International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I5P120

Volume 74 Issue 5, 295-311, May 2026

Original Article

Powering the AI Era: Sustainable Approaches for Intelligent Computing Across HPC and Embedded Systems

Hajar OUAAROUCH[1*], Safae DAHMANI[1], Kaouthar BOUSSELAM[1], Mouhcine CHAMI[1]

[1]*Institut National des Postes et Télécommunications, STRS Lab., Rabat, Morocco.*

[*]*Corresponding Author : ouaarouch.hajar@doctorant.inpt.ac.ma*

Received: 13 January 2026 Revised: 10 February 2026 Accepted: 10 March 2026 Published: 30 May 2026

Abstract *- The evolution of modern Computing has known, in recent years, a significant rapid growth in performance and scalability. This progress has revealed unprecedented computational capacities while the requirement for energy efficiency is simultaneously increasing, especially for embedded systems. In that context, the utilization of intelligent techniques such as Machine Learning (ML) to improve performance and reduce energy consumption in computationally intensive applications has also been explored as an interesting direction. This survey presents a general assessment of the latest energy-aware high-performance computing trends, focusing overall on intelligent optimization techniques. By leveraging recent advances in architecture innovation, energy-efficient design techniques, and predictive learning methods, this paper presents a discussion of the opportunities and challenges leading to the evolution of green and sustainable high-performance systems. The aim of this work is to inspire and guide future research toward energy-efficient and scalable modern computing infrastructures driven by intelligent learning frameworks.*

Keywords *- Energy Efficiency, Embedded Computing, High-Performance Computing (HPC), Heterogeneous Systems, AI Workloads, Processing-In- Memory, Processing-In-Network.*

1. Introduction

From massive High-Performance clusters to specialized embedded systems, the ever-growing computational demands are increasingly constrained by energy efficiency considerations. As computers are nowadays reaching exascale performance, power and energy consumption can no longer be considered as operational issues only, but as a constraint for scalability and sustainability. According to the International Energy Agency (IEA), [1] Data centers worldwide used around 415 Terawatt-Hours (TWh) of electricity in 2024. This is roughly 1.5% of total global electricity consumption. Moreover, over the last five years, there was 12% growth per year. As modern workloads (i.e., Artificial Intelligence) continue to rise quickly, this value is expected to nearly double again, reaching a figure of 945 TWh in 2030. Ultimately, data centers will account for about 3% of global electricity demand [1], similar to levels that are expected to be over 22 times the annual electricity demand of a country like Morocco, which consumed 42.3 TWh in 2024 [2].

This growth in energy consumption of data centers is principally driven by compute-intensive systems operating at large scales, including emerging exascale supercomputers and AI model training clusters, highlighting the immense scale of future energy challenges in the computing sector. According to IEA, cooling consumes 7-30% of data center energy [1], with McKinsey and the U.S. NREL both estimating that ~40% of total energy use goes to cooling [3, 4], which means that approximately 60% of total energy consumption is for IT equipment, particularly processors, accelerators (GPUs, TPUs), and memory systems. It is notable also that within that 60%, system components like GPUs and DRAM are particularly energy-intensive. With the convergence of HPC, Edge computing, and AI, energy efficiency has become a critical constraint for system design as well as for sustainable scalability.

This challenge is handled through the development of energy-aware systems where heterogeneous architectures are rising as a viable alternative. Heterogeneous systems leverage general-purpose CPUs and energy-efficient accelerators like GPUs or specialized cores (for example, E-cores by Intel [5] to tailor workloads to the most efficient processing entities in terms of performance and power. A well-known example here is Intel's heterogeneous computing platforms, which, integrating performance and efficiency cores, can perform dynamic workload migration to optimize power consumption.

The remainder of this paper is organized as follows: Section 3 establishes the foundational context through background analysis of energy efficiency challenges and the transition from homogeneous to heterogeneous architectures. Section 4 surveys state-of-the-art approaches across systems management techniques such as scheduling algorithms, compiler optimizations, and runtime adaptation, then across architectural innovations, including multicores/MPSoC designs, interconnection topologies, PIM, and PIN. Section 5 analyses emerging trends from next-generation systems like El Capitan.

In section 6, existing simulation tools used for performance and power analysis are analysed and compared based on their performance and accuracy. Finally, section 7 summarizes future research directions toward sustainable modern computing systems, in order to guide developments toward greener and more scalable high-performance Computing. The final section is dedicated to discussing current research gaps and some guidelines for the future.

2. Research Methodology

This paper provides a review of energy-aware techniques within HPC and embedded environments. The survey was conducted using a structured narrative review of recent literature. Most of the presented references are sourced from Google Scholar and Scopus, with a focus on the past decade. To ensure academic rigor, mainly peer-reviewed conference and journal papers are considered. The current survey consists of three main search dimensions, which are (a) Recent technological context of modern Computing, (b) Across-level energy-aware solutions review with real-world case study analysis, and (c) Simulation-based energy evaluation tools.

(a) The first dimension focuses on references highlighting the need to optimize energy in modern computing systems. It requires choosing references from academic as well as industrial sources (e.g., Intel, NVIDIA). Both architecture and application aspects are taken into account in this axis.
(b) From high-level management software to architectural aspects, the search process here focuses on cross-layer and adaptive approaches with relevant energy-saving results. Recently published journal and conference papers that introduce strong experimental results are prioritized. To illustrate real-world efficiency, two of the top-tier supercomputers are selected to provide an insight into optimization techniques and their efficiency under challenging benchmarks.
(c) The state of the art of simulation tools starts from the previously identified literature. The relevance of each tool is extracted from its specific experimental setup. This approach allows an accurate comparison that is aligned with the requirements of the review analysis.

3. Background: Energy Efficiency is a Core Challenge in Modern Systems

Energy efficiency has become a central constraint in modern Computing, whether by only considering data center servers or even supercomputers. This section addresses it from two complementary perspectives: architectural, focusing on how heterogeneous multiprocessors, accelerators, and some innovative paradigms influence performance-per-watt; and application, examining how large-scale AI models and continuous inference workloads drive unprecedented energy demands. These perspectives together highlight the interaction between system design and workload characteristics, defining the challenges and solutions in energy-efficient Computing.

3.1. From Homogeneous to Heterogeneous Architectures

The increase in demand for further computing performance has led to the incorporation of a progressive number of cores in a single chip. However, this approach has not been efficient in building sustainable systems. The homogeneous server with CPU usage only faced problems of overheating, an increase in cooling bills, limitations in system density, and throttling due to heat, which in turn impacts performance. Heterogeneous multiprocessors, which include CPUs, GPUs, and other specialized processors such as FPGAs and ASICs, are a balanced source of performance and power efficiency offered at a reasonable cost [6]. This makes them useful in terms of workload management, where jobs can be assigned to the processors that are most suitable for that task.

In practice, software technology companies have progressively oriented to domain-specific hardware accelerators for performance and energy improvement. Google, for example, designed Tensor Processing Units (TPUs) [8, 9] to accelerate machine learning workloads. These accelerators are now used in several products, including Coral devices for edge AI. Pixel smartphones also feature custom AI chips, such as the Pixel Neural Core and the Tensor SoC, for on-device machine learning [7]. Google TPUs achieved a 30x to 80x improvement in tera-operations of computations per Watt of energy consumed compared to traditional chips.

On the other hand, to meet the rising demand for Video-On-Demand (VOD) and live-streaming across its platforms, Meta has developed a custom in-house ASIC solution specifically optimized for high-speed video processing and transcoding. [10] This custom processor delivers the same video quality while consuming only half the energy required by traditional CPU-based encoding. The NVIDIA DGX family is an additional real example of heterogeneity for energy efficiency. It is designed for large-scale AI models, demonstrating extreme speed, efficiency, and scalability [11, 12]. NVIDIA DGX systems are widely adopted across leading organizations, including OpenAI, Meta, Google, Microsoft, and Tesla, as well as by national laboratories like Argonne National Laboratory, and research universities, including the

University of Florida and La Trobe University. As a major cloud service provider, Amazon has developed its own processor, Graviton, a 64-bit ARM-based chip built on the Cortex-A72 microarchitecture. In their paper [13], the authors Q. Jiang, Y. Choon Lee, and Y. Zomaya have reported encouraging outcomes, showing that there could be potential savings up to 37% in video transcoding workloads for AWS A1 instances with Graviton chips over CPU-based Intel Xeon servers.

In conclusion, as the need for energy efficiency becomes a fundamental design requirement, the need to adopt a heterogeneous approach can no longer be a design choice but a necessity for the future of sustainable and scalable Computing. As heterogeneous architectures can potentially increase processing capabilities depending on the tasks by facilitating specialized processing, there are additional costs associated with handling the additional dimensions introduced.

