DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. Reference herein to any social initiative (including but not limited to Diversity, Equity, and Inclusion (DEI); Community Benefits Plans (CBP); Justice 40; etc.) is made by the Author independent of any current requirement by the United States Government and does not constitute or imply endorsement, recommendation, or support by the United States Government or any agency thereof.

HTTPS://DOI.ORG/10.2172/2476961

BASIC RESEARCH NEEDS IN

Energy-Efficient Computing for Science

September 9-11, 2024

U.S. DEPARTMENT of ENERGY | Office of Science

PREPARED FOR THE U.S. DEPARTMENT OF ENERGY ADVANCED SCIENTIFIC COMPUTING RESEARCH

EECS: Basic Research Needs for Energy Efficient Computing for Science

January 11, 2026

Abstract

In September 2024, the US Department of Energy's Advanced Scientific Computing Research program convened a **Workshop on Energy-Efficient Computing for Science** to address the critical research challenges and opportunities in this field. The workshop brought together experts from academia, government, and industry to explore innovative approaches to improve energy efficiency across the computing stack over the next two decades. Participants identified five priority research directions (PRDs) that emphasize the need for a holistic approach. These PRDs include (a) co-designing energy-efficient hardware devices and architectures for important workloads; (b) defining the algorithmic foundations of energy-efficient scientific computing; (c) reconceptualizing software ecosystems for energy efficiency; (d) enabling energy-efficient data management for data centers, instruments, and users; and (e) developing integrated, scalable energy measurement and modeling for next-generation computing systems. The workshop's findings emphasize that achieving significant advancements in energy efficiency requires a multifaceted strategy. This includes innovations in hardware, algorithms, software, and data management as well as crosscutting efforts in modeling, simulation, and benchmarking. In addition, the participants identified multiple enablers for technical progress in this area, such as computing test beds and software maintainability, which are crucial for rapid progress in this area. **The research directions outlined in the workshop report aim to guide the development of energy-efficient computing technologies that can support future scientific discoveries and address the growing energy demands of scientific computing and AI.**

Contents

Executive Summary

Large-scale computing has enabled numerous scientific discoveries, including ground-breaking achievements facilitated by the US Department of Energy (DOE) supercomputers and advances in applied mathematics and computer science. While important advances were made in energy efficiency to enable exascale computing, continued efforts are needed to meet the energy efficiency necessary for the next generation of high-performance computing (HPC) systems and, more broadly, AI data centers. Without substantial improvements in energy efficiency, the energy consumption associated with computing could become a limiting factor for future scientific discovery, national security, and technological advancement. Therefore, realizing energy-efficient scientific computing remains a key challenge, and significant questions remain. In particular, how will next generation energy-efficient systems be designed, what new algorithms are needed to solve problems of interest on these platforms, how will they be programmed, and how will users manage the data generated by these tremendous computational capabilities in an energy-efficient way?

In September 2024, DOE's Advanced Scientific Computing Research (ASCR) program in the Office of Science convened the Workshop on Energy-Efficient Computing for Science (alongside workshops on analog and neuromorphic computing) to identify research opportunities and grand challenges on this topic. Over three days, the participants identified five synergistic priority research directions (PRDs).

Priority Research Directions

PRD 1: Co-design energy-efficient hardware devices and architectures for important workloads

Identifying promising candidates for energy-efficient devices and bridging device-level and architectural research are essential for the future of scalable scientific computing. These tasks involve taking a broad view of innovative computing paradigms, including analog, stochastic, optical, cryogenic, neuromorphic, quantum, and biological systems. Research on these paradigms must target large scale systems and include end-to-end evaluation of performance, energy use, manufacturing readiness, and yield. To maximize the benefit of specialization, innovation should focus on important computing kernels that are most likely to benefit DOE workloads and then develop synergistic technologies that ease the heterogeneous integration of specialized hardware for science into future scalable heterogeneous production systems. Advancements in synergistic technologies, such as chiplets and chip design software, will lower integration efforts and costs, and access to fabrication facilities and tool chains will aid in accurate estimation and frequent prototyping. By co-designing these components, the DOE community can maximize the scientific impact while achieving unprecedented energy efficiency across diverse computational workloads.

PRD 2: Define the algorithmic foundations of energy-efficient scientific computing

The study of energy efficiency in algorithms requires a mathematical notation that includes the energy costs of data movement and storage. With this new notation, it becomes easier to understand and optimize the energy complexity of different algorithm-hardware combinations and make informed algorithm choices based on the execution environment. By understanding how various aspects of computational workflows influence energy demands, users can optimize not just individual tasks but entire scientific campaigns. Strategies must be developed to adjust algorithmic parameters dynamically, responding to the specific hardware and resource availability in each environment. This approach balances energy costs across the entire lifecycle of scientific computation, including AI training and inference phases in tasks, such as training neural network surrogates. Integration of the algorithmic model with the dynamic heterogeneous hardware environment will provide substantial opportunities for reducing energy consumption across entire scientific campaigns.

PRD 3: Reconceptualize software ecosystems for energy efficiency

Software systems must evolve to achieve energy efficiency across existing and emerging architectures. Effective languages and compilers should produce energy-efficient code, while specialized libraries need to encapsulate hardware complexities without sacrificing efficiency. Runtime systems should dynamically optimize workload distribution with energy use in mind, while operating systems must intelligently balance energy consumption against the needs of performance, security, and reliability. Automating these capabilities will enable the integration of energy-efficient practices across diverse hardware technologies and facilitate development of fresh techniques that streamline abstractions, protocols, and interfaces to minimize energy use across deep software stacks.

PRD 4: Enable energy-efficient data management for data centers, instruments, and users

Data movement constitutes a major energy expenditure in scientific workflows, but current storage systems are largely energy-agnostic and homogeneous, creating significant barriers to energy-efficient scientific computing. To address this, a paradigm shift in data management and storage strategies is imperative for energy-efficient HPC across distributed environments while preserving performance and dependability. Research efforts must span the entire storage infrastructure, including hardware and software stacks. Additionally, integrating innovative energy-efficient storage technologies will be essential to constructing robust data systems that operate within a limited energy budget while maintaining performance targets. Through these efforts, the scientific community can continue to derive vital insights as data volumes grow and experimental and computational infrastructures become more complex and geographically dispersed.

PRD 5: Develop integrated, scalable energy measurement and modeling capabilities for next-generation computing systems

Accurate, scalable modeling tools must quantify and predict energy use across all system levels—from devices to facilities. Current methods remain fragmented, rely on proprietary software, and ineffectively model emerging heterogeneous systems and technologies as parts of larger systems. Innovative simulation and modeling strategies should integrate energy models for diverse hardware, share open interfaces, support co-design of energy and performance, and accelerate innovation in novel computing paradigms. To ensure high confidence, these tools should enable reproducible and accurate results that can be replicated by industry, government, and academia. Similarly, energy measurement methodologies for operational systems and prototypes need to enable real-time, energy-aware optimizations and validate modeling predictions. Partnerships with prototyping facilities will create hardware and software test beds that validate energy models for real applications, refining designs for real-world applicability.

Summary

Energy-efficient computing is vital to the continued advancement of scientific research and the realizable development of cutting-edge computing technologies. As we advance beyond the exascale era, the need to balance computational capabilities with energy consumption becomes increasingly crucial. The Workshop on Energy-Efficient Computing for Science, organized by the DOE's ASCR program, identified research opportunities and grand challenges for this topic. The workshop identified five priority research directions that promise to drive significant progress toward energy-efficient computing. Collectively, solutions to these PRDs will ensure that scientific advancements can be sustained.

Taken as a whole, the workshop findings and recommendations sketch a creative and integrated vision in which new algorithms and programming paradigms create opportunities for scientists to exploit these emerging hardware technologies to address the most challenging scientific problems of the day within responsible energy budgets. In this vision, thoughtful metrics and telemetry enable a deep understanding of how energy is consumed in the system. These metrics can then be fed to detailed and ever-improving system models that can enable runtime decision-making on power utilization and inform the design of future computing platforms.

Moreover, new resource management and system software advances will allow facilities to adapt their computational loads to proportionally match the availability of power on a dynamic grid, and new approaches to storing and positioning increasingly valuable scientific data can improve availability to the scientific community at lower energy costs and in tight coordination with the integrated research infrastructure and distributed scientific workflows. The community is poised and excited to help realize this vision.

1 Introduction

The need for transformative improvements in energy-efficient computing has never been more pressing. As global reliance on computing continues to expand into critical areas such as scientific discovery, AI, and large-scale simulations, the energy consumed by high-performance computing (HPC) systems and data centers is reaching unsustainable levels [1,2]. Over the past decade, the demand for computing power has escalated exponentially, driven by increasingly complex scientific applications, AI, and data analytics—both in centralized data centers and at the edge in ubiquitous applications such as smartphones, automobiles, and entertainment systems. This surge in computational activity has led to staggering energy consumption, with data centers alone estimated to account for between 1% and 2% of current global electricity usage, with projections much higher over the coming decade [3].

Historically, advances in computational power have been closely linked to improvements in hardware energy efficiency. Reductions in energy per operation were largely achieved through innovations in transistor design and materials. However, this trend has slowed considerably. For decades, Moore's Law guided the exponential improvement in computing performance [4]. However, while raw computational capabilities have continued to grow, the associated energy consumption has skyrocketed. The decline of Dennard's scaling [5–7] in the mid-2000s marked a critical turning point: power efficiency no longer improved in proportion to transistor miniaturization, leading to stagnation in energy efficiency gains. Within the US Department of Energy (DOE), past generations of supercomputers achieved milestone performance levels (e.g., teraflop computing with ASCI Red [8] and petaflop computing with Roadrunner [9]) with only modest increases in power consumption. However, today's exascale systems, including Oak Ridge National Laboratory's Frontier, operate at over 21 MW of power, nearly 9× the energy footprint of earlier systems. If left unaddressed, this trajectory will render future supercomputing infrastructures incapable of serving the nation's security, scientific, or economic needs.

Consequently, as gains in conventional hardware energy efficiency diminish, software and algorithms must take on a more central role in addressing this challenge [10]. Optimizing software to reduce redundant computations, improve memory locality, and minimize data movement can lead to substantial energy savings. Algorithmic advancements such as novel algorithms, mixed-precision computing, energy-aware scheduling, and workload-specific optimizations have demonstrated significant efficiency improvements across scientific computing and AI workloads. Furthermore, intelligent runtime systems and adaptive execution strategies can dynamically adjust computation to balance performance and energy consumption.

The challenge is further exacerbated by the increasing role of AI, which are particularly energy-intensive due to their reliance on extensive data handling, complex model training, and large-scale inference across distributed cloud and HPC resources. Although specialized AI accelerators and low-precision computing architectures have been developed to mitigate energy consumption, these solutions will fundamentally depend on conventional CMOS semiconductor technologies for the foreseeable future.

The extreme energy demand of hyperscale commercial AI facilites has prompted suggestions of constucting gigawatt-scale nuclear power plants (and corresponding cooling plants) in conjuction with these data centers. Construction projects of this type and magnitude impose substatial capital, regulatory and scheduling burdens upon hyperscaler and scientific HPC facities alike.

As such, this trajectory poses a significant threat to future advancements in computational science. Addressing this challenge requires more than incremental optimizations—it necessitates a fundamental rethinking of computing paradigms to establish a foundation for long-term innovation. Emerging computational approaches, such as analog, photonic, and cryogenic computing, offer the potential for orders-of-magnitude improvements in energy efficiency. However, realizing these gains demands a holistic strategy that improves energy efficiency across the entire computing stack, encompassing algorithms, software, and hardware. Without substantial breakthroughs, energy consumption will become a bottleneck for future scientific discovery, national security, and technological progress.

1.1 Workshop

In September 2024, DOE's Advanced Scientific Computing Research (ASCR) program convened the *Workshop on Energy-Efficient Computing for Science* to identify the grand challenges and define priority research directions (PRDs) for improving energy efficiency in computing. This workshop convened expert leaders from academia, government, and industry to examine this challenge from multiple perspectives.

The workshop was organized around breakout groups focused on eight topics: (a) algorithms; (b) hardware; (c) data management and storage; (d) modeling and simulation (of computer architectures); (e) system, facility, and edge; (f) resource management; (g) programming systems; and (h) crosscuts. The breakout groups generated a number of findings from their respective discussions (Section 3), which were distilled into **five** PRDs. In addition, the workshop advertised an open call for contributed white papers on this topic. These contributed white papers are archived on OSTI.gov (https://doi.org/10.2172/2506700, forthcoming).

