercomputer, etc.) for energy optimizations techniques, tools, and measurement. In this work, we survey the energy concern on HPC systems in a more general way considering all kind of approaches for energy/power management as presented in the taxonomy displayed in Figure 1

3 OVERVIEW OF ENERGY METRICS IN HPC

The matter of appropriate energy and performance metrics has been investigated in several survey papers. However, since technology and associated features are evolving very rapidly, these studies lack some aspects that we present in this paper together with updated information related to news technologies.

There are various approaches to power measurement and different types of outputs. We can classify theses measurement approaches into two groups: *Out-of-band* (e.g. power meters) and *in-band* (e.g. RAPL counters). Out-of-band measurement is the easiest approach to consider. It uses an external device to measure power consumption without a little to no interference in the computational performance. In-band measurement requires some technical information about the target hardware and can access specific registers in a programmatic manner. Both types of measurement can be enhanced with an application-level profiling. However, it might be difficult to assess the type and detail of the measurements that are needed to obtain satisfactory insights from the energy profiling of the application. This is a major concern with the Out-of-band measurement, which uses an external device whose output data cannot be directly obtained within a program.

3.1 Standard energy consumption metrics

In order to express the energy consumption at any level, we will use the most basic formula that links *energy* to *power* and *time*. The energy consumption E can be expressed as:

$$E = \sum_{i=0}^{n} P_i \delta_i, \qquad (1)$$

where we assume a constant power P_i for time period δ_i, with $\delta_0 + \delta_1 + \cdots + \delta_n = T$, where T is the overall time period considered. One might consider an average power P of the P_i over period T and therefore write $E = P \times T$. With this basic formula, we can clearly see which are the two orthogonal levers at our disposal to act on energy consumption. The variations of P and T are quite opposite, indeed the energy optimization of an HPC system is a matter of a good trade-off between the *execution time* and the *consumed power*. The goal is to optimize one while keeping the other at an acceptable level.

The reference unit of *energy* measurement according to the international system of units is the Joule (J). In relation with a time period, there is the watt-per-hour or watt-hour (Wh), with the relation $1Wh = 3.6x10^3 J$. In this study we will use both of them, but in most of the cases we will refer to the Watt, which is reference unit of *power* (i.e. energy consumed in a time period of 1 hour).

The first approach to get the energy consumption of a given application is to directly measure the electrical power of the targeted hardware through specific devices (*out-of-band* approach). The second approach is seek an approximation of the energy consumption using a prediction/estimation model (usually considered for performance). The first method is often used to assess the accuracy of estimation approaches.

3.2 Energy metrics in supercomputers

Energy consumption is one of the major concerns when it comes to the deployment of large-scale HPC infrastructures. This must be taken into account at all levels *(from hardware to software tools)* and raises new scientific and operational challenges.

In the top500 ranking of June 2022[37], the exascale performance (both theoretical and sustained) has been reached with the Frontier supercomputer [99]. The exascale was an important milestone in the HPC roadmap, and this level of potential performance is the current target of several high-end HPC infrastructures. The cost associated to the energy consumption by large-scale supercomputers is noticeable and the associated carbon footprint is becoming a serious concern. The Green500 [70] consider an evaluation of the *FLOPS per Watt* to rank supercomputers. Correlating the two metrics, we can state that the challenge is to increase the performance per energy consumed (FLOPS/Watt). Energy-efficient computing is a multi-dimensional problem, especially in the extreme-scale computing. The electricity consumption, thus the associated bill, includes the power due to machines operation and cooling system. A 2019 estimates “A typical supercomputer consumes anywhere between 1 to 10 megawatts of power on average, which is equal to the electricity needs of almost 10,000 homes” [19]. For instance, the electricity bill paid by the RIKEN institute in 2020 for their (energy-efficient) *Fugaku* supercomputer was nearly $60 millions [8]. Table 1 give some illustrative data about the electricity bill of the top five supercomputers of the november 2022 top500 ranking [37]. We assume that the whole supercomputer is running continuously during 1 hour, thus we get the estimate electricity cost (in dollars, last column) by considering the cost per KWh that applies in the geographical location of the computing center. We considered the electricity prices per country on September 2022 [44].

Machine	Peak Perf.	Power	$/KWh	Total(K$)
Frontier	1.685 EFLOPS	21.1MW	0.150	3.165
Fugaku	537.2 PFLOPS	29.9MW	0.219	6.548
LUMI	428.7 PFLOPS	6.02MW	0.198	1.192
Leonardo	255.7 PFLOPS	5.61MW	0.561	3.147
Summit	200.8 PFLOPS	10.1MW	0.150	1.515

Table 1: Electricity cost per hour for the top five supercomputers.

The so-called *thermal design power* (TDP), also called *thermal design point*, is the maximum amount of generated heat (by a computer chip or component) that the cooling system is designed to dissipate. The *power rating* (highest power input allowed) for a microprocessor is generally 1.5 times the TDP [105]. The purpose of the TDP is to provide system designers with a power target so as to guide the selection of a convenient thermal solution. Under a steady workload, the TDP is the maximum power consumption of the processors. However, during the turbo mechanism or certain

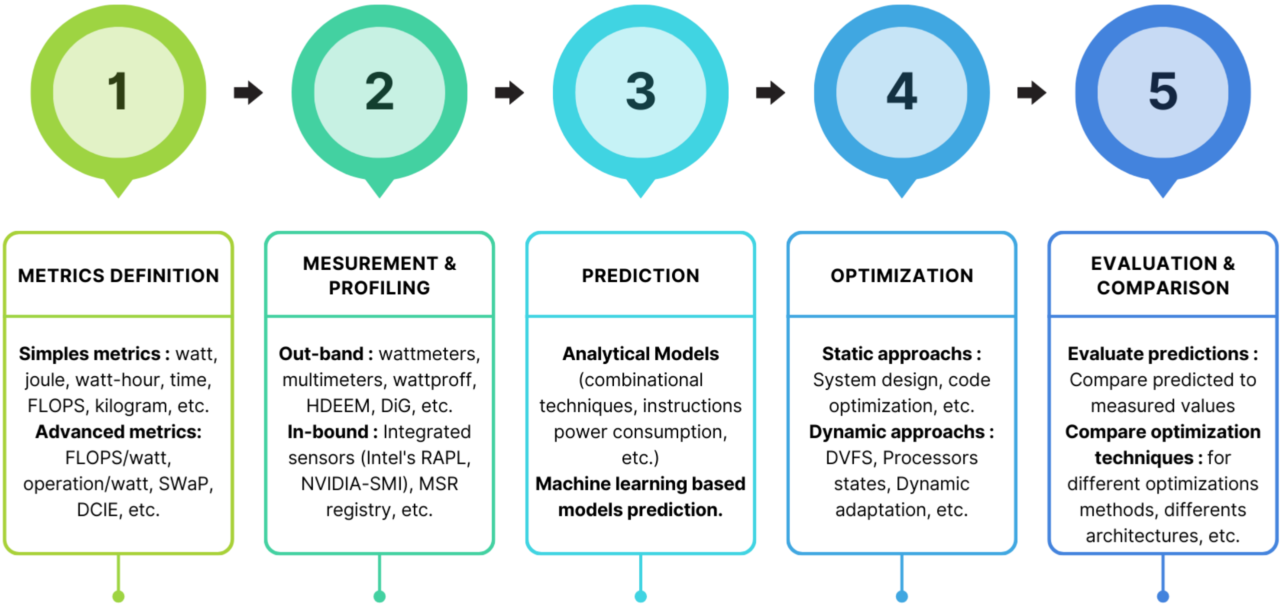
1
2
3
4
5
METRICS DEFINITION
Simples metrics : watt, joule, watt-hour, time, FLOPS, kilogram, etc.
Advanced metrics: FLOPS/watt, operation/watt, SWaP, DCIE, etc.
MESUREMENT & PROFILING
Out-band : wattmeters, multimeters, wattproff, HDEEM, DiG, etc.
In-bound : Integrated sensors (Intel's RAPL, NVIDIA-SMI), MSR registry, etc.
PREDICTION
Analytical Models (combinational techniques, instructions power consumption, etc.)
Machine learning based models prediction.
OPTIMIZATION
Static approachs : System design, code optimization, etc.
Dynamic approachs : DVFS, Processors states, Dynamic adaptation, etc.
EVALUATION & COMPARISON
Evaluate predictions : Compare predicted to measured values
Compare optimization techniques : for different optimizations methods, differents architectures, etc.

Figure 1: Taxonomy of power/energy management solutions with related approaches and tools.

types of workload such as vectors instructions, it can sometimes exceed the maximum TDP.

The so-called *Average CPU Power* (ACP), a concept defined by AMD for Opteron processors, is the average dissipated power of a processor while running a defined set of benchmarks (Transaction Processing Performance Council (TPC Benchmark*-C), SPECcpu*2006, SPECjbb*2005, and STREAM) [121]. AMD indicated that these measurements to determine the ACP value are not to be considered for every processor, but only for some particular ones selected by its manufacturing units [9].

From thermal standpoint, the processor TDP specification is a critical value because any thermal solution should dissipate at the level of that rated indication. Intel and AMD both agree on this point. If a given processor design is based on the ACP, then it might be undersized and out of its thermal specifications. For servers the main concern is not on how much power a specific component dissipates, but instead the power of the entire server when running a given workload. The corresponding measurement can be easily made following and *out-to-band* approach with a power meter on the input cord(s).

SWaP (Space, Wattage and Performance) [33] is an objective three-dimensional metric that provides a more comprehensive and realistic way to evaluate servers. It is calculated considering *performance, power* as indicated by equation 2 that follows:

$$SWaP = \frac{Performance}{Space * Power}, \tag{2}$$

where *Performance and Power* is measured by any convenient benchmarks, and *Space* is related to the size of the computer.

The so-called *Power usage effectiveness* (PUE)[12] is a metric used to determine the energy efficiency of a data center. It is determined by dividing the total amount of incoming power by the consumed power as expressed by formula 3.

$$PUE = \frac{Total\_Facility\_Energy}{IT\_Equipment\_Energy} = 1 + \frac{Non\_IT\_Facility\_Energy}{IT\_Equipment\_Energy} \tag{3}$$

According to the *"Uptime Institute Annual Global Data Center Survey 2021"* [31], PUE and power consumption are among the top tracked sustainability metrics. But in 2022, key findings reported in the Uptime Institute Global Data Center Survey 2022 [66] indicated the requirement of additional metrics to supplement PUE for future efficiency gains, which should focus on IT power. A similar benchmarking standard considered by the Green Grid is DCiE (*Data Center Infrastructure Efficiency*), which is just the inverse of PUE. Both metrics apply to a more global level, thus they do not capture the consumption specific to the computing activities. Indeed, information technology (IT) equipment include computing units and all associated peripherals. Nevertheless, having such a macroscopic information makes sense as all considered facilities are related to the computing activities.

A survey by Jin et al. [69] presents the state-of-the-art on green data center techniques including *energy efficiency, resource management, thermal control* and *green metrics*, with a detailed comparison among them and key challenges for future research.

3.3 From energy to Carbon footprint

Climate change currently stands as a critical concern because of its significant impact on ecosystems and livelihoods across the world. It's a clear fact that carbon dioxide emissions are the primary driver of global climate change. According to recent estimates, the total CO_2 emissions of the information and communications technology (ICT) sector account for around 2.1%–3.9% of global CO_2 emissions[39]. Therefore, estimating and reducing the carbon footprint in ICT is worth all related efforts.

The typical way for carbon footprint estimate of IT infrastructure activities is to derivative it from power consumption. The paper by Patterson et al [103] provides a valuable study of the carbon footprint of computing workloads. They stated that CO_2 *equivalent emissions* (CO_2e) accounts for *carbon dioxide* and all the other greenhouse gasses as well like *methane and nitrous oxide* for instance. This equivalent emission can be calculated from the *electric power* by multiplying it with the *carbon intensity* of the energy supply as expressed through formula (4) [103]:

$$CO_2e = Wh * (CO_2e \ per \ Wh) \tag{4}$$

Carbon intensity (CO_2e per Wh) is the amount of carbon dioxide (CO_2e) that is released to produce a watt-hour of electricity. The average data-center carbon emissions in 2020 was 0.429 tCO_2e (ton of carbon dioxide equivalent emissions) per MWh (Megawatt hour), but the gross tCO_2e per MWh can be 5x lower in some specific data-centers [103]. Table 2 is provided as an illustration of the carbon footprint for top ranked supercomputers. We used formula (4) and an estimation of the *carbon intensity* from 2022 data [100]. We clearly see that the floating-point performance and the necessary (CO_2e) are not directly correlated, the hardware profile of the machines is a key factor.

Machine	Peak Perf.	Power	Kg(CO_2)/KWh	CO_2(kg$)
FRONTIER	1.685 EFLOPS	21.1MW	0.379	7 997
FUGAKU	537.2 PFLOPS	29.9MW	0.479	14 322
LUMI	428.7 PFLOPS	6.02MW	0.132	795
LEONARDO	255.7 PFLOPS	5.61MW	0.372	2 087
SUMMIT	200.8 PFLOPS	10.1MW	0.379	3 828

Table 2: CO_2 per hour for the top five supercomputers.

4 HARDWARE ARCHITECTURE FROM THE ENERGY STANDPOINT

4.1 Accelerators

4.1.1 GPU. GPUs are specialized devices designed for efficient graphics rendering and image processing. Their parallel structure makes them more efficient than *traditional CPUs* for algorithms that process large blocks of data in parallel. Nowadays, GPUs stand as the reference accelerator in the HPC landscape. In addition to being now designed as general purpose units, GPUs have a top consideration when it comes to energy efficient in HPC [56]. From the absolute standpoint, modern GPUs consume a significant amount of power (from 50-600W or even more). However, because of their noteworthy processing speed, they show better performance-per-watt than standard CPUs for specific workloads.