3.2. New Energy Challenges in a New Area of Applications

The rise of AI-powered applications brings numerous challenges, particularly in handling the massive data deluge generated by modern workloads. Datasets now routinely reach terabyte to petabyte scale, while model sizes have grown from millions of parameters a decade ago to hundreds of billions today. For instance, GPT-3, with 175 billion parameters, required several months of running for training and consumed an estimated 1,287 MWh of electricity [14]. This exponential increase in computation translates directly into unprecedented energy consumption, raising fundamental concerns regarding efficiency and sustainability in either datacentres or embedded AI systems. The exponential leap in performance of datacentres has not come without trade-offs. With the increased computing capacity, the power required to sustain them has increased too, placing energy consumption at the center of HPC design concerns. The tension to balance exascale goals with sustainability in modern systems pushes to prioritize innovations in design architecture: energy-aware architectures, power-efficient accelerators, and workload management strategies. These trends are clearly reflected in recent supercomputing benchmarks and rankings.

Figure 1 shows the evolution of Top500 supercomputers in terms of performance (Flop/s) and energy efficiency over the past 12 years. It also emphasizes an ideal growth rate for the computational performance of the top supercomputers on the list from 2013 to 2025, wherein a record-breaking peak performance of 1.7 exaFLOPS is surpassed by modern supercomputers such as Frontier and El Capitan today. This growth rate is accompanied by efficient energy, expressed in GFlops per Watt, emphasizing their direction towards sustainable high-performance computing solutions. The performance efficiency remains gradually increasing, keeping up with modern supercomputer design, while El Capitan, ranking 1st for TOP500 and 25th for Green500 rankings, respectively, as of Jun. 2025, exhibits better raw performance and better optimized efficiency, supporting the typical performance-energy balance for modern supercomputing developments today.

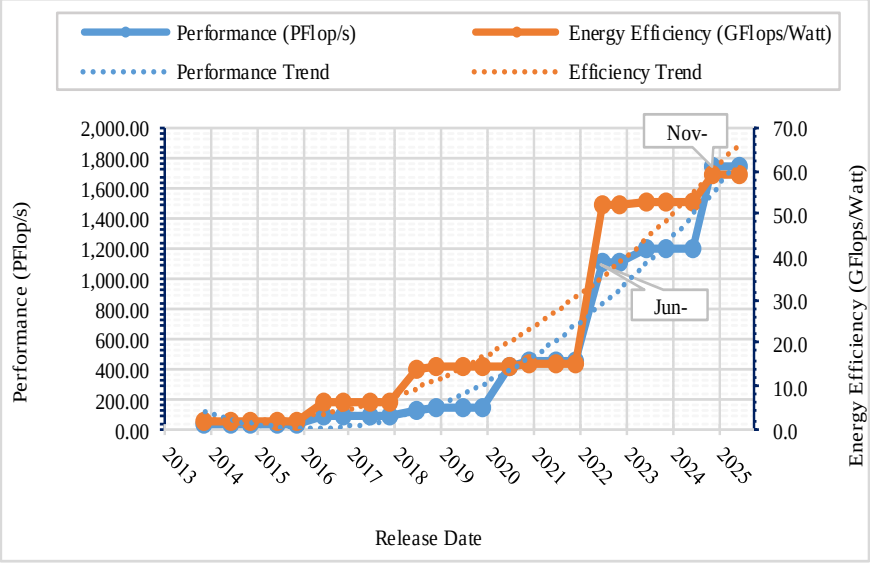

Fig. 1 Trends in performance and energy efficiency of the Top-ranked supercomputers between 2013 and 2025 based on Top500 & Green500 statistics, highlighting the shift toward energy-aware architectures

Building on this observation, it is important to recognize that, even given the steady improvements in energy efficiency, fundamental architectural constraints persistently limit system performance. The most well-known example of this is the long-standing memory wall, as introduced in [15], which is a persistently growing gap between processor speeds and

memory access. In contemporary AI and HPC workloads, this challenge persists but in a more critical form: the energy required to move data across the memory hierarchy often outweighs that of performing the computation itself. For example, the energy overhead of moving data between memory levels is expected to be two orders of magnitude higher than the cost of performing a double-precision floating point operation [16]. Accordingly, the data transfer cost has become a determining factor in the overall efficiency of the system.

Beyond the challenges of data movement, thermal constraints are also viewed as a factor of equal importance influencing the modern computing systems' energy efficiency. As intensive workloads place sustained and high demands on computing hardware, heat dissipation becomes a dominant limitation that directly affects reliability, performance throttling, and overall power consumption. As a result, thermal behavior understanding and optimization are now considered key elements for the design of energy-aware systems.

In this context, recent research has highlighted the importance of thermal-aware workload scheduling and cooling control as complementary strategies to architectural and algorithmic energy optimizations. For example, in the paper [17], the authors worked on a study exploring thermal-aware workload scheduling and fan control and showing direct correlations between thermal strategies and energy savings. This reflects a comprehensive approach for improving energy efficiency in HPC data centers through customized optimization strategies that consider the dynamic interplay between workloads and infrastructure. The main goal of this study is to introduce a holistic methodology for data collection, analysis, and proactive resource and workload management.

For that purpose, the authors introduce a Data Collection Framework (DCF) capable of handling large volumes of heterogeneous data from different sources and formats. Designed for scalability and flexibility, the DCF structure is deployed in the cloud using Red Hat OpenShift. It enables seamless data storage and user access for analysis. Tools such as Amester and IBM Spectrum LSF Explorer are employed for system-wide data acquisition, while machine learning models, principally Long Short-Term Memory (LSTM) networks and Gaussian Processes (GP), are used to predict power consumption patterns and drive energy efficiency optimizations.

The proposed methodology relies on the use of regression-based ML models for the classification and the prediction of workload and infrastructure behavior. It also includes a Decision-Making Framework (DMF) that allows dynamic control of both. The study implements single-agent management for the cluster under examination, noting the potential need for hierarchical agents in larger deployments. Key optimization techniques investigated include thermal-aware scheduling as well as fan speed control policies. An equation is used to quantify improvements in IT-power Usage Effectiveness (ITUE). The authors present two bespoke server-level optimizations, using ITUE as the primary evaluation metric. The results show that the GP model achieved the highest prediction accuracy at 2.1%. Furthermore, the DMF proved effective in interacting with workloads, schedulers, and infrastructure to support real-time or near-real-time optimizations. Notably, power consumption reductions of up to 10% were observed by preferentially assigning larger jobs to cooler nodes. Additionally, fan speed management provided savings up to 44% in terms of energy use related to fan operation. It can be concluded overall that even small optimizations in fan power efficiency (for example, 2% reductions) will provide significant improvements in ITUE (at least 0.5% reductions per node), which highlights the effectiveness of tailored energy-saving approaches in HPC environments.

Although such innovations have reduced the overhead associated with cooling and thermal management, a new limiting factor has emerged: the intrinsic energy cost of computation. Modern AI workloads and HPC systems are further limited by the energy consumption needed for massive computations. For example, the training of current leading models like GPT-3 [18] consumes thousands of GPUs for an extensive period, consuming energy in terms of tens to hundreds of megawatt-hours, thereby causing concerns in terms of operational expense and environmental concerns [19, 20]. For example, exascale supercomputers, which support peak performances above 1.7 exaFLOPS, generally work on power consumption in terms of tens of megawatts, implying massive energy consumption for leading computations [21].

Such a context highlights a key question in future system design, in which, in addition to the system's ability to provide high computation performance, it is important to think about the underlying energy consumption in a feasible and sustainable manner. In other words, to effectively tackle such a problem, there is a need for intelligent approaches, which may include not only energy-efficient processors and accelerators, architecture, and workload schedules, but also intelligent power prediction and adaptive energy management.

4. Approaches for Energy Efficiency at the System Level

From the applicative perspective, energy optimization focuses on how software and workloads take advantage of hardware resources efficiently. AI and ML techniques are employed to partition tasks, schedule jobs, optimize compilation, and dynamically adapt runtime behavior to minimize energy consumption without highly impacting the performance requirements.

4.1. Scheduling and Resource Allocation Algorithms

In terms of energy-efficient scheduling, the allocation of workloads across heterogeneous resources is carefully managed to not only maximize overall system throughput but also to reduce energy consumption, thereby improving the efficiency and sustainability of computing operations.

Tang et al. [22] present a methodology of design-time optimization for achieving energy-aware and high-throughput Convolutional Neural Network (CNN) inference on embedded CPU-GPU MPSoCs. Their approach transforms CNN models into Synchronous Dataflow (SDF) representations that explicitly express task- and data-level parallelism, facilitating efficient mapping onto heterogeneous architectures. A two-objective Genetic Algorithm (GA) is employed to identify Pareto-optimal mappings that jointly optimize throughput and energy efficiency. The inclusion of Voltage And Frequency Scaling (VFS) configurations further enhances energy savings, while analytical models for throughput and power enable rapid evaluation within the optimization process. Experimental results obtained on the NVIDIA Jetson TX2 demonstrate that the proposed framework outperforms conventional deployment frameworks such as TensorRT, achieving superior performance-per-watt characteristics. This contribution highlights the importance of integrating algorithmic optimization with architectural-level configuration to meet the dual goals of high performance and low energy consumption in embedded inference workloads.