1.2 Related Efforts in Energy-Efficient Computing and Semiconductor Innovation

Given the critical role of computing and semiconductors in the US economy and national security, numerous organizations have undertaken initiatives to address challenges in energy-efficient computing. Several DOE reports [11–15] have informed this discussion by providing foundational insights into computing architectures, materials, and advanced manufacturing. Although these reports do not specifically focus on energy efficiency, they offer valuable perspectives on emerging technologies that influence this domain.

One of the most relevant DOE initiatives is the **Energy Efficiency Scaling for Two Decades (EES2)** report, which was published by the Advanced Materials and Manufacturing Technologies Office. This roadmap envisions a 1,000× increase in microelectronics energy efficiency over the next two decades to address the escalating energy demands of HPC, AI, and data-intensive applications. The report advocates for cross-layer co-design, in which hardware and software innovations are developed in tandem to maximize efficiency. Research priorities include advancements in semiconductor materials, novel computing paradigms such as neuromorphic and quantum computing, and in-memory processing. The roadmap also highlights the need for new energy-efficiency benchmarks, metrics, and test beds to validate emerging technologies in real-world applications and emphasizes collaboration among academia, industry, and government.

The **Semiconductor Research Corporation (SRC)** has also played a pivotal role in shaping research priorities for the future of computing. Two major reports produced by SRC, the **Decadal Plan** and the **MAPT (Microelectronics and Advanced Packaging Technologies) report**, outline critical challenges and research directions in semiconductor technology. The Decadal Plan identifies pressing research needs in areas such as novel materials, energy-efficient computing, and quantum technologies and emphasizes the importance of collaboration between academia, industry, and government to sustain innovation. The MAPT report, on the other hand, focuses on heterogeneous integration and advanced packaging solutions to address the scaling limitations of traditional semiconductor manufacturing. By integrating multiple technologies within a single package, advanced packaging approaches enhance performance, improve energy efficiency, and provide new avenues for sustaining Moore's Law–era advancements beyond traditional transistor scaling.

Another major federal initiative for addressing the challenges of semiconductor scaling and energy efficiency is the **DARPA Electronics Resurgence Initiative (ERI)**. Launched in 2017, ERI was designed to sustain US semiconductor leadership at a time when traditional silicon scaling approaches were reaching their limits. The initiative targeted several key areas, including new materials and architectures to overcome the physical constraints of silicon-based transistors, the development of advanced chip design tools to improve efficiency, and the co-design of hardware and software to optimize system performance. A core focus of ERI was reducing the energy consumption of microelectronics, which aligns with national goals for energy-efficient computing. By fostering collaboration between government, academia, and industry, the initiative sought to ensure a continuous pipeline of semiconductor innovation while addressing economic and security concerns.

In parallel, the **CHIPS and Science Act** of 2022 represents an effort to revitalize domestic semiconductor manufacturing and innovation. With over $52 billion allocated to semiconductor production and research, the act addresses vulnerabilities in the global supply chain while bolstering US leadership in chip manufacturing. It provides direct incentives for domestic chip fabrication, establishes the National Semiconductor Technology Center as a hub for advanced research and prototyping, and funds defense-related semiconductor development to ensure national security. Additionally, the act emphasizes workforce development and aims to expand semiconductor expertise through education and training programs. Research and development funding under the CHIPS Act focuses on next-generation semiconductor materials, process improvements, and emerging chip architectures to reinforce US competitiveness in the global technology landscape.

Complementing the CHIPS Act, the **Department of Defense (DoD) Microelectronics Commons Program** aims to accelerate the transition of semiconductor research into commercial and defense applications. This initiative creates regional innovation hubs dedicated to prototyping and scaling new microelectronics technologies, particularly in heterogeneous integration, advanced packaging, and novel computing architectures. By bridging the so-called *lab-to-fab* gap between research and production, the program seeks to strengthen the domestic semiconductor supply chain while reducing reliance on foreign manufacturing. It also plays a crucial role in ensuring that critical defense applications have access to secure, domestically produced microelectronics to preserve US technological and national security interests.

As computing continues to play an essential role in scientific discovery, national security, and economic growth, these coordinated efforts will be critical in ensuring that future computing systems balance performance and productivity across a range of energy scenarios. Without significant breakthroughs in energy-efficient computing, the growing energy demands of AI, HPC, and data-intensive applications could become a fundamental barrier to progress. Addressing this challenge requires a holistic approach that spans hardware, software, and systems-level innovation—an effort that is well underway through these national and global initiatives.

2 A View of the Computing Landscape

Over the past 20–30 years, the field of microelectronics has undergone a remarkable transformation, enabling unprecedented advances in computational power, device miniaturization, and cost efficiency [1]. This evolution was historically driven by Moore's Law, which predicted the doubling of transistor density on integrated circuits approximately every 2 years, and, more importantly, Dennard Scaling, which stated that as transistors shrink in size, their power density remains constant, thereby enabling increased clock speeds and transistor density without a proportional rise in power consumption. This exponential scaling fueled continuous improvements in performance and cost per computation and led to increasingly powerful computing, telecommunications, and consumer electronics devices. The profound impact of Moore's Law and Dennard Scaling has shaped nearly every aspect of modern society by accelerating innovations in scientific research, AI, edge computing, and data center infrastructure [1].

However, as the physical and economic limits of traditional semiconductor scaling become increasingly apparent, the trajectory of computing is shifting toward a new era—one defined not by raw transistor density but by energy efficiency, architectural specialization, and heterogeneous integration. The future of computing is no longer solely about achieving higher performance; it is about balancing computational capability with energy consumption while unlocking new computing paradigms that will define the next two decades of progress.

Early adopters of technologies like extremely heterogeneous systems-on-a-chip (e.g., mobile phones), specialized AI architectures and software (e.g., TPUs [tensor processing units]), and GPU-powered exascale supercomputers (e.g., DOE's Frontier) have demonstrated this transition is already well underway. Over the next two decades, this transition will accelerate and require a fundamental rethinking of computing at all levels of the stack—from materials and device physics to architectures, programming models, and algorithms. The move toward heterogeneous computing—leveraging a diverse array of specialized accelerators, memory technologies, and computing paradigms—will be essential to delivering energy-efficient performance. Emerging technologies such as analog and photonic computing, cryogenic and superconducting logic, bio-inspired solutions (e.g., neuromorphic and DNA computing), and quantum computing present many opportunities, but their integration into practical, large-scale computing infrastructures demands foundational research to scale them up and enable them to solve practical problems of mission importance.

To realize this future, substantial advancements will be needed across multiple dimensions. **New hardware devices and architectures** must continue the five-decade energy-efficiency trends of integrated circuits beyond conventional CMOS devices. **Algorithms** must be adapted to new hardware paradigms to exploit novel architectural features (e.g., analog computing, mixed-precision arithmetic) while minimizing energy costs. **Software ecosystems and programming systems** must evolve to productively support energy-efficient execution on heterogeneous platforms, thereby abstracting complexity while exposing opportunities for optimization. **Distributed data and workload-aware resource management** will play a crucial role in reducing energy-intensive data movement across large-scale systems. **Modeling, simulation, and co-design methodologies** are becoming increasingly critical for effectively designing, evaluating, and integrating emerging computing paradigms for specific workloads.

As early adopters have demonstrated, the potential gains are significant—not only in each individual research area but also in their synergistic effects, in which the combined impact of innovations across hardware, software, and algorithms can be multiplicative rather than additive. Achieving this vision requires a concerted research effort across the scientific community, industry, and government.

2.1 Industry Perspectives on Energy-Efficient Computing

The workshop opened with an industry panel moderated by Tanya Das (Bipartisan Policy Center), bringing together leaders from across the computing ecosystem: Tamar Eilam (IBM), Dan Ernst (NVIDIA), Chris George (Intel), Mark Helm (Micron), Andy Hock (Cerebras), Andrew Wheeler (HPE), and Thomas Zacharia (AMD). The discussion highlighted how DOE priorities and partnerships shape industrial innovation in energy-efficient computing, and how advances across hardware, software, and systems must be coordinated to maximize scientific impact per joule.

DOE's Role in Driving Innovation. Panelists consistently noted that DOE's ambitious system-level goals, from Titan through Summit and Frontier, have catalyzed industry progress in energy efficiency. By setting aggressive performance-per-watt targets and funding high-risk, high-reward investments, DOE has pushed companies to co-design architectures with the scientific community. Speakers emphasized that DOE's investments in software—through programs like SciDAC and the Exascale Computing Project—were as critical as hardware procurements, incentivizing researchers to adapt codes to new architectures. Several participants stressed that continued DOE leadership is essential, especially as commercial hyperscalers deploy computing infrastructure at unprecedented scale. While DOE procurement analysis demonstrates that DOE computing centers remain less expensive than commercial cloud providers, the sheer magnitude of private sector investment, where the five largest hyperscalers collectively spending over $400 billion annually [16–18], threatens to shift the center of gravity for high-end computing capabilities away from federally-supported research infrastructure and requirements.

Architectural and Device-Level Advances. Industry representatives described a range of strategies to reduce energy consumption. IBM underscored the dual relationship between computing and technology lifecycle, calling for heterogeneous hardware—including accelerators, neuromorphic designs such as TrueNorth, and quantum devices—while also considering lifecycle emissions from semiconductor manufacturing. Intel emphasized the importance of new materials, interconnect technologies, and chiplet packaging, while warning that software ecosystems must evolve to keep pace. Micron noted that energy efficiency has become a first-order design criterion alongside cost-per-bit, with high-bandwidth memory and processing-in-memory approaches offering major opportunities. Cerebras showcased wafer-scale integration as an example of designing from first principles to reduce data movement and achieve orders-of-magnitude improvements in throughput and efficiency.

Systems and Data-Center Perspectives. Participants highlighted that energy efficiency must be addressed at the full-system level, dynamically, not just at the system design phase. DOE has supported R&D to allow data centers to rapidly respond to variability in power cost and availability, as well as advanced cooling technologies. These topics were identified as an emerging frontier, particularly as AI workloads drive unprecedented power densities. Panelists emphasized the need for co-design at scale, aligning chip, rack, system, and data center innovations to integrate renewables and balance grid demand dynamically. HPE argued that DOE procurement requirements should explicitly prioritize energy efficiency, since market incentives alone do not yet compel vendors or users to pay a premium for efficiency.

Software, Algorithms, and Co-Design. Throughout the discussion, panelists agreed that the most impactful energy savings may come not from hardware, but from methods and algorithms. DOE can accelerate progress by supporting new algorithmic approaches—such as mixed-precision and AI surrogates—that reduce unnecessary computation. Several participants noted that DOE mini-apps and co-design projects have historically been effective in aligning hardware and software development, and new exemplars are needed for AI-driven science workloads. Panelists stressed that the "most energy-efficient FLOP is the one never executed," underscoring the importance of algorithmic innovation alongside device advances.

Partnerships, Standards, and Workforce. Speakers called for stronger public–private partnerships and cross-disciplinary R&D efforts that span materials science, device physics, software engineering, and domain science. Industry-wide standards for measuring and reporting energy use were seen as critical to accelerate adoption of best practices. Workforce development was also highlighted as a barrier: attracting and training scientists and engineers requires access to leadership systems and compelling DOE mission problems. As one participant observed, DOE systems serve as the "rockets of the 21st century," inspiring the next generation while tackling humanity's most pressing scientific challenges.

Key Takeaways. We summarize the key takeaways from the industry panel as follows.

- DOE leadership in setting ambitious energy goals has been a critical driver of industrial innovation.
- Integration of software and hardware, not just advances in devices, will define the next phase of progress.
- R&D in data center demand response, advanced cooling, and heterogeneous integration are near-term opportunities.

- Renewed focus on algorithms and multidisciplinary, codesign approaches can unlock the greatest energy savings.
- DOE's role as convener, risk-taker, and steward of mission-driven applications remains essential for guiding industry toward solutions that maximize scientific discovery per joule.

2.2 Hardware

As highlighted by Peter Kogge in the workshop's keynote presentation, the next 10–15 years will most likely witness the emergence of an even more diverse range of computing technologies and paradigms that are driven primarily by the pursuit of energy efficiency and higher performance.