AMD Instinct MI250X was ranked world's fastest HPC accelerator in 2022[3]. This GPU has a (double-precision) peak performance of 47.9 TFLOPS and a peak power between 500 and 560 TDP. A combination with cutting-edge processors yield very powerful HPC systems, like the (AMD EPYC CPU, AMD Instinct MI250X) CPU-GPU pairing of the FRONTIER (Exascale) supercomputer and other top ranked machines from the top500 list of November 2022[37]. NVIDIA H100 Tensor Core GPU [96] is the response from NVIDIA about this innovation from AMD as competitive material with 700W TDP in maximum configuration. Intel Launched *Intel Data Center GPU Max Series* project in 2022 with PONTE VECCHIO as the first competitive product for data center GPU market [63] with 600W TDP. A comparative view in terms of *flops performance* and *power supply* of the major accelerators is provided in table 3.

4.1.2 TPU. With the high computing power required for cutting-edge AI, domain-specific architectures for Neural Network computations have emerged, like the *Tensor Processing Unit (TPU)*, a Deep Neural Network (DNN) accelerator from Google. An individual Edge TPU can perform 4 trillion operations per second (4 TFLOPS) with only 2 watts of power. The latest TPU (*version 4*) has an average TDP of 192W. For illustration, the Edge TPU can execute state-of-the-art mobile vision models such as MobileNet V2 at almost 400 frames per second in a power efficient manner [46]. Pandey P. et al [102] parameterized the extreme hardware under-utilization in a TPU systolic array and proposed UPTPU: an intelligent data-flow adaptive power-gating paradigm that yields a improvement of the TPU energy efficiency by factor 3.5 to 6.5 on different input batch sizes. This ultra low power devices is nowadays integrated as an accelerator into microcontrollers (single-board unit), as we can see with *Coral Dev Board* for instance.

4.1.3 FPGA. Field Programmable Gate Arrays (FPGAs) are integrated circuits with ability to be reconfigured to implement a specific processing at the hardware level. Initially applicable to very specific domains, FPGAs has extended so as to now stand as a important components of servers and supercomputers, as well as edge computing systems[15]. However, their energy efficiency is still an important concern, with no easy or standard ways for hardware/software power management. Hosseinabady and Nunez-Yanez [55] investigate the use of FPGAs in an embedded system for energy saving. They study the energy efficiency of a hybrid FPGA-CPU device that can switch between hardware and software on periodic tasks. In addition, they successfully applied the voltage and frequency scaling (VFS) to reduce the energy consumption. Moreover, they showed that in some cases, if the task's period is higher than a specific threshold a reduction of the energy consumption cannot be obtained on the FPGA, hence the effectiveness of a software support for energy saving. Experimental results show up to 48% energy reduction by applying the proposed techniques at runtime on thirteen individual tasks. As previously said, the major accelerator in the HPC landscape remains the GPU, however the FPGA is becoming a serious candidate.

4.2 Embedded systems: Microcontrollers

Due to their small size and single-chip configuration (thus at the expense of processing power, memory and storage), microcontrollers have a little energy consumption while keeping a certain level of computing efficiency. Much more energy is required to power a GPUs and standard CPUs, which yields some limitations and constraints in their usage. Micro-controllers are typically not wired into main power, they instead rely on batteries or residual energy. For example, a micro-controller can run on a single coin battery for weeks or even months. However, having a low power system does not yield lower energy consumption by itself. Indeed, it is important to optimize the software, not just in terms of functionality or processing efficiency, but also with respect to energy efficiency. We now describe some of the major devices of embedded computing.

4.2.1 Arduino. When it comes to microcontrollers and embedded systems, one of the first candidate that pops is *Arduino*. Arduino is an open-source electronics platform based on easy-to-use hardware and software for ultra low power chips, which consumes less than a single watt of nominal power. The Arduino Portenta H7 [7] is currently the most powerful IoT Cloud compatible boards of the Arduino series. Arduino can be used to connect devices, visualize data, control and share projects online. Beginners and advanced users can meet their specific needs from

Name	RAM (GB)	core frequency (GHz)	TDP (w)	Peak TOPS	peak TFLOPS(fp32)	performance/ watt (INT8)
Tesla A100 SXM4	80	1.41	400	312(bf16)/624(int8)	19.5	1.56 TOPS/W
Tesla H100 SXM5	80	1.98	700	1000(bf16)/2000(int8)	60	3.33 TOPS/W
AMD Instinct MI250X	128	1.7	560	383 (bf16 or int8)	95.7	0.68 TOPS/W
Intel Ponte Vecchio	128	1.6	600	720(bf16)/1440(int8)	45	2.40 TOPS/W
Google TPU v4	32	1.05	192 (idle)	275 (bf16 or int8)	/	1.43 TOPS/W

Table 3: SOTA accelerators systems characteristics

its wide range of features and possibilities. However, even with Portenta H7, Arduino is not powerful enough to handle HPC workloads in the context of embedded systems as compared with others microcontrollers (see Table 4).

4.2.2 Raspberry Pi. While Arduino is an electronic board with a simple microcontroller, Raspberry Pi is a full-fledged computer. Unlike Arduino, Raspberry Pi has its own operating system, thus it can carry out more complex operations (e.g. *robot control* and *weather monitoring*, to name these two). The *Raspberry Pi 4 Model B* [112] is the latest model of the Raspberry Pi microcontrollers series. It offers a noteworthy increase in processing speed, multimedia performance, memory and connectivity over the previous generation (Raspberry Pi 3 Model B+), while keeping full compatibility with earlier versions and same level of power consumption. The Model B offers a level of performance comparable to that of entry-level x86 PC systems but with the advantage of energy efficiency as it has a maximum nominal power of 10W.

4.2.3 Intel NCS2. Intel Neural Compute Stick 2 (Intel NCS2)[64] is a *plug-and-play* Development Kit for AI Inference. NCS2 is based on an Intel Vision Processing Unit(VPU) chip named Movidius X. Movidius provides its Neural Compute Stick (i.e. *Fathom*) to bring a basic-level deep learning capabilities into embedded devices. It can be used to develop, fine-tune, and deploy convolutional neural networks (CNNs) on low-power applications that require real-time inference. It supports heterogeneous execution across computer vision accelerators (CPU, GPU, VPU, and FPGA) using a unified API. Its so-called Vision Processing Unit (VPU) includes vision accelerators, a Neural Compute Engine, imaging accelerators, and 16 SHAVE vector processors paired with a CPU in one heterogeneous package. The combination of the aforementioned units provides a total of up to 4 TFLOPS with 1.5W of power [106]. However, Intel is discontinuing this product and its technical support will continue until June 30, 2023, while warranty support will continue until June 30, 2024[64].

4.2.4 Nvidia Jetson. The Jetson Nano Developer Kit [95] is the most popular board from Nvidia Jetson series. It delivers a noteworthy processing capability to efficiently support high-performance AI at low power and cost. The developer kit can be powered by micro-USB and comes with extensive I/Os. This makes it simple for developers to connect a diverse set of new sensors to enable a variety of applications at a little power of 5 watts. The Jetson AGX Orin Developer Kit [97] is currently the most powerful board from this series, with up to 275 TOPS for running the NVIDIA AI software stack. It enables to create advanced robotics and edge AI applications. But this performance incurs a higher cost with currently more than 2000$ for 60W TDP.

4.2.5 Coral Dev Board. The Coral Dev Board[47] is a single-board computer with a removable system-on-module (SOM) that contains eMMC, SOC, wireless radios, and Google's Edge TPU. It's perfect for IoT devices and other embedded systems that demand fast on-device ML inferencing. Coral dev is also the most efficient out of all the microcontrollers we have found. With on board TPU, it is capable of performing 4 tera-operations per second (TOPS), using 0.5 watts for each TOPS (2 TOPS per watt)[48]. The USB version can be connect to any system running Debian Linux (including Raspberry Pi), macOS, or Windows 10. Coral Dev is very fast but, with bad tech support, faulty units and seems like a very common problem.

4.3 General Purpose Processors

4.3.1 x86 based processors. x86 is a family of CISC instruction set architectures, initially developed by Intel from Intel 8086 microprocessor and its 8088 variant. It was introduced in 1978 as a fully 16-bit extension of Intel's 8-bit 8080 microprocessor, with memory segmentation as a solution for addressing more memory than can be covered by a plain 16-bit address. Embedded systems and general-purpose computers used x86 chips before the IBM Personal Computer in 1981. Nowadays, most desktop, workstation, laptop and server computers are based on the x86 architecture family, while mobile categories such as smartphones or tablets are dominated by ARM. The fastest supercomputer in the TOP500 list for November 2022 (Frontier) is built with *AMD Epyc* CPUs that are based on the x86 ISA. The market of CPUs in the HPC landscape and data centers is still dominated today by x86 CPUs.

AMD claim that its EPYC processors power the most energy-efficient x86 servers, delivering exceptional performance with lower energy consumption [4]. AMD EPYC 9654 servers shall use up to 29% less annual power than Intel Xeon Platinum 8490H servers at the same performance, while helping reduce capital expenditure up to 46% [4]. Note that these two CPU models require respectively 350W and 360W for Intel and AMD CPUs respectively.

4.3.2 ARM based processors. Energy saving has become one of the main challenges for new generation servers and supercomputers. Many manufacturers of HPC systems consider low-power ARM components that are also present today in the vast majority of embedded or mobile systems. Indeed, the particularity of ARM components is their low energy consumption with a competitive processing performance as Intel and AMD x86 architectures. Several international collaborative projects like the Japanese Post-K, the European Mont-Blanc, or the UK's GW4/EPSRC, announced the adoption of ARM technology for their high-performance computing (HPC) systems [85]. On November 2018, for the first time, an ARM-based system was listed in the Top500 ranking. It was the Astra[136] supercomputer powered by Marvell's ThunderX2 ARM CPU and hosted at the Sandia National Laboratories (USA).

Name	memory(GB)	core frequency (GHz)	TDP (w)	Peak TOPS	peak TFLOPS(fp32)	performance/watt
Raspberry Pi 4B	8	1.5	10	/	0.135	2.02 GFLOPS/W
Jetson Nano	4	1.43	10	0.472(int8)	0.236	0.047 TOPS/W
Jetson AGX Orin	64	2.0	60	275(int8)	5.3	4.58 TOPS/W
Arduino Portenta H7	0.008	0.48	1.15	/	/	/
The Coral Dev Board	4	1.5	0.65	4(int8)	/	2 TOPS/W
Intel NCS2	8	0.7	1.5	4(int8)	/	2.66 TOPS/W

Table 4: Characteristics of selected state of the art embedded systems

Name	memory(GB)	core frequency (GHz)	TDP (w)	Peak TOPS	Peak GFLOPS (fp64)	performance/watt
Intel Platinum 8490H	4000	1.9	350	/	3 648	10.42 GFLOPS/W
AMD EPYC 9654	6000	2.4	360	7763(int8)	3 686	10.23 GFLOPS/W
Fujutsu A64FX	32	2.6	150	3.4(int8)	3 400	22.66 GFLOPS/W
Marvell ThunderX2	512	2.2	180	/	563	3.12 GFLOPS/W

Table 5: SOTA General purpose computers characteristics

5 ENERGY MANAGEMENT TOOLS

5.1 Energy tools for GPUs

NVIDIA-SMI (NVIDIA System Management Interface) [94] is a command line utility for the management and monitoring of NVIDIA GPU devices that is based on the NVIDIA Management Library (NVML). The tool can be used to set the power range (max and min, in Watt) of the execution of a given application. Its GPU Operation Mode (GOM) allows to reduce the power usage and optimize the GPU throughput by disabling some features accordingly. It also implements a power scaling algorithm to dynamically reduce the clock frequency when the GPU is consuming too much power.

5.2 Energy tools for CPUs

- Intel RAPL[65](Running Average Power Limit Interface) is an interface for reporting the (accumulated) energy consumption of various system-on-chip (SoC). The RAPL's energy reporting feature has been available on many generations of Intel SoC products. Intel processors utilize this energy information for internal SoC management purposes such as the control of power limits in association with the Turbo Boost power limit settings. Energy information from the RAPL interface gets updated every 1 ms, which is several orders of magnitude slower than what physical side channel probing could achieve. RAPL measurements ignore a large part of the power consumption of servers because they focus on CPU and RAM. Some experiments on Intel processor from Grid5000 [51] show that it just represent 42% of the overall servers consumption [5].

- AMD RAPL counters : Concerning Zen architecture, AMD replaced APM (Application Power Management) with RAPL. The implementation is similar to the corresponding Intel's RAPL, but uses different control registers. While Intel typically provides multiple domains and the option to limit power consumption over various time frames, AMD only considers registers for memory reads and core power consumption. However, the latter is available with a per-core spatial resolution, while a per-package applies for Intel's core domain. Schöne et al. [118] highlighted various energy efficiency aspects of the AMD Zen 2 micro-architecture in order to facilitate system comprehension and optimization. Key findings include qualitative and quantitative descriptions regarding *core frequency transition delays*, *workload-based frequency limitations*, and *effects of I/O die P-states on memory performance.* The authors made a comparative study with some high-end Intel architectures (i.e., Cascade Lake, Skylake, Haswell) for power efficiency and provided details on power measurements accuracy on both architectures. The work shows that AMD RAPL is unsuitable to optimize the overall energy consumption. Their approach failed on reflecting the influence of the operands, which can also be seen as a benefit when it comes to side-channel attacks that are based on power measurement.

- For ThunderX2[86] chips, there is no RAPL counters but there are other *harware specific* on-chip sensors. These sensors are not yet supported by common libraries like PAPI[17], perf-tools[38] for instance. However, Marvell[86] has provided an tool named *tx2mon* [87], which is based on the Linux kernel driver *tx2mon_kmod* to provide access to specific system data and allow to configure the way to measure energy.

- *Model-Specific Register* (MSR) is any of the various control registers in the x86 architecture used for debugging, program execution tracing, computer performance monitoring, and toggling certain CPU features.

- ACPI (Advanced Configuration and Power Interface)[130] is an open standard that the operating system can use to discover and configure the components of the computer, to perform power management, auto configuration, and status monitoring. ACPI defines the performance states, designated by P-States. P-States correspond to different performance levels that apply while the processor is actively executing instructions according to energy saving and performance trade-off scenarios. Each system manufacturer decides its way to implement this specification standard to save energy in the system. For example, Intel CPUs, regarding Haswell architecture, provides voltage regulators per core, thus each core has its own P-State.