In their paper [23], the authors A.Borghesi and A. Bartolini provide a solution for predicting the future power consumption of a job and then scheduling it so that the total power used does not exceed the HPC system's limits. Indeed, they handled that job dispatching challenge within the Eurora supercomputer based on an existing ML model. It was previously developed in other research [24], trained on job Submission parameters to predict CPU power consumption. The total power consumption is assumed to be proportional to the number of additional resources, e.g., GPUs, used and their Thermal Design Power. The study proposes two main strategies for scheduling to improve the energy efficiency within the Eurora system: a heuristic approach based on Priority Rules Based (PRB) scheduling and a hybrid approach combining Constraint Programming (CP) with a heuristic too. The heuristic method consists of a PRB scheduler helping to order jobs based on their expected waiting time and the energy efficiency of nodes. For the hybrid approach, it starts by generating a schedule using a relaxed CP model and then applies the heuristic method again on it. This second approach is used principally to overcome the constraints of using heuristics alone. The authors highlight that experiment results obtained with the suggested solution show an average improvement (for jobs waiting time and Quality-of-Service QoS) with a value of 8.5% in comparison to state-of-the-art techniques [23].

Overall, this solution underlines a smart way to ensure speed, quality, and optimization for job dispatching. However, it is notable that the model could be enhanced by considering not only CPU power consumption, but also further relevant parameters like job-specific characteristics to have a stronger energy efficiency strategy in the scheduling dimension.

4.2. Compiler Energy Optimization

Energy-aware compilation techniques can automatically transform code to minimize power usage, while preserving computational correctness and performance. Z.Wang and M. O'Boyle, in their paper [25], thoroughly examine how Machine Learning (ML) can enhance compiler optimization for energy-efficient HPC applications. They describe approaches where program features, both static (e.g., instruction counts, control-flow characteristics) and dynamic (e.g., cache misses, branch mispredictions), are extracted and used to train predictive models that guide the tuning of compiler optimization passes for each program or workload. The survey highlights Genetic Programming (GP), which evolves sequences of compiler transformations to identify novel energy-efficient strategies, and Reinforcement Learning (RL), where models learn optimal pass sequences through feedback on performance and energy metrics. The authors state that such ML-based approaches can obtain substantial savings in energy consumption with either equivalent or better performance as opposed to the traditional heuristic-based compilers. The authors' analysis proves the flexibility of the compilers with the help of machine learning for varied workloads and machine architectures, while mentioning the challenges of increased cost of training models, interpretability of machine models, and the application of the models on novel machine hardware.

In summary, the paper focuses on the potential of intelligent learning approaches to enable the development of self-adaptive compilers that are energy-conscious. The paper on compiler optimization and machine learning concludes that the application of machine learning to compiler optimization is an area of high potential and promise, and can be seen as an area of major future directions. Adding intelligence to the compiler optimization process through machine learning can thus help researchers and developers make progress on making HPC systems more sustainable.

4.3. Adaptive Runtime

Dynamic runtime strategies monitor system state and application behavior, for instance, adjusting task allocation, frequency scaling, and resource usage to optimize energy efficiency. Using intelligent learning techniques for Adaptive runtime can really help. V. Sundriyal and M.Sosonkina (2022) in their work [26], propose machine learning models to predict performance at runtime under varying core and uncore frequencies. The proposed model uses features including CPU utilization, memory access patterns, and historical performance counters collected at runtime to estimate the

energy-performance trade-off for different frequency configurations. Based on these predictions, the runtime system dynamically adjusts operating points during execution, allowing the multicores application to optimize energy consumption according to the application behavior without significant performance loss.

Experiments conducted on a 28-core HPC node with different workloads, including the GAMESS quantum chemistry package, showed up to 26% of reduction in energy consumption with only a 5% degradation in performance.

This is an important trade-off compared to classic runtime optimizations. This approach shows how intelligent learning methods can enable accurate energy optimization tailored to workload and based on the application behavior in real time and hardware dynamics.

While the targeted platform in this work comprises 28 cores and has proved significant energy saving, further testing at a larger scale is required to confirm the approach. Another limitation is that the use of machine learning models depends on training and may need to be retrained for new applications or hardware platforms, which may limit generality and scalability. Finally, these aspects suggest opportunities for further research to enhance model generalization and runtime adaptability across heterogeneous platforms.

4.4. Binary Acceleration for Energy-Aware Embedded Computing

Binary acceleration of applications is the process of recompiling existing executables with further optimizations for a specific hardware target. Thanks to this technique, instruction-level parallelism can be extensively exploited, leading to significant performance and energy gains.

Paulino et al. review in their paper [27] the existing binary acceleration techniques and report a gain from 1.3x to 3.9x in terms of energy consumption. The binary acceleration approach relies on specialized hardware accelerators to which the binary is translated or instrumented. The translation flow of such a process starts with a static or dynamic analysis of the code to identify 'hot spots' (e.g., loops). Then, it generates a Data Flow Graph (DFG) from the binary, which emphasizes the Instruction Level Parallelism (ILP). Afterward, the DFG is mapped to the targeted hardware units (e.g., FPGA) that take over those specific computations.

This technique enables application adaptation and execution on customized heterogeneous architectures without the need to write complex code. Also, the specialized hardware is fetch- and decode-free, which leads to high energy efficiency. For all of this, binary acceleration lays the foundations to a new generation of highly adaptive, reconfigurable, and energy-aware embedded systems.

5. Architectural Techniques for Energy Optimization

Energy efficiency at the architectural level focuses on optimizing the physical and low-level components of computing systems. By leveraging AI and machine learning, researchers can enhance the performance-per-watt of processors, memory subsystems, and networking interconnections without changing application-level behavior.

5.1. Multicores and Multiprocessor System-on-Chip designs

Recent breakthroughs in Multicores and Multiprocessor System-on-Chip (MPSoC) design have moved attention not only to static power management methods, but also to intelligent and adaptive energy management. The ever-growing heterogeneity in embedded computing platforms, consisting of general-purpose CPUs, GPUs, DSPs, and specialized hardware units, has rendered heuristic approaches alone inadequate to effectively handle energy, performance, and thermal stability issues. Consequently, research has emphasized adaptive, data-driven, and architectural methods that dynamically adjust system behavior to workload and environmental variations.

Hoffmann and Fröhlich propose an online machine-learning-based framework for energy-aware control in multicore real-time embedded systems [28]. Their approach replaces static power-management techniques with a runtime adaptive control loop that leverages hardware performance counters to predict future workload behavior and dynamically adjust architectural parameters, including Dynamic Voltage and Frequency Scaling (DVFS) and core utilization. Experimental results demonstrate an average energy consumption reduction of 24.97%, with a maximum reduction of 30.30%, compared to static policies, while preserving real-time guarantees and introducing minimal runtime overhead.

Building on the concept of learning-based runtime control, other approaches have explored workload-aware DVFS mechanisms to improve energy efficiency in multicore systems further. Gupta and Bhargava propose a regression-based workload-aware DVFS model for multicore processors that adapts voltage–frequency settings based on application phase classification and per-core power budgets [29]. By continuously profiling workloads, their method outperforms traditional static or heuristic DVFS, achieving 33% average power reduction compared to MaxBIPS, and 25% compared to TPEq, while maintaining performance level.

Extending this idea to heterogeneous MPSoCs, Dey et al. propose a coordinated DVFS strategy across CPU, GPU, and memory that jointly adjusts voltage–frequency settings of all major processing elements to match application demands while minimizing energy and thermal costs [30]. Evaluated on an Odroid XU4 platform with an Exynos 5422 MPSoC, this approach achieves approximately 26% improvement in

overall power efficiency and around 21% reduction in peak temperature compared to a literature baseline, demonstrating that multi-domain DVFS can significantly enhance energy and thermal performance in heterogeneous embedded systems. Together, these works highlight the effectiveness of workload-aware and component-coordinated learning-based DVFS in improving energy efficiency beyond traditional static or heuristic approaches.