2.2.1 Historical Trends in Computing Hardware

For decades, Dennard scaling [5, 6] played a central role in improving energy efficiency in computing hardware. It established that, as transistors shrink, their power density remains constant, allowing engineers to increase transistor density and clock speeds without a proportional rise in power consumption. This principle enabled continuous performance scaling while keeping power constraints manageable. However, by the mid-2000s, physical limitations such as current leakage, thermal dissipation, and increasing power density disrupted this trend and led to a breakdown of Dennard scaling. As voltage scaling became impractical, power consumption rose sharply with each successive technology node, thereby making energy efficiency a critical bottleneck in HPC system design.

By the 2010s, the slowdown of Moore's Law further exacerbated this challenge and prompted a shift toward heterogeneous computing. Instead of relying solely on general-purpose CPUs, the industry increasingly turned to specialized accelerators (e.g., GPUs [19] and TPUs [20]) to deliver higher performance for specific workloads, including computational science, AI, data analytics, and graphics processing. While these innovations provided significant computational gains, they also underscored a new reality: energy efficiency was now the dominant constraint for system scalability.

This shift has been particularly pronounced in data centers, where energy consumption has grown substantially. Today, data centers are estimated to account for between 1% and 2% of global electricity consumption [3], a figure that is expected to rise sharply without transformative improvements in energy-efficient computing [21]. This challenge extends beyond large-scale computing facilities with the growing prevalence of edge computing, AI-driven applications, and mobile devices, all amplifying the need for energy-aware design strategies across the entire computing landscape.

Accordingly, the computing community stands at a pivotal moment [22, 23]. The relentless demand for computational power—driven by AI, HPC, and emerging edge applications—is reaching levels that cannot be sustained through transistor miniaturization alone. Energy efficiency has become the defining constraint and requires a paradigm shift in computing architecture. The traditional approach of scaling transistor density is being replaced by a holistic, multifaceted strategy that integrates novel computing paradigms to ensure continued performance increases for the decades ahead.

In the near term, we expect to see domain-specific architectures that use conventional CMOS. In the longer term, we expect CMOS to persist as an important component of all computing systems, but it will be augmented by non-CMOS accelerators (e.g., analog crossbars that calculate matrix-vector products).

2.2.2 Domain-Specific Architectures in the 'Golden Age of Computer Architectures'

In response to the growing energy constraints in computing, the field is shifting toward architectures that offer dramatic improvements in energy efficiency [22, 24]. In the near term, heterogeneous integration, chiplet-based architectures, and waferscale integration have emerged as strategies to meet the increasing demands of AI, scientific simulations, and real-time data processing while maintaining acceptable levels of power consumption.

Although chiplet-based designs were initially introduced to improve yields in semiconductor manufacturing, their adoption has accelerated with the need for specialized, domain-specific computing [25, 26]. Traditionally, monolithic chips integrated all components onto a single die, thereby limiting flexibility in optimization and increasing manufacturing complexity as transistor scaling reached fundamental limits. Heterogeneous integration offers a more modular approach by combining diverse processing elements—CPUs, GPUs, TPUs, memory, and other accelerators—within a single package to allow each component to be optimized for a specific computational task and possibly use an optimal manufacturing process for each chiplet. This approach enhances both performance and energy efficiency.

Chiplet architectures enable domain-specific compute capabilities while reducing power consumption. Unlike traditional designs, in which performance improvements were achieved primarily through frequency scaling and transistor miniaturization, chiplet-based architectures leverage task-specific accelerators to more efficiently execute workloads. For example, GPUs optimize parallel processing for graphics and AI inference, TPUs accelerate deep learning workloads, and encryption and compression processing cores are commonplace. One of the most energy-efficient platforms, the ubiquitous smartphone, was designed to connect many specialized functional units on a monolithic die or package—a precursor to chiplets—to provide workload specialization for dramatic energy efficiency improvements. Once designed, these specialized functional units were easily reusable. The next generation of computing systems will use chiplet-based architectures and heterogeneous integration as they continue to evolve to meet the demands of high-performance and energy-efficient computing.

Along another dimension of customized integration, **waferscale computing** assembles entire computational systems on a single silicon wafer, thereby reducing chip-to-chip communication overhead and significantly improving energy efficiency. Unlike traditional architectures, in which wafers are diced into smaller chips, waferscale designs provide a contiguous, high-bandwidth processing substrate that is ideal for massively parallel tasks such as deep learning and AI training. Current system architectures are primarily homogenous funcitional units replicated on the same wafer; however, wafers populated with heterogeneous functional units are also an option. Cerebras has pioneered waferscale computing with processors that feature billions of transistors across a single monolithic wafer to enable highly efficient execution of AI and scientific workloads [27]. Recent demonstrations have shown advantages even beyond AI, with applications in seismic processing and numerical simulations [28].

2.2.3 Beyond CMOS: Hardware Innovation over the Next Two Decades

Looking beyond current CMOS-based architectures, a range of emerging computational paradigms could deliver orders-of-magnitude improvements in energy efficiency. Innovations such as low-power analog and photonic processors, cryogenic logic, and bio-inspired architectures offer promising avenues for reducing energy consumption if they can be manufactured economically at scale to address mission challenges. Many of today's AI dataflow accelerators, near- and in-memory computing technologies, and waferscale integrated system have already demonstrated impact. These advancements indicate a growing potential to reshape computation by shifting energy-intensive tasks to architectures specifically designed for efficiency.

Opportunities in hardware innovation will continue to expand, with cryogenic and superconducting logic unlocking pathways to ultra-low-power, high-speed computation, analog systems advancing brain-like efficiency, and native probabilistic hardware enhancing randomized algorithms and uncertainty quantification (UQ). Looking further ahead, entirely new paradigms such as quantum computing and DNA computing promise to fundamentally alter how certain classes of problems are solved. However, rather than serving as a universal replacement for conventional architectures, these technologies are expected to function as specialized components within a broader heterogeneous computing ecosystem, in which each platform is optimized for specific applications.

To set the stage for this evolving landscape, we provide a brief overview of selected candidate technologies discussed at our workshop; we highlight key research directions, ongoing developments, and the challenges that must be addressed for large-scale adoption. A comprehensive survey of all technologies is outside the scope of this workshop; we refer readers to recent work [24, 25]. A broader discussion on quantum computing can be found in the 2023 Basic Research Needs in Quantum Computing Report [29].

Cryogenic and Superconducting Computing. Cryogenic Computing and Superconducting Logic represent a transformative approach to high-efficiency computation, leveraging operating temperatures much lower than room-termperature and, in some designs, near absolute zero (ultra-low temperatures), to dramatically reduce energy con-

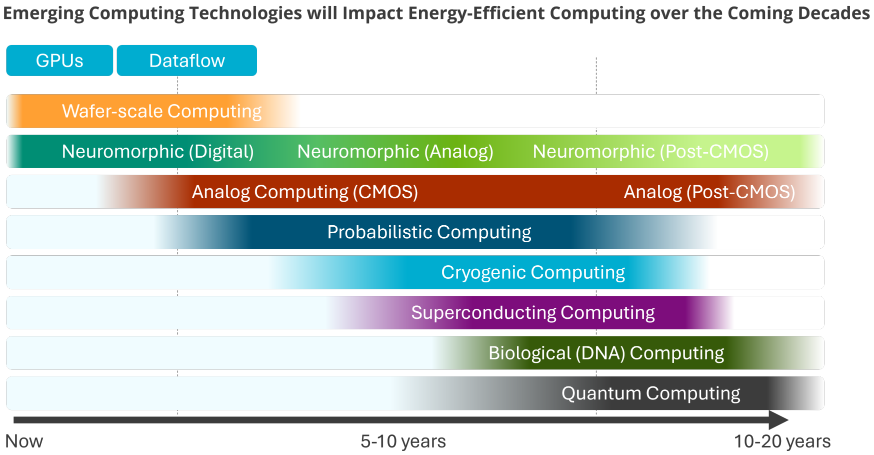

Figure 1: Notional projected timeline of when emerging computing architecture and beyond CMOS technologies will impact science computing applications. All of these technologies have the potential for rapid technological advancement that could accelerarate their arrival.

sumption. Cryogenic Computing, in its general sense, minimizes thermal noise and significantly reduces resistive losses in electronic circuits, enhancing transistor and memory performance for applications [30, 31]. The most prominent realization of this field is Superconducting Computing, which is a specific type of cryogenic computing that utilizes materials exhibiting zero electrical resistance under these ultra-low-temperature conditions. This feature enables lossless current flow and allows the development of logic gates, memory, and communication pathways with unprecedented energy efficiency for both classical AI and HPC applications [31, 32]. A key driver for interest in this technology is its compatibility with quantum computing. Since cryogenic environments are necessary for qubit stability, the development of high-efficiency, large-scale cryogenic computing platforms creates a powerful potential synergy between classical superconducting logic and quantum processing. Research initiatives are actively investigating superconducting logic circuits and cryogenic memory to enable this ultra-efficient data storage and processing capability. Despite advances in closed-cycle cryogenic refrigeration that have made the technology increasingly practical, several challenges related to scalability and integration remain. Current research efforts are focused on addressing cryogenic cooling constraints, scaling superconducting interconnects, ensuring compatibility with standard semiconductor manufacturing processes, and mitigating the challenges of limited I/O and commensurate memory access and capacity. Future research directions include developing materials optimized for cryogenic operation and designing hybrid cryogenic-electronic computing architectures.

Analog Computing. Analog computing processes data in continuous waveforms rather than discrete binary values, thereby eliminating energy-intensive digital-to-analog conversions. This approach is particularly advantageous for AI, signal processing, and certain scientific simulations in which calculations can be performed more efficiently at lower precisions. Analog processors have demonstrated the ability to execute fundamental matrix multiplications—a core AI operation—with significantly lower energy consumption than digital architectures [33].

The energy efficiency of analog computing arises from its ability to directly manipulate physical quantities such as voltage and current rather than translating them into binary code. However, key challenges—including noise susceptibility, limited precision, and complex error correction—must be addressed for widespread adoption. Future research will focus on refining low-power analog circuits and integrating analog computing into hybrid digital-analog architectures, particularly for AI workloads, real-time signal processing, and ultra-low-power embedded applications.

(N.B. DOE ASCR held a Analog Computing for Science Workshop, which was colocated with this workshop.)

Neuromorphic Computing. Neuromorphic computing looks to leverage inspiration from the brain's unique architecture and components (i.e., neurons and synapses) to offer a time- and energy-efficient alternative to stored-program architectures like Von Neumann systems. This area was first explored as early as John Von Neumann [34] and advanced by Carver Mead [35]. More recently, a number of digital and hybrid digital–analog neuromorphic platforms have reached near brain-like scales [36], and neuromorphic computing could have application impact in domains ranging from AI to scientific computing.

While digital neuromorphic systems are reaching sizes suitable for scalable algorithm exploration and integration with conventional technologies, extensive work continues toward analog neuromorphic solutions. The dynamics of neurons, synapses (the connections between neurons), and learning all appear suitable for analog hardware approaches, which could offer considerable energy and space savings. Furthermore, the distinct dynamics of biological neural systems are seen as inspirations for a wide range of emerging materials and novel device research, suggesting neuromorphic computing could be an avenue for non-silicon, post-Moore devices.

(N.B. DOE ASCR held a Neuromorphic Computing for Science Workshop, which was colocated with this workshop.)

Photonic Computing. Photonic computing [37, 38] utilizes light instead of electrons for computation and communication, an approach that leverages the inherent advantages of photons for high-speed, low-energy processing. Unlike electronic circuits, photonic systems generate minimal resistive losses and heat, making them particularly promising for AI, HPC, and telecommunications [39].

A major breakthrough in this field is silicon photonics, which integrates optical components into traditional silicon-based systems. This hybrid approach enables energy-efficient, high-bandwidth data transfer, thereby reducing the power overhead associated with electrical interconnects, which is one of the largest energy bottlenecks in modern computing. Researchers are actively developing optical neural networks and fully photonic data centers, which could dramatically lower power consumption and improve processing speeds.

Key challenges include improving photonic interconnects, developing compact and low-power lasers, and refining materials such as silicon and indium phosphide for optimal photon transmission. Overcoming these hurdles could enable energy-efficient photonic computing systems capable of handling demanding workloads.