- Device Tree (DT)[80]: While ACPI was historically created for x86 platforms, the ARM ecosystem developed "Device Tree" (DT) to describe the same information for ARM-based devices. Thus, ACPI and DT overlap in that they both provide mechanisms for enumerating devices and attaching additional configuration data

Name	type	Objective	Techniques	Portability
NVML(NVIDIA-MSI)[94]	Software	power management	Dynamic Power Management, Power capping, Sampling measurement	Linux with Nvidia GPU devices; never tested it on Windows

Table 6: SOTA accelerators energy/power management tools

to devices (which can be used by higher layers of software). The biggest difference between DT and ACPI is that DT is effectively a structured mechanism for passing arbitrary data, while ACPI provides standardised data.

- PAPI (Performance API) [17] library is a platform independent tool which provides developers with an interface and methodology for gathering performance-related hardware data. The basic principle is to allow developers to see the relation between the software performance and corresponding processor events. McCraw et al. [88] extended PAPI to measure and report energy and power values even on complex architectures.

- Intel Power Gadget [61] is one of the most easy-to-use energy profiler. It provides a graphical user interface with a few plots showing CPU and DRAM utilisation (%), cores frequency (GHz), temperature (°C), and power consumption (W). The total energy consumption of the CPU and DRAM written into files (i.e. *Log files*). When installing Intel Power Gadget, its command-line interface (named PowerLog) is also installed.

- Powerstat [26] is an easy-to-use tool to measure energy consumption on Linux. Intel Power Gadget and PowerLog are not compatible for Linux system, so Powerstat was developed similarly to the previous tools. Powerstat is just another wrapper around an Intel library RAPL. However, it provides a simple interface for a command-line usage.

- PowerTOP[62] is a Linux tool used to diagnose issues with power consumption and power management. In addition to being a diagnostic tool, PowerTOP also has an interactive mode that can be used to handle various power management settings in case the direct mode is restricted by the OS. Its main advantage is the ability to estimate the energy consumption of the considered machine. It provides an interactive mode to fine-tune power management settings in Linux system.

- Perf tools[38]: A very quick and easy way to obtain the energy consumption of a program in a Linux environment, is through Perf. It is a command-line tool that offers a wrapper to Intel's RAPL. It facilitates the collection of energy measurements of the components of a computer and associated devices.

- Another quick way of getting energy and power measurements for Intel processors is through Likwid[129]. Likwid uses the RAPL interface, developed by Intel, to fetch energy and power measurements from different types of CPU. Compared to Perf, Likwid does not offer an option to run a given test several times. However, it provides *power* estimation in addition to energy measurement. Moreover, Likwid offers other options such as thread's temperature monitoring.

- PyJoules[59] is a software toolkit written in Python to measure the energy footprint of a given host machine. It monitors the energy consumed by a specific device of the host machine. It works ionly with intel CPUs, RAM (for intel server architectures), intel integrated GPUs and nvidia GPUs.

5.3 Energy tools for microcontrollers

- EEMBC CoreMark-Pro [36] is a benchmark that aims at becoming the industry standard for embedded platforms. It contains five (resp. four) prevalent integer (resp. floating-point) workloads. The workloads in CoreMark-Pro represent a wide variety of performance characteristics, memory utilization, and instruction-level parallelism, highlighting the strengths or and weaknesses of the target processor in term of performance and energy efficiency.

- EEMBC ULPBench [35] is a benchmark whose the goal is to overload a given processor in order to help determining the maximal amount of energy consumed. The benchmark consists of a number of mathematical and sorting operations. The STMicroelectronics PowerShield provides the backbone of the framework for probing an embedded system energy measurement.

- Dr. Wattson [131] is an Energy Monitoring Module for high quality energy monitoring and measurements for microcontrollers boards. It is coupled with easy to use Arduino and Python libraries to provide quality AC energy data like *RMS Current*, *RMS Voltage*, *Power Factor*, *Line Frequency*, *Active/Apparent Power*, with just a few of lines of code.

- PSoC (Programmable System on Chip) 5LP[68] is a data acquisition (DAQ) system for measuring and analyzing the power consumption of microcontrollers. DAQ system consists of a current measurement circuit using potentiostat technique (i.e, apply constant voltage during experiment). The DAQ device is based on system on chip PSoC 5LP and Python program for the analysis, storage and visualization of measured data. Implemented DAQ device is connected with a computer through a USB port and tested with developed Python program.

- N^3 profiler [34] is a power consumption monitoring tool to detect anomalies in power consumption for ARM-Based embedded systems at the level of the components. The authors used NARX (Nonlinear AutoRegressive eXogenous) [16] neural networks model as estimator to monitor energy/power for profiling and diagnosis purposes. N^3 improves upon the accuracy reported in the literature while maintaining low power and computational overhead. Experimentation was done on a smartphone considered as an embedded device.

Comment : For a more accurate power measurement on micro-controller board, the following actions can be considered: disable HDMI and LEDs if present; minimize accessories usage (a connected keyboard for instance); be selective with Software (different programs running) and disable WiFi. Different system commands can be used depending on the micro-controller to disable the aforementioned features.

5.4 Energy tools for Modern HPC systems

- HDEEM (High Definition Energy Efficiency Monitoring)[52] is an FPGA-based system on-chip that is intended to equip a compute node for its power measurement. The aim is to aggregate at high frequency (1 kHz) the measurements made by watt-metrics probes distributed among the components of the compute node.

Name	Type	Objective	Techniques	Portability
RAPL counters[65]	hardware	power management	Dynamic Power Management, Power capping, Sampling measurement	x86 CPU
ACPI[130]	specification	power management	Dynamic Power Management, Power capping	x86 CPU
DT[80]	specification	power management	Dynamic Power Management, Power capping	ARM CPU
Perf tools[38]	software	performance and energy management	interface to hardware counters	Linux with Intel devices for energy
PAPI[88]	software	performance and energy management interface	interface to hardware counters	All Linux systems
Likwid-powermeter[129]	software	power profiling	query RAPL counters	Linux devices with Intel processor
PowerTOP[62]	software	energy monitoring	query Intel RAPL	Linux with AMD or Intel devices
PyJoules[59]	software	energy monitoring	query RAPL and Nvidia SMI interfaces	Linux with AMD, Nvidia or Intel devices
Powerstat[26]	software	measure energy consumption	query Intel RAPL	Linux on Intel PCs
Power Gadget and PowerLog[61]	software	energy/power and temperature monitoring	query Intel RAPL	Mac or Windows on Intel PCs
tx2mon[87]	software	energy/power and temperature monitoring	query hardware counters	Marvell ThunderX2

Table 7: SOTA General purpose computers tools for energy/power management

Name	Type	Objective	Technique	Portability
EEMBC CoreMark[35]	software	system benchmark for energy consumption	stress CPU with specific workload	8 to 64-bit microcontrollers
EEMBC ULPMark[35]	software	system benchmark for energy consumption	stress CPU with specific workload	8, 16 and 32-bit microcontrollers
PSoC 5LP[68]	software and hardware	system benchmark for energy consumption	SoC Module based on PSoC and python program	all microcontrollers
Dr. Wattson [131]	software and hardware	Energy Monitoring	SoC Module based on Arduino and python program	Arduino, Raspberry and simillars microcontrollers
N^3[34]	software	Monitoring, diagnosis, software optimization	machine learning prediction	embedded systems

Table 8: SOTA Embedded systems energy/power management tools.

The samples associated to the last 7 hours of execution can be stored in a local memory of HDEEM for direct accesses through a programming interface in C language and/or through reads from report files.

- Similar to HDEEM, WattProf[111] is a system-on-chip based on an FPGA that can be connected via a PCIe interface to a compute node. WattProf comes with dedicated wattmetric probes that can be plugged on the PCIe interface of the targeted hardware components and also on the connectors for the DRAM. WattProf includes a memory for storing samples of energy consumption measurements, and an API to access those samples.

- DiG (Dwarf in a Giant) [78], is another system on-chip based on an Arduino 5. Unlike HDEEM and WattProf, DiG connects to the power supply of the computer and thus captures its overall energy consumption rather than that of individual components. The Arduino board is used to process the energy consumption data, as well as to send them out through the network of the supercomputer. It might be more convenient or efficient to dedicate an individual unit to the management of the measurements coming from the participating compute nodes. That unit will thus serve as the provider of energy measurements to the user. In addition, DiG also allows for accurate and high frequency sampling, while remaining a low cost system-on-chip for HPC.

- PowerPack [42] was the first tool to isolate the power consumption of common devices including *disks, memory, NICs*, and *CPU* within a given machine and correlate the corresponding measurements with the main subroutines of the applications being profiled. The framework support multi-core and multiprocessor-based nodes and provides in-depth analyses of the energy consumption of parallel applications. These analyses include the impacts of multiprocessing at the level of the chip on energy efficiency. The authors used the framework to study the power dynamics and energy efficiencies of DVFS techniques on clusters, and the experiments showed that DVFS scheduling can intelligently enhance system energy efficiency while maintaining processing performance. They claim that their methodology as described in their work can be extended to other architectures and measurement devices. For instance, one can directly use the power sensors integrated in emergent computer systems for a more convenient power measurement.

- BDPO(Bull Dynamic Power Optimizer)[123] is a dynamic reconfiguration tool that runs as a daemon behind a given HPC application and adapts the clock frequency of the CPUs according to the workload. It has the particularity of being completely agnostic to the considered application, as well as to the platform, while not requiring any configuration from the user. The authors of the tool experimentally got that the use of BDPO reduces the energy consumed by the execution of NEMO and HPCG applications by about 15%, while maintaining the associated overhead below 4% [123].

- Phase-TA [123] is a tool for analysing the profiles of iterative HPC applications, especially those produced by Bull Dynamic Power Optimizer (BDPO) [123] (see the previous paragraph). It detects locally periodic behaviours and try to characterise them by constructing patterns corresponding to the associated periodicities. The authors experimentally showed that the patterns constructed by Phase-TA are relevant representations of the considered periodicities, which seem to dominate the execution time. The observed performance of Phase-TA allows to consider the use of Phase-TA during the execution of an HPC application for its energy monitoring.

- PMAC (Power Monitoring and Controlling) Tool [21] is a web-based power monitoring and controlling tool for energy optimization of HPC applications. PMAC reports the power consumption of the sofatwer as well as for the hardware in real-time. It allows to manages power based on application's profile and DVFS mechanism. The specificity of the tool is that it can be used as an energy profiling as well as an energy optimizer. In the latter case, the tool uses its own profile report to guide the power optimization strategy. Experimental results have shown an energy saving of 12 -15% when using P-MAC. P-MAC uses CMAF (C-DAC Multi-Agent Framework) for the transmission and execution of control policies.

- EAR (Energy Aware Runtime) [77] is an energy management framework for *energy measurement, energy management* and *energy optimization*. EAR supports standard CPUs as well as (NVIDIA) GPUs. It is constantly being enhanced to support other and upcoming technologies as well. The optimization of the energy consumption of an HCP cluster is done at two levels: the *compute node level*, which is provided by the EAR library and the *system level* for power caping using DVFS techniques.

- EERT (Energy Aware Rescheduling Tool)[20] is another energy management tool that act on the internal scheduling of HPC applications in order to the reduce energy consumption through maximizing the CPU utilization and switching off idle nodes. The benefit is more noticeable when there is a important imbalance in the distribution of workloads over the nodes. EERT seamlesly uses the *distributed multithreaded check-pointing (DMTCP)* mechanism for check-pointing. Experimental results provided by the authors show 15% energy saving when using EERT.

- FIRESTARTER [120], is a handy utility that aims at creating near-peak power consumption on standard compute nodes. It can be used for tests of cooling and power infrastructures, system stability test, or as a maximum power consumption baseline for application energy efficiency studies. FIRESTARTER is currently only available for the Linux operating system and has supports for Intel architectures (Nehalem, Westmere, Sandy Bridge, Ivy Bridge, Haswell, Broadwell, Skylake, Knights Landing), AMD family 15h and 17h (Zen, Zen+, Zen 2) processors, and NVIDIA GPUs. The tools stresses the most important *power consumer* parts of compute nodes: CPU (cores + memory related components such as the caches), GPUs, and main memory and report some metrics that include power consumption.

- lo2s [57] is a lightweight performance monitoring tool. The tool collects performance and energy data w.r.t various metric (i.e., perf counters, kernel trace-points, model specific registers, and custom metric data provided by plugins). These trace data are stored in the Open Trace Format 2 (OTF2) that can be used for offline analysis using tools like Vampir [71]. Ilsche et al.[58] investigated improvements of lo2s by combining a detailed recording of system events with information from a high-resolution power measurement in the process of recording the scheduling of applications and C-state transitions.

- READEX [98] is a tool suite that supports users to improve the energy-efficiency of their HPC applications. It enables them to exploit the dynamic behaviour of their applications by adjusting the system to the actual resource requirements and thus improves energy-efficiency and performance. It uses a multi-agent based approach to identify runtime situations and to determine optimal system configurations. The tools also provides insights for the specification of domain knowledge to improve the automatic tuning impact. The result of the analysis step guides runtime tuning. Figure 2 provides an overview of READEX working diagram.

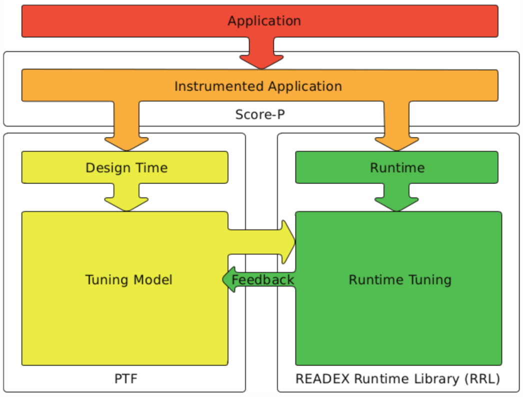

Figure 2: READEX working diagram

- MERIC [119, 134] is Lightweight C/C++ library (with an interface for Fortran applications) that measures energy consumption and timings of annotated regions inside a user application. The MERIC library evaluates application behavior in terms of resource consumption and runtime parameters including *Dynamic Voltage and Frequency Scaling* (DVFS) and *Uncore Frequency Scaling* (UFS). It performs dynamic application tuning following the READEX approach. The library was originally developed for Intel x86 systems, but additional supports for AMD, IBM and selected ARM systems chips was added. It also supports HDEEM and DiG hardware tools for energy measurement. A tool called RADAR VISUALIZER [134] for visualization of the analyzed application behavior in different system configurations was proposed to analyze MERIC results.