5.2. Interconnection Topologies

Within existing network architectures, a comparative analysis of interconnection network topologies, primarily Torus, Fat-Tree, and Dragonfly, to identify configurations that optimize the energy consumption and performance trade-off. Using the Hiperion simulator, the authors analyze real-world HPC cluster data and apply power-aware routing (POWAR) techniques in combination with Low Power Idle (LPI) and Power-Down Transition (PDT) strategies to evaluate network energy usage. Their results indicate that the Torus topology achieves the best balance between energy efficiency and performance under high-traffic conditions, while the Fat-Tree topology performs more efficiently under lighter traffic but faces scalability limitations under heavier loads. Evaluations conducted on 64- and 256-node configurations reflect the importance of adapting topological choices to workload and network conditions. These findings show the critical role of designs that account for topology in improving energy efficiency; also, they motivate more research for alternative or hybrid network architectures in HPC systems.

Apart from conventional HPC interconnection networks, energy-efficient interconnection networks emerge as another prominent area of the embedded MPSoC domain, especially after the rise of Network-on-Chip (NoC) architecture. NoCs provide scalable and highly bandwidth-efficient interconnection between heterogeneous cores and accelerators, but the power consumed by the interconnect can form a considerable part of the total energy of the system.

Some latest works have explored adaptive routing, clock gating, and the deployment of voltage islands to reduce dynamic and static power in NoCs [31]. There are other studies investigating approaches considering energy management in topologies, where link activation and routing decisions are dynamically adjusted based on traffic patterns, workload locality, and quality-of-service requirements [32]. Machine-learning-based techniques have also been applied to optimize energy consumption in interconnects. For example, predictive traffic modeling and reinforcement learning controllers have been employed to anticipate congestion, selectively power down idle links, and adjust routing adaptively to minimize energy without compromising performance [33].

Together, interconnect optimization and architecture-aware control form a cross-layer approach to energy efficiency, highlighting the importance of considering both computation and communication subsystems in the design of next-generation embedded and HPC platforms. In the following sections, adaptive and intelligent energy management techniques are explored at the architectural and core level, integrated with interconnect strategies to achieve holistic energy optimization in multicores MPSoCs.

5.3. Processing-In-Memory

Modern processors have shown that the energy expense of moving data over a unit distance surpasses the energy expense of the computation itself, and this disparity grows with scaling. However, the memory-computation wall will become more significant for the newly arising ML workloads, where the massive matrix computations and convolutional layers preserve large data reuse patterns. Therefore, the recently introduced studies have utilized intelligent learning methods to improve the memory system itself to mitigate the related energy and latency penalties. For example, the authors propose in [34] an adaptive learning-based write optimization controller for phase-change memory. It uses neural networks and reinforcement learning to adjust write parameters based on runtime operating conditions continuously. Results show reducing write energy consumption by up to 63% and improving performance by up to 51% compared with static policy baselines. Such adaptive memory control mechanisms demonstrate that ML models can dynamically optimize memory access behavior to improve both energy efficiency and performance. There are further methods, not relying only on the general memory view, but on the architectural approaches.

Processing-in-Memory (PIM) techniques have emerged as one of the most promising architectural responses to the energy and throughput limitations imposed by conventional memory hierarchies. As shown in Figure 2, PIM consists of bringing the computation much closer to where the data is stored. This is achieved by integrating simple processing units inside the memory chip itself. This approach allows for reducing data-transfer overhead by avoiding the constant movement of large data volumes between the processor and memory.

In that context, the Heterogeneous-Hybrid PIM (HH-PIM) framework proposed by Jeon et al. introduces a new approach based on PIM architecture to balance performance and energy efficiency for AI applications [35]. HH-PIM combines two types of PIM modules: high-performance (HP-PIM) units designed for computations that require low latency, and low-power (LP-PIM) units optimized for tasks with strict energy constraints. This configuration presents dynamic workload allocation by assigning tasks and data to the most suited module.

BASED on their computational load and memory access patterns. In the methodology of HH-PIM, a combination of

hardware prototyping and system-level simulations is used. The authors implemented an FPGA-based prototype for proof-of-concept validation and used cycle-accurate simulation to evaluate energy consumption, throughput, and latency under multiple workload conditions. The most important component of the approach is a data-placement and module-activation algorithm, which monitors the dynamic characteristics of inference workloads to determine the optimal mapping of neural network layers to HP-PIM or LP-PIM. This strategy directly targets one of the most critical sources of energy waste in conventional architectures: unnecessary data movement across memory hierarchies.

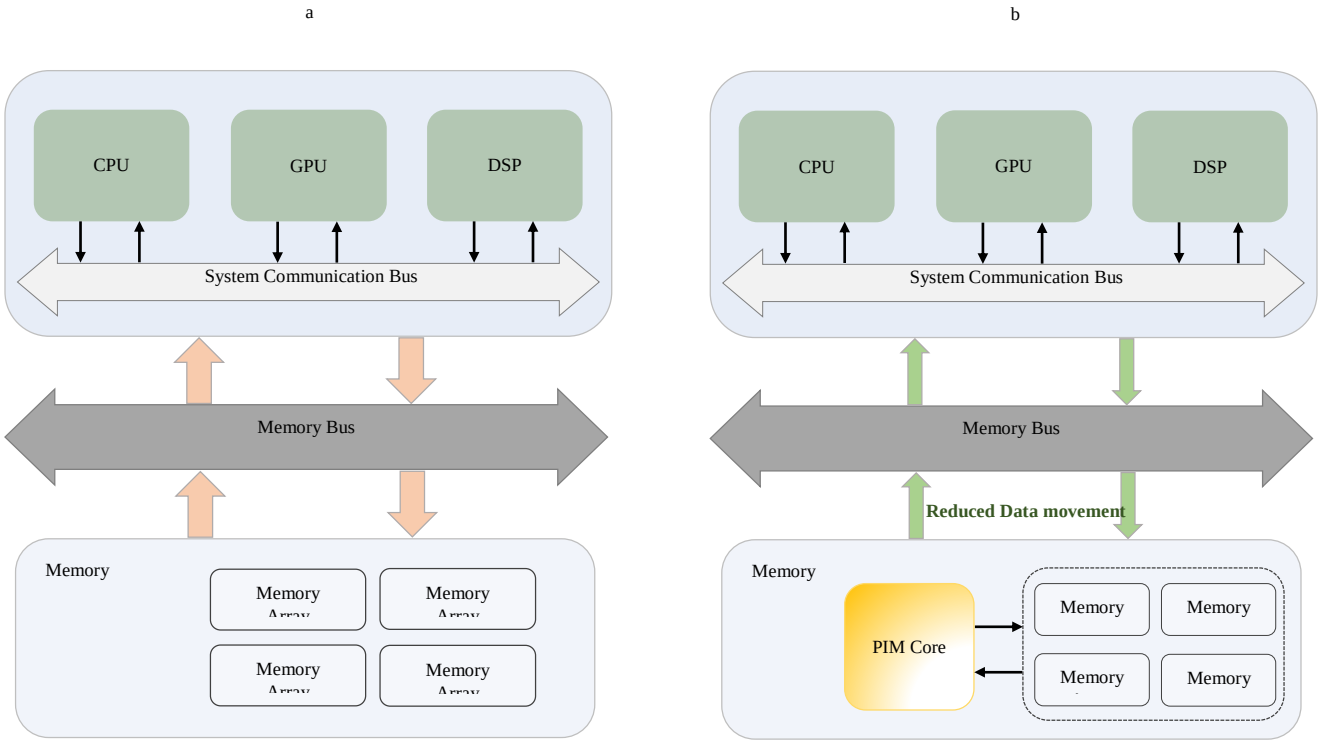

Fig. 2 Comparison between Traditional Architectural Overview (a) and Processing-in-Memory one (b) [36], 2019, Sensors. The figure illustrates how PIM reduces data movement by integrating a computing unit within memory.

The experimental results from that study show its effectiveness. Indeed, HH-PIM allows up to an average of 60.43% for energy savings over traditional PIM systems for standard AI inference benchmarks. In terms of performance, HH-PIM can retain a comparable level of throughput offered by homogeneous PIM designs but with a significantly lower energy consumption. That heterogeneous hybridization is thus proven to be effective in balancing performance and efficiency. However, the HH-PIM framework also highlights areas for potential improvement. The energy savings are, for example, independent of workload, and the actual gain could be lower for workloads with highly irregular memory access patterns. Moreover, the framework relies on accurate runtime monitoring and predictive modeling, which introduces overhead and could limit scalability. In summary, HH-PIM illustrates how heterogeneous PIM architectures can perform adaptive power mitigations of the energy costs associated with memory-bound applications. A combination of HP and LP modules, along with intelligent runtime workload allocation, may therefore indicate a very promising direction for future HPC and edge systems. However, scalability, generality across workloads, and Integration with general system-level energy management strategies still require further efforts.

5.4. Processing-In-Network

Processing-In-Network (PIN) moves selective computation from edge servers and cloud systems into the programmable network infrastructure, namely switches, routers, and NICs. This allows aggregation, filtering, or inference tasks execution in route, closer to the data storage. It reduces redundant network traffic and offloads main computing units. As a result, the overall energy footprint is lowered. The key insight is that by performing lightweight data processing operations in the network, expensive links' bandwidth consumption can be drastically reduced (e.g., Datacentre Interconnects), resulting in important energy savings.