Probabilistic Computing. Probabilistic Computing [40, 41] is an emerging computational paradigm that intentionally incorporates controlled stochasticity (randomness) into the processing pipeline. This methodology optimizes workloads where the algorithms can inherently tolerate approximate or non-deterministic results, including large-scale AI inference, statistical modeling, and various randomized algorithms. Initial probabilistic CMOS and noise-tolerant architectures have shown significant reductions in power consumption for AI and real-time data analysis.

Future research must focus on error mitigation strategies, hybrid probabilistic-digital architectures, and adaptive computation models that can dynamically adjust precision to optimize energy efficiency. This paradigm holds particular promise for power-constrained edge devices and IoT systems.

DNA Storage and Computing. As data generation continues to grow exponentially, DNA storage and computing offer a molecular approach to storing and processing information with unparalleled energy efficiency and data density. DNA's theoretical storage density of approximately 215 PB/g far exceeds that of conventional storage media, while its stability allows for long-term archival storage with minimal energy requirements [42–45].

DNA computing, which leverages biochemical interactions to perform computations, has shown promise in combinatorial optimization and cryptographic applications [46]. However, challenges remain in scaling DNA synthesis and sequencing, improving error correction, and integrating DNA-based processing with traditional computing architectures. Continued research could lead to hybrid DNA-silicon systems that enhance data storage and specialized computational workloads.

Future research must address heat dissipation challenges, improve fabrication techniques (including emergent packaging strategies and 3D and heterogeneous integration), and explore broader scientific applications. If successful, waferscale architectures could become a foundational technology for energy-efficient HPC.

Hardware Summary. As energy efficiency becomes the defining constraint in computing, the next two decades will require radical departures from traditional architectures. While CMOS-based domain-specific accelerators will continue to dominate in the near term, emerging technologies such as cryogenic, neuromorphic, waferscale, analog, photonic, superconducting, probabilistic, and DNA computing offer compelling long-term solutions. Rather than replacing existing platforms outright, these innovations will play complementary roles in a heterogeneous computing landscape in which workloads are matched to the most energy-efficient hardware available. Sustained research across materials science, device physics, architectures, software, and algorithms will be essential to realizing this vision and ensuring that computing remains scalable, viable, and scientifically impactful well into the future.

2.3 Software

Although hardware innovations are essential for improving energy efficiency, software plays an equally critical role in ensuring that these advancements translate into real-world energy savings [10]. As computing systems become increasingly complex and heterogeneous, the software stack—from algorithms to operating systems to programming models—must be reimagined to actively minimize power consumption [12]. Traditionally, software development has prioritized performance, flexibility, and portability, often with little consideration for energy efficiency. However, as energy consumption emerges as a defining constraint in computing, software must be designed not just for speed but also for energy-aware execution across diverse workloads and architectures.

A key challenge lies in bridging the gap between the energy-efficient hardware and the software that can leverage it effectively. Without explicit energy-aware optimizations, even *the most advanced low-power hardware can operate inefficiently due to poor data locality, excessive memory accesses, or inefficient scheduling*. To address this, the software ecosystem must evolve at multiple levels—from algorithms and compilers to operating systems and programming models—by integrating energy efficiency as a first-class design objective.

2.3.1 Programming Languages, Compilers, and Developer Tools

To support energy-efficient software development, the compiler tool chain and programming models must evolve to expose energy-aware constructs and enable power-efficient execution by default. Modern compilers can analyze energy consumption at the source code level by applying energy-efficient transformations during optimization passes. These include cache-friendly data layouts, loop transformations to improve locality, and instruction selection strategies that minimize power-intensive operations.

Several emerging programming languages and libraries integrate energy-aware tools to provide developers with better control over power consumption. Languages such as Rust and Julia, known for their high-performance optimizations, are also being explored for their utility in energy-efficient computing. Domain-specific libraries for energy-efficient computing provide metrics that evaluate the power footprint of various computations to allow developers to make informed trade-offs between accuracy, performance, and energy usage.

Additionally, hardware-aware compiler frameworks (e.g., MLIR and LLVM) enable cross-layer energy optimizations by leveraging heterogeneous hardware more effectively. These compilers can automatically generate energy-efficient code by selecting appropriate vectorized instructions, accelerator offloading strategies, and memory layouts to minimize power consumption. Future research must continue to advance auto-tuning compilers that dynamically optimize software energy efficiency, thereby adapting code to the run-time conditions and available hardware.

2.3.2 Runtime Systems and Power Management

One major shift in software design has been the development of energy-aware runtime systems that dynamically manage power consumption based on workload characteristics. These runtime environments adjust resource allocation, processor voltage, and frequency scaling in real time to ensure that computational resources are used efficiently. Dynamic power management techniques, such as adaptive power scaling and intelligent core scheduling, allow software to reduce power consumption without sacrificing performance.

For example, modern task schedulers in cloud computing and HPC environments can migrate workloads across low-power cores or specialized accelerators to dynamically optimize energy consumption. Similarly, data center management software is increasingly incorporating energy-aware load balancing by shutting down idle servers or consolidating workloads to reduce energy waste. Such strategies enable energy-efficient execution without requiring direct modifications to applications.

Beyond large-scale computing, mobile and edge computing platforms rely on software-driven energy optimizations to extend battery life and manage power constraints efficiently. Lightweight AI inference models, adaptive streaming algorithms, and real-time power-aware workload distribution are actively shaping next-generation mobile and embedded software.

2.3.3 Software Development and Profiling Tools

To foster widespread adoption of energy-efficient programming practices, developers must have access to real-time energy profiling tools that provide actionable insights into software power consumption. Energy profiling frameworks (e.g., Intel's Power Gadget, NVIDIA's NVML, and ARM's Streamline) allow developers to visualize and analyze power consumption across different workloads.

Advanced profiling tools can be included in integrated development environments and continuous integration/continuous deployment (CI/CD) pipelines to enable developers to identify and optimize energy hotspots early in the development cycle. Emerging AI-driven optimization tools automatically suggest energy-efficient code transformations by applying power-aware best practices without requiring extensive manual intervention.

The future of energy-efficient software lies in seamless automation—in which software optimizes itself based on power constraints, hardware availability, and workload characteristics. By integrating energy awareness into every stage of the software development process, from algorithm design to deployment, computing can achieve orders-of-magnitude improvements in power efficiency without sacrificing usability or performance.

2.4 Algorithms

At the algorithmic level, researchers are developing energy-optimized algorithms that minimize data movement and computational intensity, both of which are among the most significant contributors to energy consumption. Unlike traditional complexity models that focus on time and space efficiency, energy-aware algorithms must consider power usage, memory access patterns, and data locality as first-class optimization criteria.

In AI, techniques such as model quantization, pruning, and sparsity have proven effective in reducing computational complexity while preserving model accuracy. By lowering numerical precision (e.g., using 8-bit integers instead of 32-bit floating point numbers) or eliminating redundant computations, these methods significantly reduce the energy required for inference and training. In deep learning, sparse tensor representations and structured pruning help minimize unnecessary operations, leading to orders-of-magnitude improvements in power efficiency. Additionally, approximate computing algorithms, which trade off precision for energy savings, are being explored in domains where exact solutions are unnecessary (e.g., image processing, speech recognition, and probabilistic modeling).

Beyond AI, fundamental research holds the promise of new, energy-efficient algorithms and data structures that align with modern hardware constraints. Traditional sorting and searching algorithms, for example, often incur significant memory access overhead, which is a major contributor to power consumption in large-scale computing systems. Optimizing these algorithms for cache efficiency, data locality, and reduced memory bandwidth (e.g., communication-avoiding algorithms [47]) can yield significant energy savings. In database systems and HPC, energy-aware indexing methods, compressed data structures, and adaptive caching policies have been proposed to minimize redundant data movements and unnecessary computations.

Some of the most promising areas of energy-efficient computing are graph processing and sparse data structures, which are particularly relevant for HPC, AI, and distributed computing. Graph-based neural networks, federated learning models, and edge AI frameworks increasingly incorporate energy-optimized matrix operations and sparse linear algebra techniques to reduce redundant computation and minimize communication overhead. In large-scale graph analytics, algorithms that prioritize locality-aware traversal and distributed workload balancing can dramatically reduce the energy costs associated with network congestion and excessive memory accesses.

Additionally, energy-aware scheduling and adaptive algorithm selection are emerging as key strategies in optimizing software execution. Dynamic algorithm selection frameworks allow systems to choose energy-efficient variants of algorithms at run time to adapt the execution to the hardware characteristics and workload constraints. These approaches are particularly useful in heterogeneous computing environments, where different processing units (e.g., CPUs, GPUs, FPGAs, and TPUs) have vastly different energy-performance trade-offs.

Moving forward, energy efficiency must become a first-class consideration in algorithm design by spanning numerical methods, data analytics, and architectural optimization. Research must continue to integrate energy complexity metrics into algorithmic analysis to ensure that future computational advances do not come at an unrealistic power cost. By adopting a multidisciplinary approach that bridges theoretical algorithm design with hardware-aware execution models, the community can move toward substantial energy savings while maintaining computational accuracy and performance.

2.5 Co-Design

Achieving maximal energy efficiency in computing requires a **co-design approach** in which hardware, software, and algorithms are developed in tandem to ensure that software is optimized for the underlying architecture [13]. Traditional software development often assumes general-purpose execution environments, but as computing systems become more heterogeneous, tailoring algorithms to specific hardware characteristics is increasingly essential [48]. Co-design strategies allow software and hardware to evolve together, thereby leveraging the strengths of specialized architectures to minimize data movement, improve parallelism, and reduce overall energy consumption.

For example, algorithms designed for in-memory computing architectures can execute certain operations directly within memory, thereby dramatically reducing energy-intensive data transfers. This is particularly beneficial for data-intensive workloads (e.g., graph processing, AI inference, and large-scale numerical simulations), in which moving data between processors and memory dominates power consumption. In AI and deep learning, co-design has led to innovations such as systolic arrays in TPUs [20,49–51] and sparsity-aware matrix multiplications, which reduce computational redundancy and lower power consumption. In scientific computing, co-designed architectures for stencil computations, finite element methods, and molecular dynamics simulations [52] optimize memory access patterns and minimize energy overhead.

Beyond gains in energy efficiency and performance, co-design also enhances programmability and usability. High-level domain-specific languages (DSLs) [53] and compiler tool chains tailored for specialized architectures enable developers to write energy-efficient code without manually tuning low-level hardware interactions. Machine-learning-driven autotuning frameworks can dynamically select optimal execution strategies to adapt software to heterogeneous and reconfigurable hardware platforms for real-time energy optimizations.

As energy efficiency becomes a fundamental constraint in computing, the importance of co-design will only grow. Future research must explore cross-layer integration—bridging algorithms, software, and hardware to develop next-generation computing paradigms in which energy-aware execution is seamless, scalable, and ubiquitous.

3 Workshop Findings

This section describes the discussions and findings of the workshop, which was organized into eight sets of breakout groups. The first seven breakout groups were largely focused on functional areas:

- 3.1 Algorithms
- 3.2 Hardware
- 3.3 Data storage and management
- 3.4 Modeling and simulation
- 3.5 Facility to edge
- 3.6 Resource management
- 3.7 Programming systems

The last breakout groups were chosen spontaneously based on crosscutting topics that emerged from earlier discussions:

- 3.8.1 Test beds and prototypes
- 3.8.2 Metrics
- 3.8.3 Benchmarking
- 3.8.4 Hardware-software integration
- 3.8.5 Algorithm-hardware (paradigm) integration

Findings and research opportunities covered in the following sections are those discussed by the participants in the respective breakout session. In some cases similar topics came up and were discussed in multiple breakouts, providing additional perspectives.

3.1 Algorithms

Historically, minimizing execution time has been a reasonable heuristic for minimizing energy consumption [54]. However, this assumption is increasingly challenged by the high energy costs associated with data movement and the increasing architectural capability to move computation closer to memory and storage. Time-saving strategies that also reduce data movement, such as randomized algorithms, quantized and sparse models [55, 56], and reduced-/mixed-/adaptive-precision computation [57], remain effective and can provide synergistic benefits while maintaining computational fidelity [58–62]. Yet, other time-centric optimizations may yield diminishing energy returns, highlighting the need for algorithmic approaches that explicitly leverage energy-efficient hardware features. Another emerging opportunity is in neural surrogate models, such as physics-informed neural networks [63], where different neural network formulations may offer similar performance but present potentially useful trade-offs in how they map onto energy-efficient hardware features and accelerators.