Name	Type	Objective	Technique	Portability
WattProf[111]	hardware	energy/power measurement	system on-chip based power monitoring board	all server node
HDEEM[52]	hardware	power measurement	system on-chip based power monitoring board	all server node
DiG[78]	hardware	energy monitoring	system on-chip based power monitoring board	all server node
PowerPack[42]	software	energy/power measurement	isolate power consumption of devices in measurement	Linux systems
EERT[20]	software	power management	dynamic rescheduling, core usage maximization	Linux HPC systems
Phase-TA[123]	software	energy profiling	analysing the profiles of HPC applications	Linux systems
PMAC[21]	software	power management	DPM and DVFS techniques; web based monitoring	Linux systems
BDPO[123]	software	power optimization	DFS on computing cores during workloads execution	Linux x86 systems
lo2s[57]	software	performance and energy profiling	Sample hardware counters events	Linux x86 systems
EAR[77]	software	energy management	DPM techniques, power capping, On/Off policies	Linux with Intel, AMD and Nvidia devices
READEX[98]	software	energy and performance optimization	exploit the dynamic behaviour of application and make resources allocation	Linux x86 and ARM systems
MERIC[134]	software	energy management	dynamic application tuning and hardware energy measurement	Linux x86, ARM and Nvidia GPUs systems; HDEEM and DiG supports
FIRESTARTER[120]	software	benchmark tests of cooling and maximum power consumption	stress execution units and data transfer between cores and memory hierarchy	x86 CPU and GPU

Table 9: SOTA Supercomputers systems tools for energy/power management

6 ABOUT COOLING SYSTEMS

Designing computers that perform tasks efficiently without overheating is a major consideration for all manufacturers nowadays. Current CPUs and GPUs has a power consumption from tens to hundreds watts. Some specific CPUs consume little power like those of embedded systems and mobile devices (few milliwatts or microwatts). Computers consume electrical energy and dissipate part of it in as heat coming from the resistance in the circuits. Excessive heat is a clear threat for the integrity of hardware components, with the risk of leading to serious damage. Thus, *cooling system*, which can be internal or external, is crucial in order to cap cap the dissipated heat so as to avoid a critical overheating.

6.1 Cooling technologies for HPC systems

Cooling is crucial to HPC systems, especially for large-scale ones, but choosing the right technology depends on several factors like the *desired temperature* limits and the *operating cost*. There are mainly four types of cooling that are commonly considered: *air cooling*, *liquid cooling*, *rear door heat exchanger (RDHX)* and *immersive cooling*.

- **Air cooling** is the most basic cooling mechanism and also the most used one. With cheapest infrastructure costs, air cooling relies on a fan to take heat away from components. This solution is not sufficient for large-scale HPC, as users require increasingly dense computing solutions, which generate more heat per rack of servers.

- **Liquid cooling** needs less energy to operate and it stands as the best cooling option because liquid has the ability to transfer heat much more efficiently than air[139]. In addition, it is a more ecological approach on a global viewpoint. Moreover, ambient heat removed from systems can then be used for a heating solution, thereby enhancing or replacing traditional heating systems[27].

- **Rear Door Heat Exchanger (RDHX)** is a cooling approach that combines both air and water cooling mechanisms. It has shown a great efficiency on data centers as it acts at the level of the rack, thus it wide and increasing consideration[117]. Technically, chilled water is fed to a coil or backplate inside the RDHX, then rack-mount devices eject the hot exhaust air through the RDHX, transferring the heat to the water and ejecting cool air out of the RDHX. The RDHX can be configured in two ways: *active* or *passive*. The benefits of RDHX include [1]: *Energy Efficiency* (it can save up to 80% of cooling on the racks and 50% on the overall data center operation); *Heat Removal* (Heat transfer is more extensive since it is very close to the heat source.), *less Space Requirement* (uses minimum floor spaces), *Flexibility* (it has a more basic operation and is easier to install) and *ow maintenance* (light and less frequent efforts and needed). Heat is removed from a system by putting the coolant in direct contact with hot components, and circulating the heated liquid through heat exchangers. Figure 3 gives an overview of how an RDHX works.

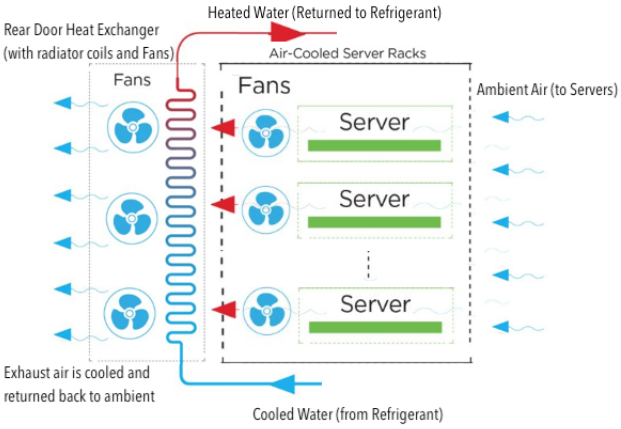
Heated Water (Returned to Refrigerant)
Rear Door Heat Exchanger
(with radiator coils and Fans)
Air-Cooled Server Racks
Fans
Fans
Ambient Air (to Servers)
Server
Server
Server
Exhaust air is cooled and
returned back to ambient
Cooled Water (from Refrigerant)

Figure 3: RDHX main principle

- **Immersion cooling** considers a direct immersion of the hardware in a dielectric (but thermally conductive) liquid (also called *coolant*), which then circulated through heat exchangers. The system absorbs the heat (from the source) and drives it out to the environment through a single-phase or two-phase system[101]. Some advantages of the immersion cooling include a *high heat transfer coefficient, stable hydrodynamic flow*, and *fast/direct cooling with liquid*. While immersion cooling can theoretically deliver the highest performance and PUE, it is seen by some ones as too troublesome for their HPC installations, as it can make the replacement of components tricky[27].

6.2 News trends in liquid cooling design

The market for high-performance cooling systems has grown significantly as technology has shifted from simple air cooling to solution using liquid (including immersion cooling). Water is still the standard for most HPC users as it provides a good balance between performance and set-up cost.

While there are various variants, the basic concept remains the same. Water is pumped through a closed system up to a back plate placed near the hot components. Water has a better thermal conductivity, so this is potentially a higher performance system, but requires additional infrastructure. For example, many water-cooled data centres have a raised floor, so so that all the pumps can be conveniently routed and driven to the targets.

Meyer and Wettig. [89] developed *iDataCool*, an HPC architecture based on *IBM's iDataPlex* platform, whose air-cooling solution was replaced by a custom water-cooling solution. A significant portion of the energy spent on HPC systems can be recovered in the form of chilled water, which can then be used to cool other parts of the computing center. The authors illustrated the cooling performance and the energy reuse efficiency through benchmarks.

Nonaka et al. [93] provided a quantitative and systematic analysis of the impact of the cooling water temperature onto HPC infrastructures. They evaluated the efficiency of the hot water cooling technique, taking into consideration not only the energy reduction on the facility side (cooling system), but also the impact on the power consumption and on the performance degradation from the machine side. They shwoed that, contrary to the gain in the energy consumption, on the HPC facility side, when using higher temperature cooling water, there is an increase in the number of nodes suffering from performance degradation, especially at synchronization barriers.

Ljungdahl et al. [81] developed a *decision support model* that takes basic information regarding a given cluster or data center as inputs and provides a parameterized output that shows different configurations and design parameters that can be utilized for the system. The main outputs include *energy savings, cost savings* and *efficiency gains* through the Power Usage Efficiency(PUE) and the Energy Reuse Efficiency(ERE). An electricity saving between 8.14% and 10.8% and a waste heat recovery of 85 to 576 MWh/year were obtained in a Danish case study. Additional system configurations beside existing local heating source showed an energy saving of 332%. The goal of the decision support model is to assist the design of future waste heat recovery applications through selection of system parameters including coolant temperatures, energy storage design parameters, District Heating supply temperatures and District Heating load coverage from the data center or HPC cluster.

7 ENERGY OPTIMIZATION TECHNIQUES

The power optimization techniques aim at minimizing the energy consumption besides traditional metrics like computing time or memory space. This concern is crucial when there is a power constraint as when the available energy is limited or its supply is costly. Power optimization can be addressed through hardware and software approaches, considering static or dynamic strategies. Figure 4 gives an overview of existing energy optimization techniques grouped by their nature.

7.1 Static energy optimization approaches

- *Hybrid CPU design* : Hybrid design in CPU is an approach that combine low power/performance and high power/performance cores. This was introduced by ARM with BIG.LITTLE architecture and similarly considered more recently by Intel with "Lakefield" chip [60]. Intel 12th generation CPUs (family code name: "Alder Lake") are designed following this hybrid model for energy saving and battery long life for laptop computers.

- *Programming languages efficiency* : Pereira et al [107] studied the energy efficiency of 27 programming languages, monitoring their performance using ten different programming problems. Out of these 27 selected languages, Python ranked 26. Python used 59x more energy than the most efficient language, which is the C language. Nowadays, Python might be the best choice is many cases, for instance when building and training neural networks. There is a significant potential energy saving when considering a more energy efficient language. The authors showed interesting findings such as slower/faster languages consuming less/more energy and how memory usage influences energy consumption.

- *Programming aspects* : A practical approach to C++ was presented in the work by Meyers et al. [90] that describes the basic rules followed by experts (i.e., the things they do or avoid as much as possible) to produce clear, correct, efficient code. This is a static optimization technique for time-to-solution and energy-to-solution by considering the best programming practices.

- *Machine learning prediction models* : Gao et al. [41] developed a neural network framework that learns from actual operating data to model plant performance and accurately predict the PUE. The results demonstrated that machine learning is an effective way of leveraging existing sensor data to model DC performance and improve energy efficiency.

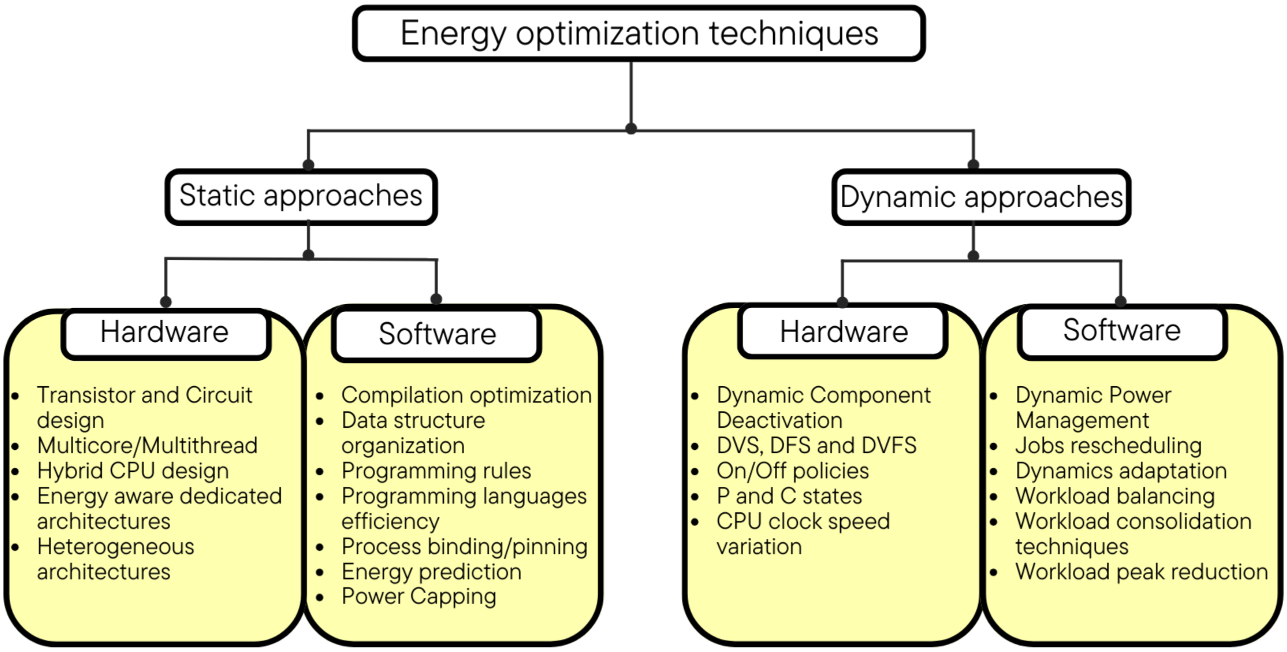

Figure 4: Taxonomy of power/energy optimization techniques in computer system.

- *Circuit design* : the work by Beloglazov et al. [13] presents Static Power Management (SPM) techniques that contains all optimization methods applied at the design time at various levels including *circuit, logic, architecture* and *system.* Circuit level optimizations consist in the reduction of switching in logic-gates and combination circuits through a complex gate design and transistor sizing.

- *Energy-aware dedicated architectures*: This category is for hardware level methods, which consider incorporating power optimization in the design process [13]. In other words, an efficient mapping of high-level specifications into the design of the chip is applied. Apart energy-aware hardware design, it is important to carefully consider a skillful programming that efficiently take into account the energy specificities of the target system. Most often, dedicated architectures (GPU, FPGA, TPU, etc.) are used for specific kernels based on the aforementioned observations.

- *Analytical models prediction for scheduling* : Tadonki et al. [128] designed a combinatorial energy-ware methodology for an efficient management of power states of the RAM. The authors considered traditional techniques like tiling to improve the efficiency of the proposed methodology. Experimental results through simulations considering two well-known algorithms (*Transitive Closure* in graphs theory and *Fast Fourier Transform*) illustrated the efficiency of their strategy, with 98% energy reduction related to memory accesses for the Transitive Closure [109]. Another analytical model for memory energy minimization was proposed by Tadonki and Rolim [127] based on a formal model that captures the relation between energy and memory mechanisms into a *mathematical programming* form (such an approach might also lead to a non-differentiable optmization formulation [10]). They successfully evaluated their approach considering the model of a standard RDRAM so as to figure out the behavior of each parameter together with the energy that can be saved or lost. Singh et al. [122] developed algorithmic techniques for memory energy reduction by exploring the structure and data access pattern to devise an efficient memory power management schedule. They investigated and discussed optimality considering theoretical lower bounds on memory energy. Simulations results demonstrated significant energy reduction over other existing approaches.