The emergence of the P4 (Programming Protocol-Independent Packet Processors) language for data plane programming has enabled high-throughput ML inference directly in switches. P4 enables programmable packet transformation and supports real-time analytics at speeds

exceeding 10-100~Gbps, with energy benefits realized by avoiding roundtrips to centralized servers [37].

P4 programs specify packet processing pipelines consisting of match-action tables and programmable packet parsing, enabling flexible packet transformation and forwarding decisions. Recent extensions support stateful computation, enabling more complex analytics.

FPGA-augmented programmable switches now support stateful stream analytics, windowed aggregations, and flow-level classification at microsecond latencies, demonstrating their viability for edge AI deployments with significant energy savings [38].

These platforms enable real-time analytics on streaming network data without buffering to centralized servers. Research demonstrates that random forest classifiers, CNNs, and lightweight neural architectures can be mapped onto network devices:

- Flow-Level Inference: Flowrest [39] deploys random forest models on production network switches (Barefoot Tofino ASICs), achieving inference at line rates, reducing datacenter CPU utilization, and saving power via TCAM access reduction. In fact, it can improve the accuracy by more than 10% on average while maintaining sub-microsecond latency.
- CNN Inference in Switches: Quark [40] framework deploys CNN inference tasks in-network at line rate (100 Gbps on Tofino switch). Experiments show that the framework achieves an average latency of 42.66 microseconds and 97.3% accuracy, while ensuring no degradation in network throughput or packet forwarding performance. The resulting power savings are proportional to the low usage of SRAM resources on the switch (around 22.7%).

To sum up, computer network programming is a very promising approach that offers many advantages, for example:

- Reduced Bandwidth and latency: Performing the lightweight functions, such as filtering and aggregation at network edges, could eliminate unnecessary long-distance traffic. This will lead to a proportional reduction of energy spent on link traversal and data serialization.
- Distributed Processing: The fact that computation is offloaded to distributed network nodes will improve load balancing and reduce peak power draw.
- Self-driving networks: Network management can completely rely on In-Network data collection and analysis, which leads to effective real-time management decisions and flexible packet forwarding at line-rate, especially with highly variable workloads.

5.5. Comparison and Analysis

To conduct a systematic evaluation of the current literature for applying intelligent learning approaches for optimizing energy usage in the modern computing context, a list of scientific evaluation criteria that takes into consideration methodology as well as further aspects is defined. These criteria have been chosen to facilitate the identification of the trade-offs between energy efficiency, performance, and adaptability across the different computational scopes discussed in the previous subsections. Table 1 summarizes representative approaches, highlighting not only the ML techniques used but also the quantitative impact on energy, performance trade-offs, and the operational context. The criteria used for this structured comparison are the following:

Integration Level: presents the architectural or computational domain addressed by the method.

Key Optimization Techniques: refers to the intelligent learning methodology, AI/ ML, applied for the optimization of energy efficiency.

Performance Impact: describes how well the strategy affects system performance.

Technical Limitations & Disadvantages: highlights the principal weaknesses of the approach.

References: the main publications or technical documentation that the scope in question is referring to.

The comparative Table 1 brings out several key trends in ML-based energy optimization approaches in HPC and embedded systems. First, design-time optimization strategies, such as GA-based compiler tuning or topology assessment, offer high-energy savings with negligible runtime overhead but lack dynamic adaptability.

In contrast, runtime adaptive techniques, primarily RL and online learning, demonstrate flexibility in reacting to workload variability, often achieving 15-26% energy reductions while maintaining minimal performance degradation. Memory and NoC-centric approaches also emphasize localized optimization, the saving in energy that results from reducing data movement and idle power. Thermal-aware approaches enhance other methods by considering total energy efficiency at the system or data center level.

In summary, the different methods highlighted in the table confirm that hybrid approaches employing design-time and runtime intelligence can achieve energy efficiency while keeping the balance with performance, adaptability, and practicality levels.

6. Case Study from TOP500: El Captain Supercomputer

Despite the gains previously mentioned and the exceptional computational capabilities that they are offering, HPC environments face numerous challenges affecting their efficiency, scalability, and reliability, pushing recent technological advances to innovate in overcoming them. The El Capitan supercomputer, ranked 1st in the TOP500 list from November 2024 [41], serves as a prime lens through which current challenges and the technological trends addressing them can be examined. EL Captain was designed to deliver over 1.7 exaFLOPS of peak performance with a comparable power budget to its predecessor, Frontier. Table 2 underlines a comparison between these two supercomputers, highlighting the advances.

Performance achieved through their architectural innovation. On the other hand, Table 3 links each specific challenge with the related practical approach implemented in the El Capitan supercomputer.

Table 1. Taxonomy of representative ML-based approaches for energy-aware systems

Integration Level	Key Optimization Techniques	Performance Impact	Technical Limitations & Disadvantages	References
Architectural & Sub-system	MPSoC Hardware, NoC Design, Network Topologies, and HH-PIM	Maximum Energy Density: Direct hardware optimizations like PIM achieve up to 60.43% average energy savings by reducing data movement. Design-time MPSoC and NoC optimizations provide high performance-per-watt with zero runtime latency.	Lack of adaptability: High design-time cost and limited runtime adaptability. Results are often platform-specific (e.g., Jetson TX2) and rely on simulation-based models that may not generalize to real hardware.	[22, 32, 35]
System & Runtime	DVFS Control, Task Mapping, and Resource Scheduling	Dynamic Elasticity: Effectively balances power and performance in real-time. RL-based and supervised learning models enable energy savings of up to 25% by adapting to fluctuating workloads.	Control Overhead: Requires representative online data for accuracy. Learning convergence time and model stability under unseen workloads remain significant challenges.	[23, 24, 28]
Application & Compiler-Driven	Compilation-Level Tuning and Application-Level Adaptation	Granular Optimization: Compiler-level tuning via Genetic Programming provides substantial savings across toolchains. Application-level predictors can reduce energy by 26% with only a ~5% performance trade-off.	Portability Barriers: Often requires per-application model tuning and retraining for new hardware. Evolved compiler transformations can be difficult to interpret or port between different ISAs.	[25]
Infrastructure & Facility	Thermal Management and Data Center Cooling	Macro-Scale Efficiency: Infrastructure-level controls can reduce fan energy by up to 44% and total facility energy by ~10%.	Integration Complexity: Heavily dependent on sensor fidelity and specific site infrastructure; Integration across different data centers is complex.	[11]

Table 2. A Comparison between El Capitan, ranked 1st in the 2024 Top500 list, and its predecessor Frontier [21]

Feature	El Capitan	Frontier
Linpack Performance (Rmax)	1,742.00 PFlop/s	1,353.00 PFlop/s
Theoretical Peak (Rpeak)	2,746.38 PFlop/s	2,055.72 PFlop/s
Cores	11,039,616	9,066,176
Processor	AMD 4th Gen EPYC 24C 1.8GHz	AMD Optimized 3rd Generation EPYC 64C 2GHz
Operating System	TOSS	HPE Cray OS
Power	29,580.98 kW	24,607.00 kW

Table 3. Key Challenges in Modern Computing Systems and their corresponding Emerging Technologies, illustrated through the El Capitan supercomputer

Challenge	El Capitan Solution	Emerging Trend
Energy consumption and power efficiency	Liquid cooling, chiplet-based design, and energy-aware system management; 29.58 MW power budget	Energy-efficient architectures and thermal-aware computing
Architectural scalability and heterogeneity	AMD MI300A APUs combining Zen4 CPU + CDNA3 GPU cores with unified HBM3 memory (5.3 TB/s)	Heterogeneous Integration and Unified Memory Architectures
Dynamic resource management and workload optimization	AI-driven orchestration for predictive scheduling, energy tuning, and workload adaptation	Autonomous and AI-assisted system management
Data movement and I/O bottlenecks	Unified CPU-GPU HBM3 memory (5.3 TB/s), HPE Slingshot interconnect, and near-node Rabbit storage units for localized caching and burst buffering	Memory- and data-centric architectures, data locality optimization
Sustainability and environmental impact	Fanless 100% liquid-cooled infrastructure powered by renewable energy sources, achieving 58.89 GFLOPS/W	Carbon-neutral HPC and eco-efficient infrastructure co-design

6.1. Heterogeneity and System Complexity

Modern HPC systems are increasingly developed based on heterogeneous architectures to optimize performance and energy efficiency. El Capitan utilizes AMD's MI300A Accelerated Processing Units (APUs), which integrate Zen 4 CPUs and CDNA3 GPU cores into a single package with 128 GB of HBM3 (High Bandwidth Memory3), providing up to 5.3 TB/s as peak theoretical memory bandwidth [42]. These architectural advances show, thus, that heterogeneity is now widely recognized as a prerequisite to achieve exascale capability as well as sustainable operation.