Beyond optimizing algorithms for existing architectures, new computing paradigms—such as analog, reversible, and probabilistic computing—require a fundamental rethinking of algorithms to fully exploit their potential efficiency gains. For example, event-driven neuromorphic processors derive much of their energy efficiency from communicating in spikes rather than dense numerical representations [36], but shifting from high-precision to distributed low-precision algorithms remains an open research challenge [64]. Notably, the relative immaturity of these emerging computing platforms presents an opportunity for algorithm designers to influence their development and ensure that the hardware innovations align with the needs of scientific computing and energy efficiency.

Critically, energy efficiency is often perceived as a trade-off against solution accuracy, but this need not always be the case. Many iterative algorithms, including those used for solving nonlinear systems and optimization problems, provide a framework for balancing accuracy requirements across subcomponents while still converging to high-precision solutions [65–67]. In other cases, iterative refinement techniques can rapidly improve an initial low-precision solution into a high-accuracy result [68–72]. If energy constraints are a key design factor, then future research should consider algorithms that maximize accuracy within a fixed energy budget rather than simply minimizing energy at a given accuracy threshold.

3.1.1 Findings and Discussion

For decades, the rapid advancements driven by Moore's Law and Dennard Scaling meant that energy efficiency was rarely a first-class constraint in algorithm design. Moreover, in general-purpose von Neumann computing environments, execution time served as a reasonable proxy for energy consumption. However, as hardware energy efficiency gains have slowed, this assumption is no longer sufficient.

Although progress has been made in algorithms that minimize data movement (e.g., incorporating Big-O complexity models for data movement), energy consumption has largely been treated as a secondary consideration. Workshop participants strongly agreed that this must change for true energy-efficient computing to be realized. However, achieving this shift requires a cultural change in how algorithms are designed and evaluated as well as the development of new theoretical formalisms and metrics that explicitly capture energy consumption.

A significant challenge arises from the increasing complexity of algorithmic analysis in heterogeneous and emerging computing paradigms. Many of the most promising energy-efficient hardware approaches (e.g., AI surrogate models, randomized algorithms, and low-precision arithmetic) introduce additional variability in solution accuracy. This is particularly relevant when considering probabilistic or approximate algorithms (e.g., Monte Carlo methods, randomized numerical linear algebra, and stochastic differential equation solvers), which trade deterministic precision for energy savings.

Participants also expressed concern that, although energy efficiency is increasingly prioritized by industry, research efforts are heavily focused on AI applications, particularly in edge computing and data centers. However, the requirements of scientific computing are fundamentally different and necessitate co-design between the scientific algorithms community and hardware developers across all technology readiness levels. The ability of this co-design to influence broader hardware development is inversely related to the maturity of the technology; i.e., it is likely easier for the scientific computing community to influence the trajectory of an emerging technology *before* widespread application impact than to after it has an established market. Without this co-design, the scientific community risks being forced to adapt to hardware architectures designed for commercial applications, thereby limiting its ability to achieve transformative energy efficiency gains.

Despite these concerns, workshop participants expressed optimism that emerging computing paradigms could introduce fundamentally new algorithmic opportunities. For instance, probabilistic hardware capable of directly sampling complex distributions is seen as a potential game-changer for randomized algorithms [41]. Likewise, the potential of neuromorphic hardware for solving stochastic differential equations (SDEs) suggests new directions in scientific computing, potentially shifting problem formulations from partial differential equations to SDEs, which can naturally embed UQ [73].

Overall, the increasing heterogeneity of computing platforms underscores the need for concrete theoretical foundations for energy-aware algorithm design. Outside of conventional von Neumann and quantum computing models, there is no widely accepted Big-O complexity formalism for energy, although there have been recent attempts at developing formalisms for physical and neuromorphic computing [74, 75]. Workshop discussions emphasized the importance of developing energy-, time-, and precision-aware complexity models for analog, neuromorphic, probabilistic, and dataflow-based architectures.

3.1.2 Research Opportunities

Workshop participants roughly partitioned the research opportunities into three categories: Energy-Aware Scientific Campaigns, Design and Characterization of Energy-Efficient Algorithms, and From Co-Design to Cooperation Between Algorithms and Energy-Efficient Hardware.

Energy-Aware Scientific Campaigns. Rather than focusing exclusively on per-algorithm optimizations, research should explore strategies that reduce energy consumption across entire scientific workflows. In many cases, using a more energy-intensive algorithm early in a computation may reduce overall energy consumption downstream by enabling more efficient data processing, sensor usage, or computational steering. This holistic view of energy efficiency—in which the energy cost of an entire scientific campaign is optimized—represents a new paradigm that is distinct from traditional local optimization approaches. Future research should develop

- **algorithms that reduce energy consumption upstream or downstream of their execution**, thereby optimizing across full computational pipelines, and
- **metrics for measuring energy efficiency at the campaign level** to enable better decision-making in large-scale scientific computing environments.

Design and Characterization of Energy-Efficient Algorithms. A major research priority is the development of theoretical models for energy efficiency akin to the Big-O complexity theory for time and space. By formalizing energy complexity models, researchers can influence the design of emerging architectures while also creating empirical methods for characterizing energy-efficient algorithms. Current theoretical models largely focus on operations per second (flops) or data movement, neither of which directly measure energy consumption. Future work should aim to

- develop **energy-aware complexity metrics** that extend beyond conventional work-based analysis,
- design **new algorithmic strategies** for leveraging probabilistic hardware, analog computation, and mixed-precision environments [63], and
- establish **empirical frameworks for measuring algorithm energy consumption** in realistic execution environments.

From Co-Design to Cooperation Between Algorithms and Energy-Efficient Hardware. Hardware-algorithm co-design must extend beyond initial system design to enable continuous cooperation throughout execution. This involves situational awareness, in which algorithms can adapt to changing hardware conditions, and standardized communication mechanisms that allow software to interact dynamically with energy-aware hardware features. Today, even in existing co-design efforts, collaboration typically ends once hardware is finalized. Moving forward, research efforts should

- develop **standardized APIs for real-time algorithm-hardware interactions** to accelerate feedback between software and hardware optimizations,
- design **adaptive algorithms** that can adjust precision and computational effort in response to dynamic energy constraints, and
- establish **hardware-algorithm test beds** to accelerate real-world energy-efficient algorithm research [64,76].

3.1.3 Summary

Algorithms are at the heart of DOE scientific workflows. By developing energy-efficient algorithms, algorithms for energy-efficient hardware, theory and metrics for characterizing algorithms as part of a scientific workflow, and mechanisms for principled hardware-software co-design and cooperation, we can ensure that future DOE computational workflows maximize scientific output within given energy constraints.

3.2 Hardware

The development of energy-efficient computing hardware spans multiple technological levels, from fundamental device components (e.g., transistors and memory cells) to system-level architectures, interconnects, and software-hardware co-design frameworks. Each of these levels presents distinct opportunities for reducing power consumption with different implementation timelines and scalability considerations. Some solutions provide incremental energy savings while requiring minimal disruption to existing systems, whereas others introduce novel computing paradigms that necessitate cross-layer innovations in software, architectures, and applications.

A balanced strategy is therefore necessary for pursuing high-risk, high-reward innovations that promise dramatic long-term efficiency gains while also leveraging near-term improvements that align with existing semiconductor and computing roadmaps. This dual approach ensures that the scientific community remains at the forefront of hardware advancements while allowing software, system, and application developers to gradually adapt to evolving architectures.

For DOE hardware research, two parallel strategic imperatives emerge:

- **Co-design partnerships with industry:** Establish hardware-software co-design partnerships with semiconductor manufacturers and system integrators to ensure that DOE workloads (e.g., scientific simulation, AI for physics, large-scale data analytics) drive architectural decisions in commercial hardware development, rather than being retrofitted to commodity designs, and
- **Early-stage investment in alternative computing paradigms:** Fund prototype development and application porting for emerging architectures (e.g., analog computing, photonic processors, neuromorphic systems) that demonstrate 10x–100x energy efficiency improvements, with explicit requirements for scalable software abstractions and vendor-independent programming models to enable research community adoption.

3.2.1 Findings and Discussion

Discussions in the hardware breakout groups focused on several key themes: Alternative Device Technologies, Architectural Specialization and Heterogeneous Computing, and Electronic Design Automation Tools and Prototyping. These findings align with insights from other DOE and community workshops on specialized computing technologies [13, 77].

Alternative Device Technologies. The workshop discussions encompassed a broad spectrum of alternative computing technologies beyond conventional CMOS approaches, including cryogenic and superconducting computing (CSC), photonic computing, DNA-based storage, and emerging memory technologies. These novel paradigms could significantly impact energy efficiency but require further research, co-design with algorithms, and scalable manufacturing processes.

Cryogenic and Superconducting Computing: CSC refers to computing architectures that operate at cryogenic temperatures (e.g., 4–77 K) to leverage superconducting materials or low-resistance cryogenic CMOS [78, 79]. An important energy advantage of CSC stems from near-zero wire resistance, which significantly reduces interconnect power dissipation. These architectures have natural synergies with scientific applications that already rely on cryogenic sensors, superconducting magnets, and quantum computing control circuits [80, 81]. However, significant challenges remain regarding scalability and integration. While cryogenic CMOS faces power dissipation limits, superconducting logic based on Josephson junctions is constrained by low device density. On-chip memory for JJ-based families remains a bottleneck, requiring integration with cryogenic CMOS memory solutions at slightly higher temperatures (e.g., 65–77 K). Additionally, efficient cable technologies are required for low-power data transfer between cryogenic and room-temperature environments. Finally, current EDA tools are not optimized for cryogenic circuit synthesis, and very few facilities can manufacture cryogenic chips at scale. Addressing these challenges requires validated simulation frameworks, EDA tool development, and infrastructure investments to support manufacturing and testing.

Photonic Computing: Photonic technologies offer high-bandwidth, low-energy interconnects, which makes them a compelling alternative to electrical data movement in HPC systems [82]. Additionally, photonic computing accelerators have demonstrated efficiency in matrix multiplication and AI workloads [83]. However, effective system integration is essential, and photonic elements must be co-designed with electronic circuits rather than treated as drop-in replacements [84].

Emerging Compute and Memory Devices: New memory and storage technologies—such as ferroelectronic, spintronic, magnetic, and DNA-based storage—promise higher density and energy efficiency [85–87]. However, scaling challenges, manufacturing readiness, and long-term reliability remain barriers to adoption [88–90]. Research must focus on the characterization and benchmarking of emerging devices for scientific applications, the development of physics-based compact models to facilitate device-architecture co-design, and variability-tolerant architectures that account for manufacturing inconsistencies.

Architectural Specialization and Heterogeneous Computing. Specialized hardware is becoming increasingly critical for energy-efficient scientific computing. However, designing heterogeneous architectures that seamlessly integrate specialized accelerators such as GPUs, FPGAs, neuromorphic, and domain-specific ASICs remains an ongoing challenge.

Chiplets and 3D Stacking: Chiplets provide a modular approach to integrating specialized processing units within a single package, thereby reducing interconnect power consumption and increasing computational efficiency [25, 91]. Open interface standards, such as Compute Express Link (CXL) [92] and UCIe [93], will be essential for ensuring interoperability between chiplet-based accelerators.

In-Memory Computing: Processing-in-memory (PIM) architectures bring computation closer to data storage, thereby reducing energy-intensive data movement. While commercial SRAM-based and DRAM-based PIM solutions exist, new techniques must be developed to enable scientific computing applications to leverage these architectures effectively.

Electronic Design Automation Tools and Prototyping. The development of open-source and commercial EDA tools is critical for advancing energy-efficient hardware. Existing design tools primarily focus on conventional CMOS architectures, making it difficult to explore emerging paradigms such as cryogenic, neuromorphic, and analog computing. Research opportunities include EDA frameworks optimized for heterogeneous, energy-efficient architectures; prototyping test beds to validate emerging hardware technologies for scientific applications; and, publicly accessible chip fabrication and characterization facilities that can accelerate hardware development cycles.