- *Compilation and programming best practices* : Quantitative modeling of energy prediction is one of the more attractive approach for static performance optimisation, as it could be used for power aware programming or compilation. Power optimization schemes can be incorporated into compilers by exploiting recurrent patterns in programs[79] and the energy cost of individual programming instructions. For the latter, Leite et al. [75] proposed a fine grained approach for power analysis and prediction, with a focus on a set of basic programming instructions (*addition, multiplication, division, memory read, memory write, memory copy, print, comparison, malloc*). The authors used a series of micro-benchmarks to measure the energy cost per operation considering both the overhead of the embedding loop and that of associated optimizations. Their results showed a 9.48% error rate on the energy prediction of a sorting algorithm. However, their work do not consider specific instructions like the FMA (Fused Multiply Add), which is now available in most of the major CPUs for floating-point operations performance consideration.

- *Power Capping* : Power capping is a technique used for setting a power threshold to not be exceeded by the considered hardware unit. Manufacturers use to provide appropriate tools to handle power capping: NVIDIA System Management Interface (NVIDIA), RAPL (Intel), Energyscale (IBM), and APM (AMD). Cabrera et al. [22] performed an analysis of the performance and the energy efficiency using power cap technologies with a selection of various applications. They illustrated the case of the Intel power cap and the NVIDIA power limit technologies. They extracted a Pareto front of the configurations that lead to the most efficient energy usage for the best possible performance. They provide a methodology base on energy-performance trade-off selection. But there is no investigation on how to automatically select the best trade-off.

7.2 Dynamic energy optimization approaches

- DPM (Dynamic Power Management) techniques : DPM techniques are approaches that include methods and strategies for run-time adaptation of a system's behavior according to current resource requirements or any other dynamic characteristic of the system's state. A paper by Beloglazov et al. [13] presents these techniques with the assumption that enabling DPM allows dynamic adjustment of power states according to current performance requirements. Another assumption is that the workload can be predicted. The authors described different levels for DPM techniques based on hardware (resp. software) considerations. Hardware DPM can be different from one hardware component to another, but they are usually classified as Dynamic Performance Scaling (DPS) such as *Dynamic Voltage and Frequency Scaling (DVFS)* and partial or complete *Dynamic Component Deactivation (DCD)* during idle periods. Some software DPM techniques (Intel RAPL and Nvidia-msi for instance) apply hardware DPM in accordance with the system's power management.

- DFS (Dynamic Frequency Scaling) and DVS (Dynamic Voltage Scaling) adjust the frequency and the power of computing devices (i.e, CPU, GPU, FPGA) by scaling the clock frequency/voltage according to the execution of memory or compute-bound application kernels [125]. So a significant reduction of the total power consumption can be achieved with different voltage/frequency reduction levels. But 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). For this case, DVFS was proposed.

- DVFS (Dynamic voltage and frequency scaling) : DVFS is a technique to systematically adjust the power through a dynamic adjustment of both voltage and frequency settings of a computing device, controller chips and peripherals in order to optimize resources allocation for the tasks and maximize power saving when those resources are not needed. Morán et al. [91] evaluated a series of strategies that can be applied to improve energy efficiency when a failure occurs. This strategy uses the *Advanced Configuration and Power Interface (ACPI).* They considered the use of DVFS techniques and system hibernation at the node level. They estimated the execution time and the waiting time of processes that do not fail through a characterization of the energy consumption required to execute the application and its communication pattern. They considered a simulator to conduct their experiments so as to not get bothered by the issues of a real system and thus focus on the essential.

- Workload consolidation techniques : Sanjeevi et al.[115] proposed an extensive background and motivation of workload consolidation techniques in the cloud computing context [76, 115]. In their paper, they described four recent workload consolidation algorithms considering the goal of reducing energy. In addition, the features of the best workload consolidation algorithm is highlighted.

- On/off policies : shutdown policies stand an appealing approach to dynamically adapt the active resource configuration to the actual workload by tuning off unused components in order to reduce power consumption. However, there are some important constraints to take into account for these policies like the cost (time and energy) of switching on and off, the power and energy consumption bounds caused by the electricity grid or the cooling system, and the availability of renewable energy. Benoit et al. [14] studied the existing approaches that are based on these policies and proposed models for translating the energy constraints into different shutdown policies that can be combined for a multi-constraint purpose.

- Workload peak reduction : Sai et al.[113] presented a space-time multiplexing (STM) power management technique implemented through DVFS for workload balancing. The physical design parameters are based on 130nm CMOS process with TSV models. Experiment results showed that their approach can lead to a peak of 38.10% power reduction and 2.60x workload balancing.

7.3 Hybrid energy optimization approaches

Vaddina et al.[132] proposed a workflow for energy and temperature profiling on systems running parallel applications. They did their experimentations standard multi-core processors using common benchmark applications. Their strategy allows full and dynamic runtime control so as to keep the frequency of the processors within a predetermined range. By this way, they showed that the energy response to frequency scaling is highly dependent on the workload characteristics and it is a convex fonction around the optimal frequency point. Another interesting result from their work is the fact that the tested low-power processor was consuming more power on average than the other standard processors. Their investigation surely contribues to the understanding of power dissipation and its link with temperature as the necessary first step towards optimizing the energy efficiency of HPC systems.

Grant et al.[50] presented a taxonomy of power profiling techniques on modern HPC platforms. The authors used three HPC mini-applications for analysis on three production HPC systems to examine meaningful details, scope, and complexity of the selected energy profiling techniques. Their work demonstrates that a combination of out-of-band measurement with in-band profilers can provide a detailed and accurate view of power usage with almost no overhead.

Jafari-Nodoushan et al. [67] proposed a heuristic battery-aware scheduling policy for periodic and non-periodic real-time tasks under DVS mechanism, with an explicit consideration of power leakage. They compared the battery consumption of their proposed policies with an optimal solution, which could be derived via Calculus of Variations (CoV). Experimental results showed a maximum of 17.7% (resp. 11.3%) battery charge saving for non-periodic (resp. periodic) tasks in comparison to the critical frequency method.

8 ENERGY OF AI PROCESSING

8.1 Motivation

Training a single AI model can emit as much carbon as five cars in their lifetimes [137]. Yet, this analysis pertained to only a one-time training run. When the model is improved by repetitive training, the energy cost is significantly greater. Many large companies, which can daily train thousands and thousands of models, are taking the energy issue more seriously. The work by Strubell et al.[124] describes and analyses the problem by exploring AI's environmental impact, studying ways to address it, and issuing calls to action.

Cutting-edge AI models have nowadays billions of parameters and more. One popular case, GPT-3, has 175 billions of machine learning parameters. The model was trained on NVIDIA V100, but researchers have estimated that the full training would have cost 34 days and $4.6 millions with 1024 A100 GPUs. While energy usage has not been disclosed, it's estimated that GPT-3 consumed 936 MWh[2]. As the models get bigger and bigger in order to handle more complex tasks and the huge volume of requests, the demand for high-end servers to process the models grows exponentially.

Since 2012, the computational resources needed to train cutting-edge AI systems have been doubling every 3.4 months [32]. This escalation in energy use/requirement stands against the common promise of reaching carbon neutrality in the coming decade.

8.2 Studies on CO_2 concerns with AI

Qiu et al.[108] provided a pioneer systematic study of the carbon footprint of federated learning. They proposed a rigorous model to quantify the carbon footprint, thereby facilitating any investigation of the relationship between federated learning design and carbon emissions. They showed that federated learning can emit up to two orders of magnitude more carbon than centralized machine learning. However, in some settings, both approaches can be comparable because of the low energy consumption of embedded devices. Their work highlighted future challenges and trends in federated learning about reducing its environmental impact considering algorithms efficiency, hardware capabilities, and stronger industry transparency.

Luccioni et al.[82] provided an estimate of the carbon footprint of BLOOM, a 176-billion parameter language model, over its lifetime. The authors estimated that BLOOM's final training emitted approximately 24.7 tons of CO_2e for the dynamic power consumption only, and 50.5 tons for all processes ranging from equipment manufacturing to the operational phase. The energy requirement and carbon emission of its deployment for inference via an API endpoint receiving user queries in real-time was also studied. The authors also discussed the difficulty of estimating accurately the carbon footprint of ML models and reported future research directions that can contribute to improving carbon emission.

Patterson et al. [104] studied the carbon footprint of large-scale neural network training and discussed about opportunities to improve energy efficiency and CO_2 emission. The authors estimated the energy consumption and the carbon footprint of several recent large models: T5, Meena, GShard, Switch Transformer, GPT-3, and Evolved Transformer. Their study illustrate that the choice of *neural network architecture, datacenter*, and *processing unit* can reduce the carbon footprint by 100-1000x. The authors also highlighted the need for more focus on how to improve emission metrics in addition to accuracy. Addressing these concerns lead to a reduction of the carbon footprint of ML through accelerating innovations in the efficiency of the algorithms, systems, hardware, datacenters, and in carbon free energy.

Wu et al.[137] studied optimizations techniques for operational energy footprint reduction across Facebook's AI applications. Their work showed improvements on different standpoints: model, platform, infrastructure, and hardware. They described optimization techniques on Platform-Level Caching and showed an improvement of power efficiency by 6.7× with application-level caching. Another optimization is GPU acceleration. In addition to caching, deploying across GPU-based specialized AI hardware unlocks an additional 10.1× energy efficiency improvement. Algorithmic optimizations provided an additional 12× energy efficiency reduction. Considering half-precision (e.g., going from 32-bit to 16-bit operations) provided a 2.4× energy efficiency improvement on GPUs. Another 5× energy efficiency gain was achieved by using custom operators to schedule encoding steps within a single kernel of the Transformer module.

Patterson et al.[103] presented some best practices to reduce the energy of ML training by up to 100x and CO_2 emissions by up to 1000x. The authors recommended that ML papers should explicitly include CO_2e to foster competition not only on model quality. Publishing emissions ensures accurate accounting. They showed that for large-scale ML deployments, minimizing emissions from training should not be the unique as subsequent steps like serving also count. Approaches like *neural architecture search* increases emissions but lead to more efficient serving and a strong overall reduction of the carbon footprint of ML. The work also highlighted the carbon footprint to be erased entirely if cloud providers could fully consider renewable energy (it is already the case with Google and Facebook, and will soon be the case with Microsoft Azure). Another interesting insight from their work is that published studies overestimate the cost and carbon footprint of ML training because they didn't have access to exact information or because they extrapolated point-in-time data without accounting for algorithmic or hardware improvements.

Ludvigsen [83] demonstrated the difficulty in determining the environmental impact of Machine Learning as a field. Moreover, he showed how easy it is for practitioners to estimate the carbon footprint of their machine learning models with tools like *CodeCarbon*[11] or *ML CO2 Impact*[73]. In addition, 17 concretes ideas on how to reduce the carbon footprint of machine learning models are also presented. Some of these ideas can be easily implemented, while others require more efforts and expertise. Indeed the energy profiling of AI applications is a serious focus worth investigating [24, 116].

The ecosystem of sustainable AI is presented and commented by Zhao et al. [142]. They presented an overview of various areas for potential changes and improvements from the standpoint of operational and hardware optimizations for HPC systems considering AI workloads. Three aspects covering the main issues from a micro-to-macro perspective analysis are proposed: infrastructure and resource utilization, user and behavior, and the community of the researchers and practitioners. They showed that concerted and unified efforts are required in order to make effective the transition to a greener ecosystem for AI researches and practices.

8.3 Energy profiling tools for AI applications

Several tools have been developed in recent works about estimating the carbon footprint of machine learning models. These tools estimate the carbon footprint from energy consumption *estimates* or *measurements*.

8.3.1 *Tools that operate from energy estimates.*

- *ML CO2 Impact* [73]: This is a tool that calculates the amount of raw carbon emissions and an estimate of the offset carbon emissions. The latter value depends on the grid used by the cloud provider. About the estimation, it does not take into account the datacenter PUE (Power Usage Effectiveness).
- *Green Algorithms* [74]: An online tool which enables users to estimate and report the carbon footprint of their computation. The tool easily integrates with given computational processes as it requires minimal information and does not interfere with the considered code, while also accounting for a broad range of hardware configurations. With power-hungry and expensive

training algorithms coming from cutting-edge AI, the tool is worth considering for the address the underlying energy concern.

8.3.2 Tools that operate from energy measurements.
- *Codecarbon* [11]: *Codecarbon* is a (lightweight) Python package that estimates estimates the amount of carbon dioxide (CO2) produced by a given code. It achieves that purpose by estimating the electricity power consumption (GPU + CPU + RAM) of the device and weighting it with the local carbon intensity (i.e. where the computing is actually done). The tool thus enables developers to track CO2 emissions across ML experiments or other programs. Power consumption will be successfully tracked only if there are RAPL files within the indicated directory. If not found, CodeCarbon will switch to a *fall back* mode.

- *Tracarbon* [133]: *Tracarbon* is a Python library that tracks the energy consumption and thereby estimates carbon emissions. It detects automatically the key information like the location and the hardware type before starting the tracking. Tracarbon is a flexible tool designed to easily include other platforms, cloud providers, carbon emission APIs, or other data exporters through a Command-line interface (CLI) with already defined metrics or programmatically with the API by defining the desired metrics.

- *Eco2AI* [18]: *Eco2AI* is a python library for CO_2 emission tracking. It monitors energy consumption of CPU and GPU devices and estimates equivalent carbon emissions by taking into account the local carbon intensity. Eco2AI is applicable to any python script. All emissions data together with information about the device are recorded in a local file.