6.2. AI-Driven Resource Management and Optimization

One of the strong points of El Captain is the Integration of AI-driven solutions in the system management. Indeed, while its predecessor Frontier primarily optimized for raw exascale performance with traditional resource scheduling, El Capitan leverages its AI-primed AMD MI300A APUs to enable intelligent and dynamic resource orchestration. The overall system efficiency is significantly enhanced thanks to AI algorithms predicting workload management and optimizing energy consumption through fine-grained power and cooling control. Furthermore, El Capitan supports "AI-coupled" workflows, such as simulation steering, anomaly detection, and in-situ analysis, running alongside traditional exascale simulations on the same nodes. This enables more integrated and efficient scientific Computing than previous systems [43].

6.3. Data Movement and Memory Bottleneck

Data movement is still one of the most energy and latency-intensive operations, and addressing this challenge is essential to achieve exascale performance. To tackle this issue, El Capitan mitigates memory and I/O bottlenecks through a combination of architectural and storage innovations. At the compute level, the AMD Instinct MI300A Accelerated Processing Units (APUs) consist of CPU and GPU cores within a unified package that shares a high-capacity HBM3 memory stack [42]. As a consequence, the supercomputer achieved 5.3 TB/s of theoretical Bandwidth [44] and drastically reduced off-chip data transfers. The machine also utilizes the HPE Slingshot interconnect, which has high Bandwidth, being an Ethernet-based networking fabric that interconnects over 11,000 compute nodes. This directly allows for mitigating the overhead that arises due to communication among the nodes [45]. To complement this, El Capitan brings with it Rabbits, a near-node storage architecture that consists of high-performance flash nodes [46]. This directly helps in reducing data transfer between the processing node and the parallel file system. Therefore, such innovative advancements obviously align with the new emerging innovation for exascale systems, where memory-centric computer system architecture is paramount for greater scalable performance.

6.4. Cooling Solutions for Sustainability

Conventional air-cooling systems, which were common in the early models of supercomputers, require a considerable amount of power usage and often auxiliary cooling infrastructure for the use of fans. The growing performance and power density of modern processors make conventional air-cooling systems inefficient for heat dissipation; for this reason, Direct Liquid Cooling (DLC) becomes necessary for the next-generation HPC systems.

El Capitan represents another technology toward direct liquid cooling as a fundamental design element for energy optimization. It uses a 100% fanless and direct liquid-cooling system that eliminates the parasitic power losses related to conventional air-cooling mechanisms. This architectural choice enables a remarkable energy efficiency metric for the system: achieving 58.89 GFLOPS/W (gigaflops per Watt), good enough for having the 18th place on the Green500 list for November 2024. This ranking shows how the design of the cooling solution can be applied for direct improvement in the power-to-performance ratio, which represents an important performance indicator for operational costs and environmental

factors. During the process, the solution directs coolant flow to the heat-producing components (processors, memory, and interconnects), thus lowering temperature gradients and overall energy operational costs for operating the solution at a safe temperature. By eliminating fans entirely and relying on direct liquid cooling, El Capitan's architecture can decrease cooling power consumption by up to 37% per server blade compared with hybrid liquid/air configurations [47].

6.5. Frontier Vs El Capitan

Table 3 illustrates how El Capitan, as a representative exascale supercomputer, addresses the challenges mentioned above through innovative design choices and emerging technological trends that shape the exascale era.

Power consumption is still one of the biggest hurdles to be overcome for sustainable exascale performance. Frontier, the first official exascale supercomputer, consumes about 21.1 MW to provide 1.194 exaFLOPS [21]. El Capitan is projected to surpass Frontier in performance while maintaining comparable power usage enabled by innovations in liquid cooling systems, chiplet-based architecture, and advanced packaging [41, 44].

7. Energy-aware simulation tools

Simulators for energy awareness are important tools for investigating the efficiency and energy consumption characteristics of HPC systems. However, due to the current trends in architectures, where hybrid systems, memory-centric systems, and sophisticated communication networks are used, simulations become inevitable for analysing their behavior under different settings and workloads.

In fact, existing simulators vary in complexity, scalability, and energy estimation models in terms of architectural details. Therefore, there arises a need for a defined evaluation framework to deduce their suitability for investigating efficiency in an HPC setup.

7.1. Evaluation Attributes for Simulation Tools Comparison

In order to compare the selected simulation tools, a set of evaluation attributes is defined. These attributes describe for each tool the simulation domain, the architectural modeling capabilities, the scalability, the extensibility, and the suitability for performance and energy analysis in HPC environments:

Tool Name: refers to the name of the simulation tool, identified by its name in the scientific publications.

Core Scope & Modeling shows how the simulator represents and simulates the structural and functional components of HPC systems, including describing what hardware and system components the simulation tool can mimic.

Operational method: describes the type of simulation technique used, including discrete event simulation, full system simulation, cycle-level simulation, and trace-driven simulation.

Analytical Capabilities: describes the capability of quantifying execution measured parameters such as latency, throughputs, and model power/energy consumption.

Performance & Energy Metrics: It is about the ability of the tool to quantify execution performance and energy metrics and shows the types of results the tool generates for evaluation, such as logs, traces, statistical data, energy reports, or visualization features, and their utility for scientific analysis and comparison.

Interface & Specialized Features: covers programming interfaces for customization, extensibility features, and specialized features. It is important to clarify that the evaluation presented herein is based exclusively on the documentation and published descriptions of each tool. No empirical testing or benchmarking has been conducted by the authors to validate the tools against these attributes.

Moreover, the selection of evaluation attributes, while grounded in common HPC research concerns, is not derived from a formally standardized scientific framework but reflects a synthesis tailored to the objectives of this comparative study.

7.2. Comparative Study of the Tools

Table 4 lists a set of tools for simulating HPC environments; they are evaluated based on the criteria defined in the previous subsection. The comparative analysis highlights their differences in simulation scope, architectural detail, scalability, extensibility, and ability to support performance and energy assessment in HPC systems.

7.3. Analysis and Interpretation

The comparison of tool utilizations from Table 4 shows that the mapped tools support different layers of the HPC stack. Some of them are cloud- or/and grid-level simulators such as CloudSim and GridSim, and others, such as gem5 and PIMeval, perform processor, memory, or PIM-oriented architectural modeling. This diversity is an explicit sign that a “one-size-fits-all” solution, providing the full coverage of performance and energy analysis across HPC environments, has not yet been achieved. Instead, each framework contributes advantages in a particular simulation domain or at an abstraction level.

CloudSim [49] and GridSim [50] operate at the highest level of abstraction, targeting cloud and grid infrastructures, respectively. Their modeling capabilities are focused mainly on the virtualized resources, broker behavior, and scheduling policies. Although both tools support large-scale simulation and are useful for power-aware schedulers using real or

synthetic traces (e.g., NASA/ClarkNet), their native energy modeling remains limited and mainly CPU-centric. Accordingly, they appear more suitable for evaluating resource allocation strategies and workload management in distributed computing environments but less appropriate for detailed architectural or component-level energy studies.

In contrast, Hiperion [51], VEF Trace Framework [52], and Dimemas [53] handle network and communication behavior within HPC systems. Hiperion provides detailed modeling of interconnect topologies and routing schemes, making it effective for analysing network performance and link-level energy behavior. VEF specializes in workload tracing and communication pattern analysis, offering insights into how real HPC applications stress communication subsystems. Dimemas focuses on application-level simulation through computation and communication skeletons, enabling what-if analyses of parallel applications. However, all three provide limited or no native energy modelling, restricting their use for power-centric studies unless external models are integrated.