3.2.2 Research Opportunities

Addressing these challenges will enable the scientific computing community to play a proactive role in shaping the future of energy-efficient hardware. Key research directions are

- novel computing models tailored for DOE workloads in superconducting, neuromorphic, and probabilistic computing;
- technology roadmaps for emerging computing paradigms to provide guidance on research investments and long-term impact;
- prototyping and test bed initiatives for cryogenic, photonic, and chiplet-based computing;
- development of EDA tools for emerging architectures and devices to enable energy-efficient circuit synthesis and system-level modeling;
- open hardware ecosystems to facilitate community-driven co-design efforts;
- energy-efficient specialized (domain-specific) architectures and FPGA designs, including analog architectures; and
- heterogeneous computing runtimes and programming models to allow software to dynamically allocate workloads across specialized hardware.

By fostering collaboration across research efforts in hardware design, algorithms, and applications, DOE can accelerate innovation in energy-efficient computing and ensure that scientific workloads benefit from the next generation of computational technologies.

3.3 Data Storage and Management

Data storage, movement, and management are central to the DOE science mission, which supports a wide range of computational and experimental research endeavors. Advances in storage density, archival techniques, and communication technologies have significantly expanded the capabilities of data management systems. At the same time, scientific datasets are growing at an unprecedented rate, driven by new experimental platforms, high-resolution simulations, and the increasing adoption of AI for automated data collection and curation. This trend mirrors the exponential growth of data worldwide [21, 94].

Despite the critical role of data management in scientific discovery, there remains a significant knowledge gap regarding the energy consumption of storage and data management activities. Although existing research has primarily focused on optimizing compute energy efficiency, the unique energy challenges of storage and data movement remain largely unexplored. This is particularly concerning as storage device density increases, leading to declining I/O operations per second (IOPS) per terabyte [95]. As a result, maintaining high I/O performance in future systems may require deploying additional storage devices and potentially increasing overall energy consumption.

Notably, a 2022 ASCR workshop on data management identified several research priorities, including co-design of data services with emerging storage devices and better understanding of data movement patterns [96]. Additionally, AI-driven approaches for data management optimization were identified as a growing research area. AI methods can optimize data placement, indexing, and partitioning to reduce energy-intensive operations, and efficient data processing strategies can mitigate the energy footprint of training and inference for large-scale AI models. Although significant research has explored reducing energy consumption during model training [97, 98], far less attention has been given to the energy impact of data processing operations required for training. Addressing this gap is essential to ensuring the scalability of AI workloads.

3.3.1 Findings and Discussion

Workshop discussions converged on four major themes: Understanding the Energy Usage of I/O and Data Management, Reducing Data Movement and Measuring Its Effect, Data Representations and Layouts for Energy Efficiency, and Energy-Efficient Storage System Design.

Understanding the Energy Usage of I/O and Data Management. A critical barrier to energy-efficient data management is the lack of metrics and measurement capabilities. Today, most HPC centers do not provide granular telemetry data on the energy consumption of storage devices, I/O operations, or data transfers. Without this data, researchers cannot evaluate the energy efficiency of new algorithms, optimizations, or system architectures. For example, more effort is required to understand how the use of mixed precision affects performance and energy efficiency. Recent findings show up to a 1.9× speedup when using mixed precision for training and inference of a U-Net with 64 filters across CPU, GPU, and TPU, but these findings do not include energy efficiency metrics [99].

Participants identified three essential capabilities for developing a deeper understanding of storage energy consumption. First, benchmarks and proxy applications are needed that capture real-world data movement and energy consumption patterns (Section 3.8.3). Additionally, access to HPC telemetry data would enable researchers to study I/O behaviors and evaluate energy-efficient techniques much more readily. Finally, models for predicting energy impact would enable performance-energy trade-off analysis for the wide variety of potential novel storage architectures.

Reducing Data Movement and Measuring Its Effect. Data movement represents one of the most energy-intensive operations in modern HPC systems [100]. The energy required to transfer data is overtaking the energy required to compute the data [101], making intelligent data placement and movement policies essential for energy-efficient computing. To satisfy the skyrocketing data volume and application demand, today's HPC storage servers are equipped with a rising number of fast (e.g., PCIe Gen5/6) and physically compact (e.g., EDSFF) NVMe drives with dramatically increased I/O density. As a result, to fully leverage storage bandwidth and sustain up to tens of millions of IOPS, one needs dozens of cores to busy-drive the I/O parallelism, and this yields dramatic power consumption (e.g., 500 Watts). At the same time, emerging storage devices experiment with richer interfaces and capabilities that can be exploited to reduce data movement, including Zoned Namespaces [102], Flexible Data Placement [103], embedded functions in the form of key-value stores [104], and networking capabilities such as CXL [92] and 1.6 terabit Ethernet [105]. In-network computing leverages programmable network hardware (e.g., switches and NICs), and techniques are being developed for network-wide programming of such a fabric [106]. Commercial products also exist for PIM [107], building on the ideas of computational RAM [108], and are being adapted for in-storage computing [109], for which commercial products also exist. Recent commercial storage systems are also beginning to provide interfaces and support for near- and in-storage processing (referred to as *active storage* or *computational storage* [110]), which is a desired capability that was initially proposed and demonstrated in the context of in-situ data analytics for HPC workflows [111, 112].

The group identified several open research topics, including methods to optimize data placement across storage hierarchies that minimize movement, leveraging AI-driven approaches for automated data partitioning and migration, and exploiting novel storage technologies such as computational storage, near-memory processing, and in-network computing to help reduce data movement overhead.

Data Representations and Layouts for Energy Efficiency. Data movement is overtaking computation as the primary driver of energy consumption in HPC [101]. However, surprisingly little research has focused on alternative data representations and layouts that could reduce this energy consumption. And as mixed-precision data representations are being increasingly exploited in compute hardware, strategies such as mixing precision of stored data or using alternative and more compact representations are hardly used in scientific computing.

The attendees identified three directions for further investigation. They noted that while mixed-precision computation is widely used in AI and scientific computing, the concept of variable-precision storage formats is largely unexplored. Optimizing data structures for locality and compression was identified as an approach that could significantly reduce memory bandwidth and I/O power consumption. Additionally, AI techniques could be used to predict which data should be stored, compressed, or discarded, thereby reducing long-term storage energy costs.

Energy-Efficient Storage System Design. Unlike compute resources, which benefit from scalable power management techniques, storage systems are fundamentally constrained by the need for data availability and durability. Several approaches have been proposed to reduce storage-device-energy consumption. Some approaches have considered reducing and shifting power by increasing utilization [113–123] and moving computation to times and locations with more renewable or cost-efficient energy [124–129]. These reductions are more difficult to accomplish in storage because lowering and shifting power consumption relies on power varying with usage. For example, extending device lifetime leads to higher rates of device failure. Notably, compute can usually just be migrated to a new server, whereas storage is fundamentally stateful. Higher failure rates increase the likelihood of data loss, thereby requiring more capacity for erasure-coding and reducing the benefit of extending device lifetime. Most prior work also assumes data replication, whereas today's data centers use erasure codes [130–132]. As a result, most existing power-saving techniques from computing do not translate directly to storage environments.

Prior work has also considered using fewer, more efficient devices [133–137]. Denser storage devices such as PLC SSDs, HAMR HDDs, and new storage media technologies could reduce the number of servers and racks required to store the same amount of data, thereby lowering power consumption. However, increasing storage density is not straightforward. Denser devices typically do not have proportionally higher I/O and could reduce the IOPS per terabyte and introduce new performance constraints, and HDDs are already creating I/O bottlenecks in data centers. For instance, the bandwidth of Seagate's Exos 18 TB HDD is only 8.4% higher than that of the 10 TB model [138, 139]. Thus, we need to reduce I/O per GB stored, but there is little headroom available, and many storage applications already saturate today's HDD bandwidth. Additionally, write endurance gets worse with higher cell density. PLC is projected to have 16% of the write endurance of today's TLC drives [140].

Essentially, the breakout group settled on two major challenges. First, storage power consumption does not scale with usage. Unlike compute clusters, where power consumption varies with workload intensity, storage systems consume power continuously—even when idle [95]. Second, balancing storage reliability and performance with energy efficiency is not straightforward. Although increasing storage device utilization could improve energy efficiency, higher utilization leads to higher failure rates, necessitating additional redundancy and erasure coding, which introduces new energy costs.

To address these challenges, future research must explore energy-aware caching and tiered storage to optimize data placement and minimize power consumption; dynamic power management strategies that allow storage systems to scale energy usage based on workload needs; and alternative storage media, including DNA-based storage [141], glass storage [142], and tape-based archival solutions.

3.3.2 Research Opportunities

Taken together, the participants identified several critical research directions for energy-efficient data management.

Developing Energy Measurement and Optimization Frameworks. For this goal, we require benchmarks and proxy applications beyond what is available today, that can emulate data movement and energy consumption patterns across sites, experimental facilities, and edge devices. We must:

- create benchmarks and datasets to analyze storage power consumption and I/O energy efficiency; and
- provide open-access telemetry data from HPC centers to enable data-driven optimization research.

Reducing Data Movement for Energy Efficiency. We need intelligent data placement and movement policies for energy-efficient computing, since data movement represents one of the most energy-intensive operations in modern systems. This requires us to:

- explore data-aware scheduling techniques that optimize movement based on energy constraints; and
- investigate AI-driven compression, deduplication, and in-network storage processing.

Optimizing Data Representations for Energy Efficiency. Optimizing data representations has the potential to greatly reduce the energy used by data movement operations. To develop this capability, we must:

- investigate mixed-precision storage formats and energy-aware data layouts; and
- develop automated methods for data reduction and format selection.

Designing Energy-Efficient Storage Architectures. Several approaches have been propposed for reducign the energy used by storage systems. However, more research is required in this area to:

- develop energy-aware storage policies that minimize idle power consumption; and
- evaluate alternative storage media for long-term durability.

By addressing these challenges, the computing community can dramatically improve the energy efficiency of data storage and management to ensure that future scientific workloads remain scalable, durable, and cost-effective.

3.4 Modeling and Simulation

Modeling and simulation (ModSim) have long been foundational techniques for designing and optimizing computing architectures—from transistor-level devices to full-scale HPC systems (see the annual ModSim workshop[1]). These tools enable designers to explore trade-offs in performance, functionality, scalability, and energy efficiency early in the design process. While performance modeling frameworks have evolved to support scalable and accurate co-design, energy modeling remains significantly underdeveloped. The lack of accessible, accurate, and scalable energy modeling tools poses a major barrier to optimizing future DOE computing systems for performance in power-constrained environments.

Workshop participants emphasized the need for credible, open-source ModSim infrastructure to evaluate energy alongside performance *before* fabrication. By integrating energy awareness early in the design pipeline, researchers can better inform architectural decisions, assess the impact of emerging computing paradigms, and support DOE's mission to develop high-performance, scalable computing platforms.

[1] https://www.bnl.gov/modsim

This section outlines critical challenges in energy modeling at the device, architecture, and system levels and provides key research opportunities aimed at bridging these gaps.

3.4.1 Findings and Discussion

During the workshop, participants converged on three basic themes: Device-Level Energy Modeling, Architecture-Level Simulation and Modeling, and System-Level Energy Modeling.

Device-Level Energy Modeling. A fundamental challenge in accurately modeling energy consumption is the limited availability of high-fidelity device models. Many process design kits (PDKs) and device-level simulation tools are proprietary, thereby restricting a researcher's ability to build accurate energy models. Although some open-source alternatives (e.g., OpenROAD, OpenLane, and SkyWater) have emerged, these efforts are limited in scope and maturity. Most publicly available PDKs only support older CMOS nodes (e.g., 90 nm, 130 nm) and do not extend to advanced process nodes or non-CMOS technologies such as photonics, carbon nanotube FETs (CNFETs), or superconducting devices.

Current device-level energy modeling challenges include vendor lock-in and limited access to PDKs, which hinders research on leading-edge technology nodes; slow and computationally expensive modeling workflows that make iterative design exploration infeasible; limited support for emerging computing paradigms, including photonics, analog devices, neuromorphic devices, and quantum-inspired architectures; and, a lack of scalable interfaces for integrating new materials and devices into architectural-level models. To address these challenges, scalable, open-source, and modular modeling interfaces are needed to enable rapid and accurate device-level energy characterization. Additionally, automated tools for generating PDK models could accelerate research in next-generation energy-efficient devices.

Architecture-Level Simulation and Modeling. Architecture-level energy modeling is highly fragmented, with no single widely adopted standard for estimating energy consumption across different architectures. Current approaches generally fall into four categories.