- *Experiment-impact-tracker* [54]: *Experiment-impact-tracker* is defined as a toolkit for tracking energy, carbon, and compute metrics for machine learning (or any other) experiments. The tool runs under Linux system on Intel chips and NVIDIA GPUs for which it records information related to carbon emissions.

- *Carbontracker* [6]: *Carbontracker* is a tool for tracking and predicting the energy and carbon footprint associated to the training of deep learning models. The output result includes *duration, energy*, and *carbon footprint* of training a given deep learning model with the main parameter (specified by the user) being the *number of monitored epochs.* The tool forecasts the *carbon intensity* related to the electricity production during the predicted duration, then uses it to predict the carbon footprint. At the preliminary stage of the development of the tool, a basic linear prediction model is considered.

- *Zeus* [141]: *Zeus* is an online optimization framework for DNNs (Deep Neural Network) training workloads. The tool provides the Pareto frontier for energy-time consumption trade-off and allows users to navigate around by automatically tuning the batch size and GPU power limit of their jobs. Zeus uses an online exploration approach in conjunction with just-in-time energy profiling, thus overcoming the need for offline measurements, while adapting to data drifts over time. The authors shows that Zeus can improve the energy efficiency of DNNs training by 15.3%–75.8% for diverse workloads.

8.4 Energy optimization of AI applications

Cutting-edge AI models require a huge number of parameters and imply a noteworthy computing load, making them being considered as cumbersome from the standpoint of the classical complexity references (running time, memory, and energy). In addition, AI applications are expected to be intensively used both at the level of a single user for routine issues or a collective level (i.e. server mode). Thus, there is clear need for optimization techniques for the implementation (training and inference) and the deployment of large AI models on low-power devices considering their limited hardware characteristics. This section enumerate and describe the mains techniques considered in the literature to cope with energy issues related to AI applications (design, implementation and execution). Those techniques can be grouped considering the following major categories: *Quantization, pruning, filters compression, matrix factorization, neural architecture search, knowledge distillation*, and *hardware selection.*

8.4.1 Quantization.
One of the most popular energy-aware approach of deep learning optimisation is *quantization.* Quantization is a technique to reduce the computational and memory costs by considering low-precision data types (e.g. 8-bit integer) instead of the ordinary ones (e.g. 32/64-bit floating point). For instance, inference could be implemented by representing the weights and activations with low-precision data types. Thus, quantization stands as a technique to speed up inference through running with quantized operators. Quantization can be also considered for the training phase in a so-called *quantization-aware training* approach. There several advantages of quantization including: *more compact model representation; wider vectorization (SIMD); less memory storage; less energy consumption (potentially), faster computation, deployment on embedded devices.* Popular Deep learning frameworks like TensorFlow and PyTorch provide a quantization API to simplify the quantization process. Gholami et al. [43] provided a survey of quantization techniques for efficient deep neural networks.

8.4.2 Pruning.
Pruning is technique applied on inference to get models of smaller in size, thus, similarly to quantization, it yields to better memory/energy/processing complexity with minimal loss in accuracy. Removing less important parameters and connections from an original deep neural network can clearly reduce the volume of memory accesses and associated computations. In addition to the aforementioned advantages, pruning might allow for the execution of the considered model in low-end devices such as mobile/embedded devices. Yang et al. [140] have shown in their work that an energy-aware pruning technique for AlexNet and GoogleNet can reduce energy consumption by 3.7.

8.4.3 Filters compression.
Convolution kernels are the bulk of the computations in DNNs, and the fully connected layers contain around 89% of the parameters in DNNs like AlexNet[45]. To reduce the power consumption of DNNs, the research efforts have focused on reducing the arithmetic operations in the convolution layers and the number of parameters in the fully connected layers. The so-called bottleneck architecture [53] can significantly reduce memory and latency requirements of DNNs . For most computer vision tasks, these techniques preserve accuracy. Filter compression is orthogonal to pruning and quantization techniques. The three techniques can be used together for a combined optimization approach to reduce energy consumption.

8.4.4 Neural architecture search.
There are many different network architectures and optimization techniques to consider when designing low-power AI applications. Neural architecture search (NAS) is a technique for automating the design of artificial neural networks (ANN). Many works address the reduction of computational cost and environmental impact of deep learning by accelerating neural network

architecture search and hyperparameter optimization. Frey et al. [40] introduced a framework called *training performance estimation* (TPE), which is based on existing techniques for estimating the speed of the training process. Ranking models (by estimated performance) without training to convergence leads to a potential saving of up to 90% of time and energy of the full training budget. Some variants of *early stopping* that generalize common regularization technique to account for energy costs were also proposed, and this approach enables significant energy savings across the entire pipeline of model development and deployment. Narsin et al. [92] proposed ENOS (Energy-Aware Network Operator Search in Deep Neural Networks) approach to address the *energy-accuracy trade-off* of a deep neural network acceleration. The search in ENOS is formulated as a *continuous optimization problem* that is solvable using gradient descent methods. This lead to a minimal overhead in the training cost when learning both layer-wise inference operators and weights. ENOS improves accuracy by 10–20% in comparison to the conventional uni-operator search approaches *under the same energy budget*. ENOS also outperforms the accuracy of comparable mixed-precision uni-operator implementations by 3-5% for the same energy budget. Some other works based on the *splitting steepest descent* algorithm for fast energy-aware neural architecture optimization were also proposed [135, 138].

8.4.5 *Knowledge distillation.*

Knowledge distillation refers to the approach of transferring the knowledge from a large but unwieldy model or set of models to a single smaller model that can be deployed under real-world constraints. In many recent publications on the topic, the teacher/student analogy is used to describe how knowledge distillation learning models work. There are three different ways that the larger teacher model is used to help training the smaller student model: *response-based knowledge, feature-based knowledge* and *relation-based knowledge* [49]. Through a varying combination of these three techniques, it has been shown that some very large models can be migrated to smaller representations. Probably the most well-known of these is DistilBERT [114], which is able to keep 97% of its language understanding versus BERT, while having a model which is 40% smaller and 60% faster.

8.4.6 *Hardware selection for training.*

Using processors that are optimized for ML training such as tensor processing units (TPUs) and recent GPUs (e.g. V100 and A100) instead of general-purpose processors can improve performance/watt by factors 2 to 5 [103].

9 PROSPECTIVE VIEWPOINT

Traditional Microcontrollers works fine under normal conditions with no need for external cooling mechanism. Some applications demand a high computational power from Microcontrollers, so it is important to keep control on heating as they are fragile devices. However, general purpose computers, accelerators and modern microcontrollers like coral dev board need explicit cooling. More generally, the correlation between computing power and energy efficiency has led to the choice of low-power computing units at the expense of the potential clock speed. Even if the gap between the peak performance and the sustained performance is a genuine argument in favor of the fact that running applications will not really suffer from this choice, the evolution of HPC cannot just keep it that way. Cooling systems have to be more efficient and scalable. Hardware cooling may be handled at the chip level in the not-too-far future. Quantum cooling is also in the agenda.

The deployment of large-scale servers provides an opportunity to build performance/energy measurement and optimization tools to ensure intensive utilization of the underlying infrastructure with efficiency and robustness. All details have to be taken into account. Indeed, for instance, *static power consumption* plays a non-trivial role in the context of the overall data center electricity footprint, thus the need for a an effective management of processor idle state.

The increasing adoption of low-power processors, often called System-on-Chip (SoC) or microcontrollers (originally designed for the embedded systems and mobile devices) is still justified in the context of the so-called *embedded HPC*, given their increasing computing potential at low cost and low electrical power. However various limitations are to be taken into account like 8, 16 or 32bit-only architectures, small memory (RAM and caches), high-latency interconnections, and the unavailability of Error-Correcting Code Memory. Some low-power designs are reducing the processing performance gap with high-end processors at competitive costs, while keep the traditional advantage of low-energy thus a reduced carbon footprint. For these reasons, such devices (like Arduino or Raspberry Pi) are widely used for equipping IoT systems.

10 CONCLUSION

Energy concerns have an increasing priority for mainly two reasons. The first reason comes from the standpoint of “energy as a cost and/or constraint”. The cost of the necessary energy to keep HPC systems running with all surrounding aspects including cooling is becoming significantly high, especially with large-scale infrastructures. The need for speed, which is primary goal of HPC, leads to the choice of faster computing units for which the design is mainly guided by processing speed regardless of energy aspects. This is for instance case with GPUs. Training cutting-edge Machine Learning algorithms are handled with large-scale GPU(-enhanced) clusters and their usage is entering into the routine by an increasingly large community (like with the case of ChatGPT), these two facts clearly exacerbate the energy concern. A complementary fact in this aspect is the “energy as a constraint” standpoint. Embedded systems and mobile platforms are typically battery-powered, thus they run with a fixed amount of energy, which thereby stands as a critical resource. Many applications including AI ones are intended to run on such systems to address common issues, thus the importance of energy efficiency at all levels (supply and consumption). The second reason for the focus on energy is CO_2 emission concern with all its consequences. Designing energy-aware solutions is very important and this can be done with several approaches including *algorithms design, programs implementations, run-time monitoring tools, compilation, hardware mechanisms, system policies*, and more, beside *energy supply* and ways to cap *heat dissipation* and *CO_2 emission.* As HPC is actively moving on through noteworthy processing performance and devices diversity, addressing energy concerns and related aspects is genuinely a crucial topic we should focus on and even anticipate.

ACKNOWLEDGEMENTS

We wish to warmly thank the Transition Institute 1.5 (tti.5 - https://the-transition-institute.minesparis.psl.eu/en/) for its support, which has led to this work and the underlying initiatives.

REFERENCES

[1] AKCP, 2021. Rear-door heat exchanger for high-density data center. URL: https://www.akcp.com/articles/rear-door-heat-exchanger-for-high-density-data-center/. accessed: 2023-06-25.

[2] Alberto, R., 2021. Meet m6 - 10 trillion parameters at 1% gpt-3's energy cost. URL: https://towardsdatascience.com/meet-m6-10-trillion-parameters-at-1-gpt-3s-energy-cost-997092cbe5e8. accessed: 2023-04-21.

[3] AMD, 2021. Amd instinct mi200 series accelerator. URL: https://www.amd.com/system/files/documents/amd-instinct-mi200-datasheet.pdf. accessed: 2023-04-21.

[4] AMD, I., 2023. Amd epyc energy efficiency. URL: https://www.amd.com/en/campaigns/epyc-energy-efficiency. accessed: 2023-05-16.

[5] Anne-Cécile, O., 2021. Measuring the energy consumption of hpc systems. URL: https://ecoinfo.cnrs.fr/wp-content/uploads/2021/12/ORAP-2021-Orgerie.pdf. accessed: 2023-04-21.

[6] Anthony, L.F.W., Kanding, B., Selvan, R., 2020. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems. ArXiv:2007.03051.

[7] Arduino, 2023. Arduino portenta h7 product overview. URL: https://store.arduino.cc/products/portenta-h7. accessed: 2023-05-11.

[8] Arm, 2020. How supercomputer performance and power efficiency can coexist. URL: https://armkeil.blob.core.windows.net/developer/Files/pdf/case-study/arm-riken-fugaku-supercomputer.pdf. accessed: 2023-06-12.

[9] Arne, T., 2022. Tdp and acp for energy estimation in processors. URL: https://www.green-coding.berlin/blog/tdp-and-acp/. accessed: 2023-04-21.

[10] Babonneau, F., Beltran, C., Haurie, A., Tadonki, C., Vial, J.P., et al., 2007. Proximal-accpm: a versatile oracle based optimization method. Optimisation, Econometric and Financial Analysis 9, 69–92.

[11] BCG, G., 2020. Codecarbon. URL: https://mlco2.github.io/codecarbon/index.html. accessed: 2023-05-26.

[12] Belady, C., Rawson, A., 2008. Green grid data center power efficiency metrics: Pue and dcie.

[13] Beloglazov, A., Buyya, R., Lee, Y., Zomaya, A., 2010. A taxonomy and survey of energy-efficient data centers and cloud computing systems. Advances in Computers 82. doi:10.1016/B978-0-12-385512-1.00003-7.

[14] Benoit, A., Lefèvre, L., Orgerie, A.C., Raïs, I., 2018. Reducing the energy consumption of large scale computing systems through combined shutdown policies with multiple constraints. International Journal of High Performance Computing Applications 32, 176–188. URL: https://inria.hal.science/hal-01557025, doi:10.1177/1094342017714530.

[15] Boku, T., 2022. How fpga can contribute to hpc ?, in: 2022 International Symposium on VLSI Design, Automation and Test (VLSI-DAT), pp. 1–1. doi:10.1109/VLSI-DAT54769.2022.9768098.

[16] Boussaada, Z., Curea, O., Remaci, A., Camblong, H., Mrabet Bellaaj, N., 2018. A nonlinear autoregressive exogenous (narx) neural network model for the prediction of the daily direct solar radiation. Energies 11. URL: https://www.mdpi.com/1996-1073/11/3/620, doi:10.3390/en11030620.

[17] Browne, S., Deane, C., Ho, G., Mucci, P., 1999. Papi: A portable interface to hardware performance counters.

[18] Budennyy, S., Lazarev, V., Zakharenko, N., Korovin, A., Plosskaya, O., Dimitrov, D., Akhripkin, V., Pavlov, I., Oseledets, I., Barsola, I., et al., 2023. Eco2ai: carbon emissions tracking of machine learning models as the first step towards sustainable ai, in: Doklady Mathematics, Springer. pp. 1–11.

[19] Businessinsider.in, 2022. Supercomputers are faster and more powerful — but need to be more energy-efficient. URL: https://www.businessinsider.in/tech/enterprise/news/supercomputers-have-become-faster-and-more-powerful-but-making-them-energy-efficient-is-the-need-of-the-hour/articleshow/92495353.cms?. accessed: 2023-06-12.

[20] C-DAC, 2022a. Eert(energy aware rescheduling tool). URL: https://www.cdac.in/index.aspx?id=hpc_gc_EERT_Tool. accessed: 2023-04-21.

[21] C-DAC, 2022b. Pmac(power monitoring and controlling) tool. URL: https://www.cdac.in/index.aspx?id=hpc_gc_P_MAC_Tool. accessed: 2023-04-21.