Table 4. Summary and comparison of tools used for simulating performance and/or energy consumption in HPC environments

Tool Name	Core Scope & Modelling	Operational approach	Analytical Capabilities	Performance & Energy Metrics	Interface & Specialized Features
CloudSim [49]	Cloud/DC; Virtualized CPU, Memory, & Networks	Discrete events; Large-scale cloud systems	Perf-focused; energy evaluation via extensions (MultiRECloudSim)	Makespan, Energy(J), SLA violations	Java-based; QoS-aware; APIs for virtual resource management
GridSim [50]	Grids/Hetero-resources; CPU (PEs) & shared models	Discrete events; Task-level Gridlets for large systems	Perf-focused; limited native energy (possible extensions); task-level traces	Makespan, Resource util., Hit count; Power (W)	Java APIs; Market-driven schedulers; advance reservation
Hiperion [51]	HPC Interconnects; Topologies, routing, & NoC queues	Discrete events; Scalable to large HPC fabrics	Latency/Throughput; Energy-aware link on/off metrics	Throughput, End-to-end flow latency; Link Power	Custom APIs; Supports Torus, Fat-tree, & Dragonfly
VEF Trace [52]	HPC Workloads; Trace-based characterization	Trace-driven; Replay of workloads	Communication profiling; MPI analysis; No energy native	Load balance, Serialization, Transfer efficiency. Energy savings (%)	Analysis tools for communication pattern profiling
Dimemas [53]	MPI Apps: Communication & computation skeletons	Trace-driven; Application-level simulation	Perf scaling graphs; MPI profiling; Non-native energy [7]	Task mapping efficiency, Scaling curves, Watts/Performance	Script-based; "What-if" analysis & task mapping
Gem5 [54]	Microarchitecture: Detailed CPU/Memory hierarchy	Cycle-accurate; Full-system & CPU simulation	Microarchitecture power/energy (McPAT, NVMain)	IPC, PDP, and cache miss rates	C++/Python; Multi-ISA; Hetero CPU/GPU co-sim
PIMeval [55]	PIM Systems; Processor & memory	Architecture modeling; PIM-specific hardware modeling	Specialized PIM energy evaluation & performance metrics	Near-data energy, PIM throughput	Custom APIs; DRAM-PIM & Near-data workload suites

At the architectural level, gem5 [54] stands out with its cycle-accurate simulation engine and detailed microarchitectural models, enabling precise analysis of CPU, cache, and memory behavior. It is the only tool in comparison with integrated support for energy modeling through frameworks such as McPAT and NVMain. This makes gem5 well-suited for studies examining the impact of architectural modifications, memory hierarchy exploration, heterogeneous compute elements, and fine-grained power-performance trade-offs. Finally, PIMeval [55] targets PIM systems and provides specialized benchmarking and modeling of near-data processing workloads. With native support for PIM-specific architectures and associated energy behaviors, it fills a critical gap left by general-purpose architectural simulators. PIMeval therefore represents an essential tool for analysing emerging memory-centric accelerators and evaluating PIM hardware

design choices, especially in the context of energy efficiency.In summary, the previous comparison highlights that while several tools support large-scale or fine-grained simulation, only a subset offers relevant energy modeling. At the same time, higher-level frameworks handle system-level behavior but lack architectural accuracy, whereas lower-level simulators provide detailed insights but have limited scalability. Alternatively, network-simulation tools offer relevant communication analysis but require complementary models to assess energy consumption. Given this, a comprehensive energy-aware evaluation would often require combining multiple tools or integrating external power models to bridge the gap between scalability and precision.

8. Conclusion

The analysis introduced in this work highlights the important progress achieved in cutting-edge energy-aware systems across various aspects, from architecture-level designs to runtime and compiler-level optimizations. From this review, it is clear that intelligence-driven techniques are crucial in addressing the inherent trade-offs between performance, power, and scalability. In fact, reviewed works illustrate relevant advances in areas such as predictive power modeling, smart heterogeneous systems, compiler-level adaptation, network interconnection optimization, and memory-centric computation. Indeed, these trends confirm that intelligent learning techniques are increasingly influential in workload characterization, anticipation of system behavior, and dynamic autonomous decision-making, with improvements in energy efficiency without compromising computational power.

8.1. Research Gaps and Future Work

8.1.1. Research gaps

From the current literature review, several challenges and limitations related to intelligent energy-aware systems could be highlighted:

- Limited Integration of intelligence into design frameworks: Most existing solutions operate as secondary trained models guiding discrete system components rather than being inherently embedded into the simulation frameworks, architectural design flows, or the hardware computing infrastructure itself. This limits their capacity for continuous, context-aware adaptation and cross-layer optimization.
- Simulation frameworks lack predictive capabilities: Existing tools mostly provide retrospective analysis and fail to enable real-time, automated recommendations for energy-aware configurations.
- Passive PIM utilization: Processing-in-Memory architectures are often treated as accelerators rather than active, intelligent elements capable of autonomously deciding workload placement based on learned performance and energy profiles.

8.1.2. Future Directions

Looking ahead, future directions point toward the creation of self-optimizing HPC systems where energy efficiency is intrinsic rather than supplementary. Different promising directions include:

- Built-in intelligent co-design: Integrating energy management and learning capabilities directly into simulation frameworks, architectural design flows, and hardware/software interfaces will enable automated, context-aware decision-making throughout the system lifecycle.
- Next-generation intelligent simulations**:** The gap mentioned previously for simulation tools point to the need for a new generation of “intelligent simulations” in which AI models are directly integrated within the architecture of simulation tools themselves to accelerate efficient design exploration, offering real-time recommendations for energy-aware configurations without extensive trial.
- Active intelligent PIM architectures: Incorporating machine learning models into PIM design can allow dynamic decisions on whether to execute workloads in memory or in processors, transforming PIM from a passive hardware accelerator into an active decision-making element in heterogeneous systems.
- Predictive and reinforcement learning in architectural design: By anticipating workload patterns, systems can proactively adjust energy consumption from early design stages, achieving better efficiency without compromising performance.

These future research directions highlight a paradigm shift toward self-optimizing intelligent systems in which energy efficiency is a feature built into the system design. Through the adoption of machine learning approaches in architecture, compilation, and runtime, the coming generation of computing systems can realize sustainable performance scaling along with the increasing environmental needs of modern Computing. Thus, the combination of the best of what has been accomplished and what is targeted in future innovations outlines a promising roadmap, in which the Integration of AI, heterogeneity, and energy awareness offers the prospect of new efficiency, scalability, and adaptability levels.

References

[1] Energy Demand from AI, International Energy Agency (IEA), 2026. [Online]. Available: https://www.iea.org/reports/energy-and-ai/energy-demand-from-ai

[2] Electricity, Ministry of Energy Transition and Sustainable Development, Morocco, 2024. [Online]. Available: https://www.mem.gov.ma/Pages/secteur0a89.html?e=1

[3] Energy Consumption in Data Centres: Air versus Liquid Cooling, Eaton, 2022. [Online]. Available: https://www.boydcorp.com/blog/energy-consumption-in-data-centers-air-versus-liquid-cooling.html

[4] Alyssa Bersine, Reducing Data Center Peak Cooling Demand and Energy Costs with Underground Thermal Energy Storage, National Laboratory of the Rockies, 2025. [Online]. Available: https://www.nrel.gov/news/detail/program/2025/reducing-data-center-peak-cooling-demand-and-energy-costs-with-underground-thermal-energy-storage

[5] M. Shamanna et al., "E-Core Implementation in Intel 4 with PowerVia (Backside Power) Technology," *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, Kyoto, Japan, pp. 1-2, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] Qiang Liu, and Wayne Luk, "Heterogeneous Systems for Energy Efficient Scientific Computing," *International Symposium on Applied Reconfigurable Computing*, Hong Kong, China, vol. 1, pp. 64-75, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[7] Norm Jouppi, Quantifying the Performance of the TPU, Google Cloud Blog, 2017. [Online]. Available: https://cloud.google.com/blog/products/gcp/quantifying-the-performance-of-the-tpu-our-first-machine-learning-chip

[8] Norm Jouppi et al., "TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings," *ISCA '23: Proceedings of the 50th Annual International Symposium on Computer Architecture*, Orlando, FL, USA, pp. 1147-1160, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Kiran Seshadri et al., "An Evaluation of Edge TPU Accelerators for Convolutional Neural Networks," *2022 IEEE International Symposium on Workload Characterization (IISWC)*, Austin, TX, USA, pp. 79-91, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] H.M. Reddy et al., "Efficient Video Processing at Scale Using MSVP," *Applications of Digital Image Processing XLVI*, vol. 12674, pp. 1-16, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Matej Spetko, Lubomir Riha, and Branislav Jansik, *Performance, Power Consumption and Thermal Behavioral Evaluation of the DGX-2 Platform*, Advances in Parallel Computing, IOS Press, pp. 614-623, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] NVIDIA DGX Spark™ Founders Edition, Leadtek Research Inc., 2025. [Online]. Available: https://www.leadtek.com/eng/products/ai_hpc(37)/nvidia_dgx_spark_founders_edition(51035)/detail

[13] Qingye Jiang, Young Choon Lee, and Albert Y. Zomaya, "The Power of ARM64 in Public Clouds," *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, Melbourne, VIC, Australia, pp. 459-468, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] Alex de Vries, "The Growing Energy Footprint of Artificial Intelligence," *Joule*, vol. 7, no. 10, pp. 2191-2194, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Xian-He Sun, and Xiaoyang Lu, "The Memory-Bounded Speedup Model and its Impacts in Computing," *Journal of Computer Science and Technology*, vol. 38, no. 1, pp. 64-79, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] Gokcen Kestor et al., "Quantifying the Energy Cost of Data Movement in Scientific Applications," *2013 IEEE International Symposium on Workload Characterization (IISWC)*, Portland, OR, USA, pp. 56-65, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[17] Robert Tracey et al., "Towards Bespoke Optimizations of Energy Efficiency in HPC Environments," *Applied AI Letters*, vol. 4, no. 4, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[18] Nefi Alarcon, OpenAI Presents GPT-3, a 175 billion Parameters Language Model, NVIDIA Corporation, 2020. [Online]. Available: https://developer.nvidia.com/blog/openai-presents-gpt-3-a-175-billion-parameters-language-model/