First-principles analytical models: Tools like Aspen [143], CACTI [144], and McPAT [145] estimate memory and processor energy usage. However, these models are often outdated and do not accurately reflect modern architectures.

Empirical projection methods: Tools such as Wattch [146] and AccelWattch [147] extrapolate power data from existing systems to future architectures. Unfortunately, these models often fail when applied to new architectures, workloads, or configurations.

Low-level SPICE-based circuit models: While providing accurate energy estimates, these do not scale to large, complex architectures.

AI-based energy models: These use empirical data to predict energy consumption [148]. Although promising, these models require large, high-quality datasets to achieve generalization.

Given the heterogeneity of modern and future architectures, energy modeling approaches must become more modular, adaptable, and scalable. The community must develop composable, open-source modeling frameworks that allow researchers to integrate varied energy models at different fidelity levels.

System-Level Energy Modeling. System-level modeling is particularly challenging for exascale and post-exascale HPC systems, in which complex interactions between processors, memory, interconnects, and cooling must be considered. Current DOE efforts, such as digital twins for leadership-class systems [149], provide a promising direction for early-stage co-design. However, existing system-level energy models suffer from several limitations. First, because many vendor-specific cooling techniques remain proprietary, thermal and cooling models are incomplete. Second, a lack of integrated energy modeling across multiple layers makes it difficult to propagate device-level energy metrics to full-system simulations. Finally, because many existing system models require impractically long runtimes for large-scale design space exploration, simulation tools tend to have poor scalability.

To enable scalable and energy-aware system co-design, new methodologies must be developed to combine multiple levels of fidelity, thereby allowing trade-offs between accuracy and simulation speed. Additionally, integrated co-design environments that simultaneously model performance, energy, and thermals would significantly enhance the ability to optimize future DOE systems.

3.4.2 Research Opportunities

Workshop participants identified several key research opportunities to transform energy modeling, simulation, and emulation tools.

Developing Open, Integrated, and Modular Energy Modeling Frameworks. The lack of standardization across energy modeling tools leads to fragmented, inconsistent, and non-reusable results. A major research opportunity lies in developing modular, open-source energy modeling frameworks that provide

- standardized interfaces for integrating energy models across different abstraction levels (device, architecture, system);
- support for heterogeneous computing, including accelerators, chiplets, and non-CMOS technologies;
- scalability across different workloads, from scientific computing to AI-driven workflows; and
- extensibility to allow researchers to contribute new energy models without requiring full-system reimplementation.

Building Open-Source Device Simulators and PDKs. Given the challenges posed by proprietary PDKs, there is a pressing need for open-source, first-principles device simulation tools. Key research opportunities include

- expanding open Technology CAD tools to support leading-edge and emerging device technologies;
- developing scalable PDK generators that can adapt to different materials and node sizes; and
- integrating open device models into larger simulation frameworks, thereby enabling full-stack energy modeling.

Accelerating Early-Stage Design Space Exploration. To improve the speed and accuracy of design space exploration, new applied mathematics and AI techniques should be incorporated into simulation workflows. Key research directions include

- mathematical optimization techniques for automated energy-performance trade-off analysis;
- UQ methods to provide confidence bounds on energy estimates;
- AI-driven surrogate models to accelerate energy estimation without sacrificing accuracy; and
- visual analytics tools for interpreting and navigating large-scale design spaces.

Longevity of ModSim Tools. To ensure continuity and impact, the community must commit to the maintenance and support of the ModSim infrastructure and workforce development for energy modeling research. Priorities include

- establishing shared, community-driven repositories for maintaining energy modeling tools and
- providing testbeds and computational resources for large-scale energy modeling experiments.

3.4.3 Summary

Energy ModSim will be critical to the success of future DOE computing systems. By developing scalable, accurate, and accessible modeling tools, the research community can drive energy-efficient innovation across the entire computing stack. The opportunities outlined in this section will help position DOE at the forefront of vibrant ModSim computing research.

3.5 Facility to Edge

The increasing heterogeneity of computing architectures has the potential to revolutionize computing across the facility-to-edge continuum [12]. This continuum spans the entire computational spectrum—from data origination at sensors, edge processing near acquisition points, and intelligent or quantum entangled network routing [150, 151] to remote HPC nodes, with control directives and decisions often flowing back to the edge. Efficiently orchestrating analysis and decision-making across this highly distributed infrastructure requires granular modularization of algorithms and hardware descriptions to ensure that computations are performed at the most energy-efficient and latency-sensitive locations.

Many DOE science applications increasingly demand real-time, distributed computing to analyze data at the edge and dynamically adjust parameters without incurring the latency of central HPC processing [152]. Examples include

- materials synthesis and nuclear reactor monitoring, in which real-time sensor analysis and parameter tuning are critical for stability;
- smart grid management, which requires low-latency adjustments to efficiently balance energy supply and demand; and
- autonomous systems and remote sensing, for which real-time AI-driven decision-making enhances responsiveness in dynamic environments.

These scenarios introduce new opportunities for deploying energy-efficient computing paradigms across the facility-to-edge hierarchy. Moreover, they provide an avenue for coordinating distributed computing resources to dynamically optimize energy use while maintaining the required performance and reliability for scientific discovery.

3.5.1 Findings and Discussion

Challenges in Energy Measurement for Facility-to-Edge Computing. Despite advances in facility-to-edge integration, energy efficiency remains an underexplored dimension. Without improved energy measurement, modeling, and prediction capabilities, optimizing distributed workflows for energy efficiency will remain infeasible.

The workshop identified three key themes in energy measurement across distributed infrastructures. First, distributed computing infrastructures involve complex measurement requirements: multiple layers of hardware, including sensors, controllers, near-edge processors, networked data centers, and HPC facilities. Measuring energy consumption across these layers requires new instrumentation and telemetry strategies. Second, facility-to-edge architectures often span multiple geographic regions and administrative domains, with jurisdictional and policy barriers that make energy transparency and measurement sharing a challenge. Finally, many facility-to-edge workflows operate over shared infrastructure (e.g., wide-area networks), making it difficult to isolate energy costs attributable to specific computations.

Limitations of Current Energy Awareness in HPC Facilities. Within traditional HPC facilities, energy efficiency is not an explicit user concern. Facility costs, including energy consumption, are typically abstracted into node-hour allocations, thereby incentivizing time efficiency rather than energy efficiency. Even when energy telemetry and user-controllable dynamic voltage and frequency scaling (DVFS) or power capping mechanisms are available, they are seldom utilized except by researchers specifically focused on energy-efficient computing.

These issues are expected to worsen in facility-to-edge workflows, where heterogeneous computing resources (e.g., FPGAs, neuromorphic processors, in-network computing) offer potential energy savings but also where users currently lack mechanisms to optimize for energy across distributed environments. The lack of fine-grained energy telemetry limits energy-aware decisions and complicates holistic energy-efficient workflow optimization.

Leveraging Compute-in-X for Energy Efficiency. Distributed facility-to-edge workflows introduce new opportunities to deploy energy-efficient computing paradigms. For compute-in-sensor, data is processed directly at the point of acquisition, thereby reducing energy-intensive data transmission. Compute-in-memory minimizes redundant data movement by performing computations near or within storage elements. For compute-in-network, in-transit data processing optimizes bandwidth and latency constraints.

Effective resource steering is essential for maximizing these energy-saving opportunities. However, current DOE computing environments provide only limited architectural diversity, making comparative energy studies difficult.

Although cloud providers offer a wider variety of hardware, DOE's production computing lacks a funding model for cloud-based scientific workloads, and commercial cloud environments provide limited energy visibility.

Furthermore, many energy-efficient hardware accelerators (e.g., FPGAs, custom ASICs) remain inaccessible to the broader scientific community because of their steep programming barriers. This restricts their adoption to low-level experts rather than allowing domain scientists to easily leverage these accelerators in their research.

Multifacility Coordination and Standardization. A single facility will not host every emerging architecture. Future facility-to-edge computing will depend on interoperability between DOE sites, private-sector platforms, and distributed computing resources. To enable this, research opportunities exist in standardized interfaces for resource telemetry to facilitate measurement, monitoring, and coordination across computing sites; algorithm modularity to ensure that neural network layers, computational kernels, and scientific workflows can be flexibly mapped to optimal computing resources; and federated computing frameworks to enable secure, real-time workload distribution across heterogeneous environments.

3.5.2 Research Opportunities

Workshop participants identified several key research opportunities in facility-to-edge computing, including Energy-Aware Telemetry and Provenance Tracking, Challenge Benchmarks for Facility-to-Edge Computing, Secure and Federated Computing, and End-to-End Energy Models for Facility-to-Edge Workflows.

Energy-Aware Telemetry and Provenance Tracking. As scientific computing shifts toward intelligent dataflow paradigms, data provenance and energy tracking will become essential. Future research should explore the following:

- embedding energy and latency metadata into provenance records to enable cost-aware optimizations at both user and system levels,
- developing real-time feedback mechanisms for users to understand the energy impact of their workflows, and
- integrating energy tracking with intelligent routing to ensure that data and computations follow energy-optimal paths from sensors to HPC facilities.

Challenge Benchmarks for Facility-to-Edge Computing. DOE would benefit from standardized, scientifically relevant, and scalable benchmarks and requisite datasets to evaluate facility-to-edge workflows. Unlike private-sector benchmarks, which often highlight specific hardware architectures, DOE benchmarks should focus on real scientific use cases that drive mission-critical research. Key research directions include the following:

- curating representative facility-to-edge workflow benchmarks that cover AI-assisted experiments, remote sensing, and extreme-scale simulations;
- encouraging industry partners to adopt DOE benchmarks to ensure that new architectures align with scientific computing priorities; and
- establishing an open repository of facility-to-edge workloads to enable cross-institutional benchmarking and collaboration.

Secure and Federated Computing. Facility-to-edge computing will require secure data sharing and workload execution across multiple institutions. Unlike traditional HPC environments, which operate within well-defined security perimeters, facility-to-edge workflows span heterogeneous, distributed infrastructure. Future research has the potential to impact the following areas:

- securing federated AI model training to ensure intellectual property protection while enabling collaborative AI development;
- developing authentication and access control mechanisms to support multi-institutional data exchange while maintaining security; and
- establishing governance frameworks for cross-agency coordination to facilitate DOE-private sector collaboration in a secure manner.

End-to-End Energy Models for Facility-to-Edge Workflows. To optimize energy efficiency across the facility-to-edge continuum, comprehensive energy models must be developed. Key research areas are as follows:

- analytical models for conceptual design to enable early-stage performance-energy trade-off analysis;
- high-fidelity digital twins to enable detailed energy impact simulations for large-scale distributed workflows; and
- real-time operational energy models that integrate AI-based energy predictions to support dynamic resource allocation decisions.

3.5.3 Summary

Facility-to-edge computing represents a transformative opportunity for DOE science by enabling real-time decision-making, intelligent workload distribution, and energy-efficient computing across distributed infrastructures. Addressing the energy challenges of this paradigm will require new research in telemetry, secure federated computing, and holistic energy modeling. By investing in these areas, DOE can establish a scalable, efficient, and resilient facility-to-edge computing ecosystem that meets the demands of future scientific discovery.

3.6 Resource Management

Efficient resource management is critical for optimizing both performance and energy efficiency in large-scale computing systems, especially within data centers and HPC environments. Traditionally, resource management strategies have emphasized maximizing performance while ensuring system reliability. However, the increasing complexity and scale of modern workloads, alongside rising energy demands from heterogeneous computing architectures, require a fundamental shift in these strategies [153]. Future approaches must balance performance, energy efficiency, and portability while effectively supporting diverse workloads and evolving hardware architectures.

Modern computing systems often operate at suboptimal utilization levels, resulting in wasted computational capacity, unnecessary energy consumption from static power losses in idle resources, and higher capital costs for system operators. A significant portion of system power is provisioned for peak workloads, but these peaks occur infrequently. As a result, many resources remain underutilized, thereby increasing operational costs and reducing overall efficiency. Future systems must support more dynamic and fine-grained resource allocation methods, including multitenancy, oversubscription, energy-aware job scheduling, and workload orchestration.

Achieving this shift requires integrating real-time telemetry, intelligent orchestration techniques, and cross-facility resource coordination. By leveraging emerging technologies such as CXL, near-memory and in-memory computing, and AI-driven optimization [154, 155], future HPC systems can reduce energy consumption while maintaining high computational throughput.