[22] Cabrera, A., Almeida, F., Blanco, V., Nieves, D., 2019. Finding energy efficient hardware configurations under a power cap.

[23] Cardoso, J.M., Coutinho, J.G.F., Diniz, P.C., 2017. Chapter 2 - high-performance embedded computing, in: Cardoso, J.M., Coutinho, J.G.F., Diniz, P.C. (Eds.), Embedded Computing for High Performance. Morgan Kaufmann, Boston, pp. 17–56. URL: https://www.sciencedirect.com/science/article/pii/B9780128041895000028, doi:https://doi.org/10.1016/B978-0-12-804189-5.00002-8.

[24] Caspart, R., Ziegler, S., Weyrauch, A., Obermaier, H., Raffeiner, S., Schuhmacher, L.P., Scholtyssek, J., Trofimova, D., Nolden, M., Reinartz, I., Isensee, F., Götz, M., Debus, C., 2022. Precise energy consumption measurements of heterogeneous artificial intelligence workloads, in: Anzt, H., Bienz, A., Luszczek, P., Baboulin, M. (Eds.), High Performance Computing. ISC High Performance 2022 International Workshops, Springer International Publishing, Cham. pp. 108–121.

[25] Chaudhry, M.T., Ling, T.C., Manzoor, A., Hussain, S.A., Kim, J., 2015. Thermal-aware scheduling in green data centers. ACM Computing Surveys (CSUR) 47, 1–48.

[26] Colin, Ian, K., 2021. powerstat - a tool to measure power consumption. URL: https://manpages.ubuntu.com/manpages/bionic/man8/powerstat.8.html. accessed: 2023-05-26.

[27] scientific computing.com, 2019. Cooling technology options for hpc. URL: https://www.scientific-computing.com/feature/cooling-technology-options-hpc. accessed: 2023-06-12.

[28] Conficoni, C., Bartolini, A., Tilli, A., Tecchiolli, G., Benini, L., 2015. Energy-aware cooling for hot-water cooled supercomputers, in: 2015 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1353–1358. doi:10.7873/DATE.2015.1100.

[29] Czarnul, P., Proficz, J., Krzywaniak, A., et al., 2019. Energy-aware high-performance computing: survey of state-of-the-art tools, techniques, and environments. Scientific Programming 2019.

[30] D'Agostino, D., Merelli, I., Aldinucci, M., Cesini, D., 2021. Hardware and software solutions for energy-efficient computing in scientific programming. Scientific Programming 2021, 1–9. doi:10.1155/2021/5514284.

[31] Daniel, B., Rhonda, A., Andy, L., Jacqueline, D., 2021. 2021 data center industry survey results. URL: https://uptimeinstitute.com/2021-data-center-industry-survey-results. accessed: 2023-04-21.

[32] Dario, A., Danny, H., 2018. Ai and compute. URL: https://openai.com/research/ai-and-compute. accessed: 2023-04-21.

[33] David, G., 2013. Swap space watts and power. URL: https://www.energystar.gov/ia/products/downloads/Greenhill_Pres.pdf. accessed: 2023-04-21.

[34] Djedidi, O., Djeziri, M.A., 2020. Power profiling and monitoring in embedded systems: A comparative study and a novel methodology based on narx neural networks. Journal of Systems Architecture 111, 101805. URL: https://www.sciencedirect.com/science/article/pii/S1383762120300953, doi:https://doi.org/10.1016/j.sysarc.2020.101805.

[35] EEMBC, 2014. An eembc benchmark. URL: https://www.eembc.org/ulpmark/ulp-cp/. accessed: 2023-04-21.

[36] EEMBC, 2023. About coremark-pro. URL: https://www.eembc.org/coremark-pro/. accessed: 2023-05-23.

[37] Erich, S., Jack, D., Horst, S., Martin, M., 2022. Top500. URL: https://www.top500.org/lists/top500/2022/11/. accessed: 2023-04-21.

[38] Firefox, 2023. Energy estimates. URL: https://firefox-source-docs.mozilla.org/performance/perf.html. accessed: 2023-05-26.

[39] Freitag, C., Berners-Lee, M., Widdicks, K., Knowles, B., Blair, G.S., Friday, A., 2021. The real climate and transformative impact of ict: A critique of estimates, trends, and regulations. Patterns 2, 100340. URL: https://www.sciencedirect.com/science/article/pii/S2666389921001884, doi:https://doi.org/10.1016/j.patter.2021.100340.

[40] Frey, N.C., Zhao, D., Axelrod, S., Jones, M., Bestor, D., Gadepally, V., Gómez-Bombarelli, R., Samsi, S., 2022. Energy-aware neural architecture selection and hyperparameter optimization, in: 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 732–741. doi:10.1109/IPDPSW55747.2022.00125.

[41] Gao, J., 2014. Machine learning applications for data center optimization.

[42] Ge, R., Feng, X., Song, S., Chang, H.C., Li, D., Cameron, K.W., 2010. Powerpack: Energy profiling and analysis of high-performance systems and applications. IEEE Transactions on Parallel and Distributed Systems 21, 658–671. doi:10.1109/TPDS.2009.76.

[43] Gholami, A., Kim, S., Dong, Z., Yao, Z., Mahoney, M.W., Keutzer, K., 2021. A survey of quantization methods for efficient neural network inference. ArXiv abs/2103.13630.

[44] GlobalPetrolPrice.com, 2022. Global electricity prices september 2022. URL: https://www.globalpetrolprices.com/electricity_prices/. accessed: 2023-05-10.

[45] Goel, A., Tung, C., Lu, Y.H., Thiruvathukal, G.K., 2020. A survey of methods for low-power deep learning and computer vision. 2020 IEEE 6th World Forum on Internet of Things (WF-IoT) , 1–6.

[46] Google, 2023. system architecture tpu v4. URL: https://cloud.google.com/tpu/docs/system-architecture-tpu-vm. accessed: 2023-04-21.

[47] GoogleLLC, 2020a. Dev board datasheet. URL: https://coral.ai/docs/dev-board/datasheet/. accessed: 2023-05-05.

[48] GoogleLLC, 2020b. Edge tpu performance benchmarks. URL: https://coral.ai/docs/edgetpu/benchmarks/. accessed: 2023-04-24.

[49] Gou, J., Yu, B., Maybank, S.J., Tao, D., 2021. Knowledge distillation: A survey. Int. J. Comput. Vision 129, 1789–1819. URL: https://doi.org/10.1007/s11263-021-01453-z, doi:10.1007/s11263-021-01453-z.

[50] Grant, R.E., Laros, J.H., Levenhagen, M., Olivier, S.L., Pedretti, K., Ward, L., Younge, A.J., 2017. Evaluating energy and power profiling techniques for hpc workloads, in: 2017 Eighth International Green and Sustainable Computing Conference (IGSC), pp. 1–8. doi:10.1109/IGCC.2017.8323587.

[51] Grid500, 2023. Grid5000:home. URL: https://www.grid5000.fr/w/Grid5000:Home. accessed: 2023-04-21.

[52] Hackenberg, D., Ilsche, T., Schuchart, J., Schöne, R., Nagel, W.E., Simon, M., Georgiou, Y., 2014. Hdeem: High definition energy efficiency monitoring. 2014 Energy Efficient Supercomputing Workshop , 1–10.

[53] He, K., Zhang, X., Ren, S., Sun, J., 2015. Deep residual learning for image recognition. CoRR abs/1512.03385. URL: http://arxiv.org/abs/1512.03385, arXiv:1512.03385.

[54] Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., Pineau, J., 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. arXiv:2002.05651.

[55] Hosseinabady, M., Núñez-Yáñez, J.L., 2018. Dynamic energy management of fpga accelerators in embedded systems. ACM Transactions on Embedded Computing Systems (TECS) 17, 1 – 26.

[56] Ikram, M., 2018. Energy-Efficient GPU-Based High-Performance Computing. Ph.D. thesis.

[57] Ilsche, T., Schöne, R., Bielert, M., Gocht, A., Hackenberg, D., 2017. lo2s — multi-core system and application performance analysis for linux, in: 2017 IEEE International Conference on Cluster Computing (CLUSTER), pp. 801–804. doi:10.1109/CLUSTER.2017.116.

[58] Ilsche, T., Schöne, R., Joram, P., Bielert, M., Gocht, A., 2018. System monitoring with lo2s: Power and runtime impact of c-state transitions, in: 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 712–715. doi:10.1109/IPDPSW.2018.00114.

[59] INRIA, U.o.L., 2019. Welcome to pyjoules's documentation! URL: https://pyjoules.readthedocs.io/en/latest/. accessed: 2023-05-26.

[60] Intel, 2019a. Headlinelakefield: Hybrid cpu with foveros technology. URL: https://www.intel.com/content/www/us/en/newsroom/resources/lakefield.html. accessed: 2023-05-16.

[61] Intel, 2019b. Intel power gadget. URL: https://www.intel.com/content/www/us/en/developer/articles/tool/power-gadget.html. accessed: 2023-05-26.

[62] Intel, 2020. Powertop. URL: https://github.com/fenrus75/powertop. accessed: 2023-05-26.

[63] Intel, 2022a. Intel data center gpu max 1550. URL: https://ark.intel.com/content/www/us/en/ark/products/232873/intel-data-center-gpu-max-1550.html. accessed: 2023-05-10.

[64] Intel, 2022b. Intel neural compute stick 2 (intel ncs2). URL: https://www.intel.com/content/www/us/en/developer/articles/tool/neural-compute-stick.html. accessed: 2023-04-21.

[65] Intel, C., 2022c. Running average power limit energy reporting / cve-2020-8694 , cve-2020-8695 / intel-sa-00389. URL: https://www.intel.com/content/www/us/en/developer/articles/technical/software-security-guidance/advisory-guidance/running-average-power-limit-energy-reporting.html. accessed: 2023-04-21.

[66] Jacqueline, D., Daniel, B., Andy, L., Owen, R., Max, S., 2022. 2022 data center industry survey results. URL: https://uptimeinstitute.com/resources/research-and-reports/uptime-institute-global-data-center-survey-results-2022. accessed: 2023-04-21.

[67] Jafari-Nodoushan, M., Safaei, B., Ejlali, A., Chen, J.J., 2020. Leakage-aware battery lifetime analysis using the calculus of variations. IEEE Transactions on Circuits and Systems I: Regular Papers 67, 4829–4841. doi:10.1109/TCSI.2020.3001064.

[68] Janković, S., Drndarević, V., 2015. Microcontroller power consumption measurement based on psoc. 2015 23rd Telecommunications Forum Telfor (TELFOR) , 673–676.

[69] Jin, X., Zhang, F., Vasilakos, A.V., Liu, Z., 2016. Green data centers: A survey, perspectives, and future directions. ArXiv abs/1608.00687.

[70] Kirk, W., C., 2022. Green500. URL: https://www.top500.org/lists/green500/. accessed: 2023-05-04.

[71] Knüpfer, A., Brunst, H., Doleschal, J., Jurenz, M., Lieber, M., Mickler, H., Müller, M.S., Nagel, W.E., 2008. The vampir performance analysis tool-set, in: Parallel Tools Workshop.

[72] Kocot, B., Czarnul, P., Proficz, J., 2023. Energy-aware scheduling for high-performance computing systems: A survey. Energies 16. URL: https://www.mdpi.com/1996-1073/16/2/890, doi:10.3390/en16020890.

[73] Lacoste, A., Luccioni, A., Schmidt, V., Dandres, T., 2019. Quantifying the carbon emissions of machine learning. arXiv preprint arXiv:1910.09700 .

[74] Lannelongue, L., Grealey, J., Inouye, M., 2021. Green algorithms: Quantifying the carbon footprint of computation. Advanced Science 8, 2100707. doi:https://doi.org/10.1002/advs.202100707.

[75] Leite, A., Tadonki, C., Eisenbeis, C., de Melo, A., 2014a. A fine-grained approach for power consumption analysis and prediction. Procedia Computer Science 29, 2260–2271. URL: https://www.sciencedirect.com/science/article/pii/S1877050914003883, doi:https://doi.org/10.1016/j.procs.2014.05.211. 2014 International Conference on Computational Science.

[76] Leite, A.F., Raiol, T., Tadonki, C., Walter, M.E.M., Eisenbeis, C., de Melo, A.C.M.A., 2014b. Excalibur: An autonomic cloud architecture for executing parallel applications, in: Proceedings of the Fourth International Workshop on Cloud Data and Platforms, pp. 1–6.

[77] Lenovo, 2022. Optimizing power and energy in hpc data centers with energy aware runtime. URL: https://lenovopress.lenovo.com/lp1646.pdf. accessed: 2023-04-21.

[78] Libri, A., Bartolini, A., Benini, L., 2018. Dwarf in a giant: Enabling scalable, high-resolution hpc energy monitoring for real-time profiling and analytics. ArXiv abs/1806.02698.

[79] Lin, C.Y., Kuan, C.B., Lee, J.K., 2013. Compilers for low power with design patterns on embedded multicore systems, in: 2013 42nd International Conference on Parallel Processing, pp. 1052–1060. doi:10.1109/ICPP.2013.125.

[80] Linaro, 2022. The devicetree specification. URL: https://www.devicetree.org/specifications/. accessed: 2023-05-16.

[81] Ljungdahl, V., Jradi, M., Veje, C., 2022. A decision support model for waste heat recovery systems design in data center and high-performance computing clusters utilizing liquid cooling and phase change materials. Applied Thermal Engineering 201, 117671. URL: https://www.sciencedirect.com/science/article/pii/S1359431121010966, doi:https://doi.org/10.1016/j.applthermaleng.2021.117671.

[82] Luccioni, A.S., Viguier, S., Ligozat, A.L., 2022. Estimating the carbon footprint of bloom, a 176b parameter language model. ArXiv abs/2211.02001.

[83] Ludvigsen, K.G.A., 2022. How to estimate and reduce the carbon footprint of machine learning models. URL: https://towardsdatascience.com/how-to-estimate-and-reduce-the-carbon-footprint-of-machine-learning-models-49f24510880. accessed: 2023-05-26.

[84] Maiterth, M., Koenig, G., Pedretti, K., Jana, S., Bates, N., Borghesi, A., Montoya, D., Bartolini, A., Puzovic, M., 2018. Energy and power aware job scheduling and resource management: Global survey—initial analysis, in: 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), IEEE. pp. 685–693.