[19] Ilpyung Yoon et al., "Comparative Study on Energy Consumption of Neural Networks by Scaling of Weight-Memory Energy Versus Computing Energy for Implementing Low-Power Edge Intelligence," *Electronics*, vol. 14, no. 13, pp. 1-19, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[20] Brad Everman et al., "Evaluating the Carbon Impact of Large Language Models at the Inference Stage," *2023 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, Anaheim, CA, USA, pp. 150-157, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[21] El Capitan Retains #1 as JUPITER Becomes Europe's First Exascale System in the 66th TOP500 List, TOP500.org, 2025. [Online]. Available: https://www.top500.org/

[22] Erqian Tang, Svetlana Minakova, and Todor Stefanov, "Energy-Efficient and High-Throughput CNN Inference on Embedded CPUs-GPUs MPSoCs," *International Conference on Embedded Computer Systems*, Samos, Greece, vol. 1, pp. 127-143, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Andrea Borghesi et al., "Scheduling-Based Power Capping in High Performance Computing Systems," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 1-13, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[24] Andrea Borghesi et al., "Predictive Modeling for Job Power Consumption in HPC Systems," *International Conference on High Performance Computing*, Frankfurt, Germany, vol. 2, pp. 181-199, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[25] Zheng Wang, and Michael O'Boyle, "Machine Learning in Compiler Optimization," *Proceedings of the IEEE*, vol. 106, no. 11, pp. 1879-1901, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[26] Vaibhav Sundriyal, and Masha Sosonkina, "Runtime Energy Savings Based on Machine Learning Models for Multicore Applications," *Journal of Computer and Communications*, vol. 10, no. 6, pp. 63-80, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[27] Nuno Paulino, João Canas Ferreira, and João M.P. Cardoso, "Improving Performance and Energy Consumption in Embedded Systems via Binary Acceleration: A Survey," *ACM Computing Surveys (CSUR)*, vol. 53, no. 1, pp. 1-36, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[28] José Luis Conradi Hoffmann, and Antônio Augusto Fröhlich, "Online Machine Learning for Energy-Aware Multicore Real-Time Embedded Systems," *IEEE Transactions on Computers*, vol. 71, no. 2, pp. 493-505, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[29] Manjari Gupta, Lava Bhargava, and S. Indu, "Dynamic Workload-Aware DVFS for Multicore Systems Using Machine Learning," *Computing*, vol. 103, no. 8, pp. 1747-1769, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[30] Somdip Dey et al., "CPU-GPU-Memory DVFS for Power-Efficient MPSoC in Mobile Cyber Physical Systems," *Future Internet*, vol. 14, no. 3, pp. 1-14, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[31] Dongyu Xu et al., "Improving Power and Performance of on-Chip Network through Virtual Channel Sharing and Power Gating," *Integration*, vol. 93, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[32] Mehdi Modarressi and S. Hossein SeyyedAghaei Rezaei, *Power-Efficient Network-On-Chip Design by Partial Topology Reconfiguration*, *Advances in Computers*, Elsevier, vol. 124, pp. 217-255, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[33] Xiaoyun Zhang et al., "A survey of Machine Learning for Network-on-Chips," *Journal of Parallel and Distributed Computing*, vol. 186, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[34] Mahek Desai, Rowena Quinn, and Marjan Asadinia, "SMART-WRITE: Adaptive Learning-Based Write Energy Optimization for Phase Change Memory," *2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC)*, Las Vegas, NV, USA, pp. 00640-00648, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[35] Sangmin Jeon et al., "HH-PIM: Dynamic Optimization of Power and Performance with Heterogeneous-Hybrid PIM for edge AI Devices," *2025 62nd ACM/IEEE Design Automation Conference (DAC)*, San Francisco, CA, USA, pp. 1-7, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[36] Xu Yang, Yumin Hou, and Hu He, "A Processing-in-Memory Architecture Programming Paradigm for Wireless IoT Applications," *Sensors*, vol. 19, no. 1, pp. 1-23, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[37] Huu Nghia Nguyen, Manh-Dung Nguyen, and Edgardo Montes de Oca, "A Framework for In-Network Inference Using P4," *ARES '24: Proceedings of the 19th International Conference on Availability, Reliability and Security*, Vienna, Austria, pp. 1-6, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[38] Jiuxi Meng et al., "Beyond Network Switching: FPGA-based Switch Architecture for Fast and Accurate Ensemble Learning," *Preprints*, pp. 1-24, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[39] Aristide Tanyi-Jong Akem, Michele Gucciardo, and Marco Fiore, "Flowrest: Practical Flow-Level Inference in Programmable Switches with Random Forests," *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, New York City, NY, USA, pp. 1-10, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Mai Zhang et al., "Quark: Implementing Convolutional Neural Networks Entirely on Programmable Data Plane," *IEEE INFOCOM 2025 - IEEE Conference on Computer Communications*, London, United Kingdom, pp. 1-10, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[41] TOP500, November 2024, Top500 list, 2024. [Online]. Available: https://www.top500.org/lists/top500/2024/11/

[42] Using El Capitan Systems: Hardware Overview, Lawrence Livermore National Laboratory, 2025. [Online]. Available: https://hpc.llnl.gov/documentation/user-guides/using-el-capitan-systems/hardware-overview

[43] Alexandra Kelley, El Capitan Supercomputer is Ready to Handle Nuclear Stockpile and AI Workflows, Nextgov, 2025. [Online]. Available: https://www.nextgov.com/emerging-tech/2025/01/el-capitan-supercomputer-ready-handle-nuclear-stockpile-and-ai-workflows/402088/

[44] AMD Instinct MI300A Accelerators, Advanced Micro Devices, Inc., 2025. [Online]. Available: https://www.amd.com/en/products/accelerators/instinct/mi300/mi300a.html

[45] HPE Delivers World's Fastest Direct Liquid-Cooled Exascale Supercomputer El Capitan for LLNL, Hewlett Packard Enterprise Wire, 2024. [Online]. Available: https://www.hpcwire.com/off-the-wire/hpe-delivers-worlds-fastest-direct-liquid-cooled-exascale-supercomputer-el-capitan-for-llnl/

[46] Brian Behlendorf, and Olaf Faaland, Rabbit Storage for El Capitan, Fast I/O through Big, Pointy Teeth, Lawrence Livermore National Laboratory, 2023. [Online]. Available: https://www.opensfs.org/wp-content/uploads/Fast-IO-El-Capitan-Rabbits.revised.pdf

[47] HPE Announces Industry's First 100% Fanless Direct Liquid Cooling Systems Architecture, Hewlett Packard Enterprise, 2024. [Online]. Available: https://www.hpe.com/us/en/newsroom/press-release/2024/10/hpe-announces-industrys-first-100-fanless-direct-liquid-cooling-systems-architecture.html

[48] Janet Morss, El Capitan Takes Exascale Computing to New Heights, Advanced Micro Devices, Inc., 2025. [Online]. Available: https://www.amd.com/en/blogs/2025/el-capitan-takes-exascale-computing-to-new-heights.html

[49] Rodrigo N. Calheiros et al., "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[50] Rajkumar Buyya, and Manzur Murshed, "Gridsim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing," *Concurrency and Computation: Practice and Experience*, vol. 14, no. 13-15, pp. 1175-1220, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[51] Fran Andújar, Hiperion, GitLab, 2026. [Online]. Available: https://gitraap.i3a.info/fandujar/hiperion

[52] Franisco J. Andújar et al., "VEF Traces: A Framework for Modelling MPI Traffic in Interconnection Network Simulators," *2015 IEEE International Conference on Cluster Computing*, Chicago, IL, USA, pp. 841-848, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[53] Dimemas: Predict Parallel Performance Using a Single CPU Machine | BSC-Tools" Barcelona Supercomputing Center, 2025. [Online]. Available: https://tools.bsc.es/dimemas

[54] Nathan Binkert et al., "The gem5 Simulator," *ACM SIGARCH Computer Architecture News*, vol. 39, no. 2, pp. 1-7, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[55] Farzana Ahmed Siddique et al., "Architectural Modeling and Benchmarking for Digital DRAM PIM," *2024 IEEE International Symposium on Workload Characterization (IISWC)*, Vancouver, BC, Canada, pp. 247-261, 2024. [CrossRef] [Google Scholar] [Publisher Link]