[85] Mantovani, F., Garcia-Gasulla, M., Gracia, J., Stafford, E., Banchelli, F., Josep-Fabregó, M., Criado-Ledesma, J., Nachtmann, M., 2020. Performance and energy consumption of hpc workloads on a cluster based on arm thunderx2 cpu. ArXiv abs/2007.04868.

[86] Marvell, 2019a. Manufacturing applications on marvell thunderx2. URL: https://www.marvell.com/content/dam/marvell/en/products/assets/server-processors/thunderx2-arm-processors/pdf/marvell-server-processors-thunderx2-manufacturing-applications-on-thunderx2-white-paper-2019-06.pdf. accessed: 2023-05-26.

[87] Marvell, 2019b. tx2mon. URL: https://github.com/jchandra-cavm/tx2mon. accessed: 2023-05-16.

[88] McCraw, H., Ralph, J., Danalis, A., Dongarra, J., 2014. Power monitoring with papi for extreme scale architectures and dataflow-based programming models, in: 2014 IEEE International Conference on Cluster Computing (CLUSTER), pp. 385–391. doi:10.1109/CLUSTER.2014.6968672.

[89] Meyer, N., Ries, M., Solbrig, S., Wettig, T., 2013. idatacool: Hpc with hot-water cooling and energy reuse, in: Information Security Conference.

[90] Meyers, S., 2005. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. Addison-Wesley Professional Computing Series, Pearson Education. URL: https://books.google.fr/books?id=Qx5oyB49poYC.

[91] Morán, M., Balladini, J., Rexachs, D., Rucci, E., 2020. Towards management of energy consumption in hpc systems with fault tolerance, pp. 1–8. doi:10.1109/ARGENCON49523.2020.9505498.

[92] Nasrin, S., Shylendra, A., Darabi, N., Tulabandhula, T., Gomes, W., Chakrabarty, A., Trivedi, A.R., 2022. Enos: Energy-aware network operator search in deep neural networks. IEEE Access 10, 81447–81457. doi:10.1109/ACCESS.2022.3192515.

[93] Nonaka, J., Hanawa, T., Shoji, F., 2020. Analysis of cooling water temperature impact on computing performance and energy consumption, in: 2020 IEEE International Conference on Cluster Computing (CLUSTER), pp. 169–175. doi:10.1109/CLUSTER49012.2020.00027.

[94] NVIDIA, 2016. nvidia-smi - nvidia system management interface program. URL: https://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf. accessed: 2023-04-21.

[95] Nvidia, 2020. Nvidia jetson nano module product details. URL: https://www.seeedstudio.com/NVIDIAr-Jetson-Nanotm-Developer-Kit-p-2916.html. accessed: 2023-05-11.

[96] NVIDIA, 2022. Nvidia h100 tensor core gpu. URL: https://www.nvidia.com/en-us/data-center/h100/. accessed: 2023-05-10.

[97] Nvidia, 2023. Nvidia jetson agx orin for robotics and edge ai applications. URL: https://www.nvidia.com/en-us/lp/embedded-computing/robotics-edge-ai-tech-brief/. accessed: 2023-05-11.

[98] Oleynik, Y., Gerndt, M., Schuchart, J., Kjeldsberg, P.G., Nagel, W.E., 2015. Run-time exploitation of application dynamism for energy-efficient exascale computing (readex). 2015 IEEE 18th International Conference on Computational Science and Engineering , 347–350.

[99] ORNL, 2022. Frontier direction of discovery. URL: https://www.olcf.ornl.gov/frontier/. accessed: 2023-05-04.

[100] ourworldindata.org, 2023. Electricity mix : Carbon intensity of electricity. URL: https://ourworldindata.org/electricity-mix. accessed: 2023-05-11.

[101] Pambudi, N.A., Sarifudin, A., Firdaus, R.A., Ulfa, D.K., Gandidi, I.M., Romadhon, R., 2022. The immersion cooling technology: Current and future development in energy saving. Alexandria Engineering Journal 61, 9509–9527. URL: https://www.sciencedirect.com/science/article/pii/S1110016822001557, doi:https://doi.org/10.1016/j.aej.2022.02.059.

[102] Pandey, P., Gundi, N.D., Chakraborty, K., Roy, S., 2021. Uptpu: Improving energy efficiency of a tensor processing unit through underutilization based power-gating, in: 2021 58th ACM/IEEE Design Automation Conference (DAC), pp. 325–330. doi:10.1109/DAC18074.2021.9586224.

[103] Patterson, D., Gonzalez, J., Hölzle, U., Le, Q., Liang, C., Munguia, L.M., Rothchild, D., So, D.R., Texier, M., Dean, J., 2022. The carbon footprint of machine learning training will plateau, then shrink. Computer 55, 18–28. doi:10.1109/MC.2022.3148714.

[104] Patterson, D.A., Gonzalez, J., Le, Q.V., Liang, C., Munguía, L.M., Rothchild, D., So, D.R., Texier, M., Dean, J., 2021. Carbon emissions and large neural network training. ArXiv abs/2104.10350.

[105] Patterson, J.L.H.D.A., 2012. Computer Architecture: A Quantitative Approach (5th ed.). URL: https://books.google.fr/books?id=v3-1hVwHnHwC&pg=PA22&redir_esc=y#v=onepage&q&f=false.

[106] Paul, A., 2017. Intel unveils movidius myriad x vision processing unit. URL: https://www.tomshardware.com/news/intel-movidius-vpu-ai-inference,35327.html. accessed: 2023-04-21.

[107] Pereira, R., Couto, M., Ribeiro, F., Rua, R., Cunha, J., Fernandes, J.a.P., Saraiva, J.a., 2017. Energy efficiency across programming languages: How do energy, time, and memory relate?, in: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering, Association for Computing Machinery, New York, NY, USA. p. 256–267. URL: https://doi.org/10.1145/3136014.3136031, doi:10.1145/3136014.3136031.

[108] Qiu, X., Parcollet, T., Fernandez-Marques, J., de Gusmao, P.P.B., Gao, Y., Beutel, D.J., Topal, T., Mathur, A., Lane, N.D., 2022. A first look into the carbon footprint of federated learning. arXiv:2102.07627.

[109] Rajopadhye, S., Tadonki, C., Risset, T., 1999. The algebraic path problem revisited, in: Euro-Par'99 Parallel Processing: 5th International Euro-Par Conference Toulouse, France, August 31–September 3, 1999 Proceedings 5, Springer Berlin Heidelberg. pp. 698–707.

[110] Ramesh, U.B.K., Mälardalen, Sentilles, S., Crnkovic, I., 2012. Technical report: Energy management in embedded systems taxonomy.

[111] Rashti, M., Sabin, G., Vansickle, D., Norris, B., 2015. Wattprof: A flexible platform for fine-grained hpc power profiling, in: 2015 IEEE International Conference on Cluster Computing, pp. 698–705. doi:10.1109/CLUSTER.2015.121.

[112] RaspberryPi, 2020. Raspberry pi 4 model b. URL: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. accessed: 2023-05-11.

[113] Sai, M.P.D., Wang, K., Yu, H., 2013. Peak power reduction and workload balancing by space-time multiplexing based demand-supply matching for 3d thousand-core microprocessor, in: 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6. doi:10.1145/2463209.2488950.

[114] Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. ArXiv abs/1910.01108.

[115] Sanjeevi, P., Perumal, V., 2017. Workload consolidation techniques to optimise energy in cloud: Review. International Journal of Internet Protocol Technology 10, 115. doi:10.1504/IJIPT.2017.085190.

[116] dos Santos, D.C., 2022. Understanding the energy consumption of hpc scale artificial intelligence. arXiv:2212.00582.

[117] Schmidt, R.R., Iyengar, M.K., 2009. Server rack rear door heat exchanger and the new ashrae recommended environmental guidelines.

[118] Schöne, R., Ilsche, T., Bielert, M., Velten, M., Schmidl, M., Hackenberg, D., 2021. Energy efficiency aspects of the amd zen 2 architecture. 2021 IEEE International Conference on Cluster Computing (CLUSTER) , 562–571.

[119] Schuchart, J., Gerndt, M., Kjeldsberg, P.G., Lysaght, M., Hořák, D., Říha, L., Gocht-Zech, A., Sourouri, M., Kumaraswamy, M., Chowdhury, A., Jahre, M., Diethelm, K., Bouizi, O., Mian, U.S., Kruzík, J., Sojka, R., Beseda, M., Kannan, V., Bendifallah, Z., Hackenberg, D., Nagel, W.E., 2017. The readex formalism for automatic tuning for energy efficiency. Computing 99, 727–745.

[120] Schöne, R., Schmidl, M., Bielert, M., Hackenberg, D., 2021. Firestarter 2: Dynamic code generation for processor stress tests, in: 2021 IEEE International Conference on Cluster Computing (CLUSTER), pp. 582–590. doi:10.1109/Cluster48925.2021.00084.

[121] Scott, H., 2011. Measuring processor power tdp vs acp : withe paper. URL: https://www.intel.com/content/dam/doc/white-paper/resources-xeon-measuring-processor-power-paper.pdf. accessed: 2023-04-21.

[122] Singh, M., Prasanna, V., 2003. Algorithmic techniques for memory energy reduction, pp. 237–252. doi:10.1007/3-540-44867-5_20.

[123] Stoffel, M., 2021. Approches statiques et dynamiques pour l'optimisation de la consommation énergétique des applications de calcul à hautes performances. (Static and dynamic approaches for the optimization of the energy consumption associated with applications of the High Performance Computing (HPC) field). Ph.D. thesis. Grenoble Alpes University, France. URL: https://tel.archives-ouvertes.fr/tel-03562771.

[124] Strubell, E., Ganesh, A., Mccallum, A., 2019. Energy and policy considerations for deep learning in nlp, pp. 3645–3650. doi:10.18653/v1/P19-1355.

[125] Sueur, E.L., Heiser, G., 2010. Dynamic voltage and frequency scaling: the laws of diminishing returns.

[126] Tadonki, C., 2013. High Performance Computing as a Combination of Machines and Methods and Programming. Habilitation à diriger des recherches. Université Paris Sud - Paris XI. URL: https://theses.hal.science/tel-00832930.

[127] Tadonki, C., Rolim, J., 2004. An analytical model for energy minimization. doi:10.1007/978-3-540-24838-5_41.

[128] Tadonki, C., Singh, M., Rolim, J., Prasanna, V., 2003. Combinatorial techniques for memory power state scheduling in energy-constrained systems, pp. 265–268. doi:10.1007/978-3-540-24592-6_24.

[129] Treibig, J., Hager, G., Wellein, G., 2010. Likwid: A lightweight performance-oriented tool suite for x86 multicore environments, in: 2010 39th International Conference on Parallel Processing Workshops, pp. 207–216. doi:10.1109/ICPPW.2010.38.

[130] UEFI, 2021. Advanced configuration and power interface (acpi) specification, january 2021. URL: https://uefi.org/htmlspecs/ACPI_Spec_6_4_html/. accessed: 2023-05-16.

[131] UpbeatLabs, 2023. Dr. wattson energy monitoring module for arduino, raspberry pi and other maker-friendly microcontrollers. URL: https://www.upbeatlabs.com/wattson/. accessed: 2023-05-23.

[132] Vaddina, K.R., Lefèvre, L., Orgerie, A., 2021. Experimental workflow for energy and temperature profiling on HPC systems, in: IEEE Symposium on Computers and Communications, ISCC 2021, Athens, Greece, September 5-8, 2021, IEEE. pp. 1–7. URL: https://doi.org/10.1109/ISCC53001.2021.9631413, doi:10.1109/ISCC53001.2021.9631413.

[133] Valeye, F., 2022. Tracarbon — track your device's carbon footprint. URL: https://medium.com/@florian.valeye/tracarbon-track-your-devices-carbon-footprint-fb051fcc9009. accessed: 2023-07-12.

[134] Vysocky, O., Beseda, M., Říha, L., Zapletal, J., Lysaght, M., Kannan, V., 2017. Meric and radar generator: Tools for energy evaluation and runtime tuning of hpc applications, in: International Conference on High Performance Computing in Science and Engineering.

[135] Wang, D., Li, M., Wu, L., Chandra, V., Liu, Q., 2019. Energy-aware neural architecture optimization with fast splitting steepest descent. arXiv preprint arXiv:1910.03103 .

[136] Wikichip.org, 2020. Astra - supercomputers. URL: https://en.wikichip.org/wiki/supercomputers/astra. accessed: 2023-05-16.

[137] Wu, C.J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Behram, F.A., Huang, J., Bai, C., Gschwind, M.K., Gupta, A., Ott, M., Melnikov, A., Candido, S., Brooks, D., Chauhan, G., Lee, B., Lee, H.H.S., Akyildiz, B., Balandat, M., Spisak, J., Jain, R.K., Rabbat, M.G., Hazelwood, K.M., 2021. Sustainable ai: Environmental implications, challenges and opportunities. ArXiv abs/2111.00364.

[138] Wu, L., Wang, D., Liu, Q., 2019. Splitting steepest descent for growing neural architectures, in: Advances in Neural Information Processing Systems, pp. 10655–10665.

[139] XENON, S., 2023. Liquid cooling: Exceeding the limits of air cooling to unlock greater potential in hpc. URL: https://xenon.com.au/white-papers/liquid-cooling-exceeding-the-limits-of-air-cooling-to-unlock-greater-potential-in-hpc/. accessed: 2023-06-12.

[140] Yang, T.J., hsin Chen, Y., Sze, V., 2016. Designing energy-efficient convolutional neural networks using energy-aware pruning. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 6071–6079.

[141] You, J., Chung, J.W., Chowdhury, M., 2023. Zeus: Understanding and optimizing GPU energy consumption of DNN training, in: USENIX NSDI.

[142] Zhao, D., Frey, N.C., McDonald, J., Hubbell, M., Bestor, D., Jones, M., Prout, A., Gadepally, V., Samsi, S., 2022. A green(er) world for a.i., in: 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), pp. 742–750. doi:10.1109/IPDPSW55747.2022.00126.