3.6.1 Findings and Discussion

Participants discussed six main topics during the workshop. An emergent crosscutting theme for these topics was current resource management practices allocating computing capacity based on entire nodes and fixed-duration job assignments. This coarse-grained approach often leads to inefficiencies because workloads rarely utilize all allocated resources throughout their execution. Additionally, traditional scheduling methods are largely static and do not adjust dynamically to changes in workload demand, leading to unnecessary energy consumption. The six discussion topics are listed below.

Static Resource Allocation and Poor Utilization.. Existing scheduling systems do not adapt to workload variability [156, 157]. They allocate resources at the start of execution and maintain these allocations regardless of actual utilization, leading to wasted energy.

Power is Not a Managed Resource. Power delivery infrastructure is provisioned based on peak usage scenarios, which are rarely reached in practice. This simplifies operation by ensuring that sufficient power is available to run any workload, but results in systems that are overprovisioned [158–161]. Power management (i.e. coordinating power use across hardware components and workloads) would enable larger HPC systems to be installed without costly infrastructure expansions, but this is not widely done. Furthermore, power profiling tools are limited in their ability to track and optimize power usage across different components, including storage, memory, and network infrastructure.

Limited Fine-Grained Telemetry. Many HPC centers lack sufficient telemetry to provide real-time feedback to users or system administrators on power consumption and resource utilization [162]. Without this data, optimization is difficult, and energy-efficient practices cannot be effectively enforced.

Siloed Resource Management. Current HPC facilities operate independently, without mechanisms for sharing compute or power resources across sites. In contrast, cloud providers leverage multiple data centers and integrate renewable energy sources dynamically [163, 164], a capability largely absent in traditional HPC environments.

Complexity of Multitenancy and Resource Pooling. Sharing resources among multiple workloads can lead to contention, performance degradation, and security concerns. While multitenancy and oversubscription can improve resource utilization, they introduce challenges in scheduling, monitoring, and ensuring fairness across users [165].

Energy Grid Integration and Stability. Large fluctuations in power consumption caused by HPC workloads (e.g., AI model training) can create instability in power grids [166, 167]. Better coordination between computing facilities and grid operators is required to manage energy demand dynamically.

3.6.2 Research Opportunities

From these findings, the participants identified several categories of research opportunities, including Optimizing Resource Utilization Through Sharing and Disaggregation, System Orchestration and Intelligent Scheduling, and Enabling Cross-Facility Resource Coordination.

Optimizing Resource Utilization Through Sharing and Disaggregation. *Multitenancy and Fine-Grained Resource Allocation:* Future HPC systems should support dynamic sharing of processing elements, memory, and interconnects across workloads. This will require hardware and software support for secure multitenancy, efficient workload placement policies to minimize interference, and dynamic resource partitioning techniques to allow fine-grained allocation.

Power Oversubscription and Energy-Proportional Computing: HPC facilities should explore oversubscription strategies to deploy more hardware than can be fully powered simultaneously, relying on statistical variations in workload intensity to maintain overall power consumption within limits. Research opportunities exist in developing models that predict power demand and optimize power budgets dynamically, enabling adaptive power capping to maximize resource availability without exceeding power limits, and improving energy-proportional computing by designing systems that scale power consumption in response to workload demand.

Resource Disaggregation and Pooling: Emerging interconnect technologies such as CXL and high-speed optical links enable flexible resource disaggregation, where compute, memory, and storage resources can be dynamically assigned based on workload needs. Research directions include optimizing memory and compute pooling strategies for different workloads, investigating new data movement and placement policies to reduce latency overheads, and developing software frameworks for managing disaggregated resources at scale.

System Orchestration and Intelligent Scheduling. *Dynamic and AI-Driven Resource Orchestration:* AI techniques should be integrated into resource management frameworks to enable real-time, adaptive scheduling. Key areas of research include developing reinforcement learning–based schedulers that dynamically adjust resource allocations, using historical workload data to predict resource demand and optimize job placement, and integrating real-time energy and performance telemetry into job scheduling decisions.

Grid-Aware Scheduling and Renewable Energy Integration: HPC centers should adopt scheduling strategies that optimize for both computational performance and grid stability. Research opportunities exist to develop algorithms that schedule workloads based on availability of renewable energy, coordinate power consumption across multiple facilities to balance grid demand, and utilize data center power storage to stabilize energy consumption patterns.

Enabling Cross-Facility Resource Coordination. *Cooperative Resource Sharing Across HPC Facilities:* Future HPC ecosystems should support federated scheduling, allowing workloads to move seamlessly between facilities based on resource availability and energy considerations. Key research challenges include developing standard interfaces for workload migration and job orchestration, creating federated authentication and security models to support cross-facility execution, and establishing policies for energy-aware job scheduling across multiple data centers.

3.6.3 Summary

By advancing research in resource sharing, orchestration, and energy-aware scheduling, DOE facilities can set a new standard for energy-efficient computing at scale. To this end, the participants also identified metrics for evaluating progress and the potential impact of advancements in these areas.

Impact of Breakthroughs in Resource Management. Achieving the research objectives outlined above will enable more efficient utilization of HPC resources while significantly reducing energy consumption.

Increased Utilization: Higher system efficiency from optimizing resource allocation will enable facilities to handle more workloads with the same infrastructure.

Lower Energy Costs: Power consumption will be reduced through adaptive power management and workload balancing.

Improved Grid Stability: Energy demand fluctuations will be mitigated through better coordination between HPC facilities and power grid operators.

Metrics for Evaluating Progress. To ensure that research efforts align with the desired goals, success will be measured through utilization rates of key resources (compute, memory, storage, bandwidth, power), reductions in system idle power and improvements in energy-proportional computing, effectiveness of scheduling strategies in minimizing power surges and stabilizing energy demand, and the ability of new orchestration frameworks to dynamically allocate resources across facilities.

3.7 Programming Systems

The increasing heterogeneity of computing architectures—spanning from traditional CPUs, GPUs, and accelerators to emerging paradigms such as neuromorphic, PIM, dataflow, and probabilistic computing—requires a fundamental shift in programming systems. Achieving energy-efficient computation on these diverse architectures requires innovations in operating systems, runtime environments, and programming models to dynamically adapt to hardware characteristics and workload demands.

The workshop participants identified the need for intelligent operating systems and runtime systems capable of efficiently mapping scientific workflows onto ensembles of heterogeneous computing systems. Emerging architectures will often operate alongside traditional computing platforms, which requires seamless orchestration between them. This also raises critical open questions: How can we best map and execute complex scientific workflows across such diverse hardware ecosystems? How can energy efficiency be embedded into compilers, runtime systems, and operating systems to dynamically optimize performance?

These topics highlight the need for a holistic rethinking of programming systems to ensure that energy efficiency is a first-class design principle rather than an afterthought.

3.7.1 Findings and Discussion

To address energy efficiency in programming systems, the participants categorized the discussion along three key areas: Languages and Compilers, Runtime Systems, and Operating Systems.

Languages and Compilers. Programming languages and compilers play a fundamental role in translating high-level abstractions into efficient executable code. As architectures become more diverse, compilers must evolve to optimize energy consumption alongside traditional performance metrics. Achieving this requires compiler analyses capable of identifying energy-intensive code patterns; energy-aware optimizations such as dynamic precision scaling, adaptive scheduling, and power-efficient instruction selection; code generation that effectively leverages hardware power management features, including voltage scaling and near-memory computing; and language-level constructs that enable developers to specify power constraints, energy budgets, and quality-of-service trade-offs.

Farther over the horizon, emerging architectures such as analog, neuromorphic, and photonic computing lack robust software stacks, thereby forcing programmers to work at low levels of abstraction. High-level programming models, compiler frameworks, and automatic code generation tools are critical for unlocking the potential of these architectures. Research must focus on developing novel compilation techniques that automatically map high-level algorithms to energy-efficient hardware, perform trade-off analyses between energy and performance during compilation, and generate energy-efficient code variants optimized for different workload scenarios. Ultimately, compilers must evolve to support both traditional and emerging computing paradigms to enable portability and energy-aware optimizations across heterogeneous platforms.

Runtime Systems. Runtime systems are critical to managing the interaction between applications and hardware. They enable dynamic resource allocation, power-aware scheduling, and adaptive data orchestration strategies. The workshop discussions identified several key challenges and opportunities in runtime system research.

Cross-Layer Energy Awareness: Modern runtime systems must integrate energy awareness into task scheduling, memory management, and data movement decisions. Achieving this requires developing energy-conscious APIs for libraries such as Kokkos (scientific libraries), MPI (communication), and HDF5 (I/O); embedding power monitoring and control mechanisms within runtime systems; and using real-time feedback loops to dynamically adjust execution parameters.

Autonomous Decision-Making: Future runtime systems should leverage AI-driven techniques to autonomously optimize power efficiency. This includes using reinforcement learning to adjust scheduling and resource allocation, implementing workload profiling for predictive energy optimization, and developing runtime policies that balance energy savings with performance guarantees.

Energy Complexity Theory: Just as time and space complexity are foundational concepts in algorithm design, an analogous energy complexity metric is needed. Establishing a rigorous theoretical foundation will provide a common framework for reasoning about energy-efficient algorithms, enable performance-energy trade-off analyses at various levels of the computing stack, and inform the design of energy-aware scheduling and resource management policies. A shared experimental infrastructure will be critical for evaluating new runtime techniques and ensuring reproducibility in energy-efficient computing research.

Operating Systems. Modern operating systems must evolve to support increasingly heterogeneous and energy-aware computing environments. Traditional operating system designs treat accelerators, AI processors, and novel computing devices as independent peripherals that require explicit application management. However, future systems must integrate these components into a unified, energy-optimized execution environment.

Energy-cooperative resource management: Operating systems face several hurdles in emerging energy-efficient computing environments. Overcoming these challenges requires developing operating system–level schedulers that intelligently assign tasks based on power constraints and energy efficiency goals; coordinating power management across different hardware components, including CPUs, GPUs, memory, interconnects, and devices using emerging computing technologies; creating energy-aware scheduling policies that optimize energy efficiency across a spectrum of timescales; and, designing mechanisms to expose fine-grained power metrics to compilers, runtime systems, and applications.

Operating System Abstractions for Energy Efficiency: New operating system abstractions are required to manage the complexity of energy-aware computing. These include APIs for fine-grained power monitoring and control, energy-aware memory management strategies that minimize data movement, DVFS coordination across heterogeneous processors, and thermal-aware task scheduling to optimize cooling efficiency and energy usage. By embedding energy efficiency into the core of an operating system, the computational stack can seamlessly integrate energy-aware optimizations across all layers.

3.7.2 Research Opportunities

Based on these findings, the participants identified several key research directions.

Developing Energy-Aware Programming Models and Abstractions. Energy awareness could be enhanced in programming models by

- designing new programming languages and DSLs that allow users to explicitly specify energy constraints and trade-offs;
- developing compiler techniques that integrate energy complexity analysis into code transformations; and
- extending intermediate representations to incorporate power-aware optimizations.

Advancing Autonomous, Energy-Aware Runtime Systems. Runtime-level energy awareness could be facilitated through

- investigating reinforcement learning for dynamic workload scheduling and power management;
- developing predictive modeling techniques for real-time energy optimization; and
- making available experimental testbeds for benchmarking runtime energy efficiency.

Innovating Operating System–Level Energy Management Strategies. Similar approaches could be applied at the operating system level:

- implementing energy-aware scheduling mechanisms that adapt dynamically to workload needs;
- enhancing operating system support for fine-grained power monitoring and control; and
- developing power-efficient memory and storage management policies.

Establishing a Unified Energy Complexity Framework. Work at all layers could be tied together by

- formalizing energy complexity metrics to evaluate and compare energy-efficient algorithms;
- integrating energy-aware performance models into compiler and operating system decision-making; and
- developing theoretical models that quantify trade-offs between energy, performance, and accuracy.

3.7.3 Summary

Programming systems must evolve to meet the challenges posed by emerging energy-efficient architectures. Achieving this transformation requires research across multiple layers of the software stack, from compilers and runtime systems to operating systems and resource managers. By embedding energy efficiency as a fundamental principle, future computing platforms can balance performance with other criteria, ensuring that scientific and HPC remains viable.

3.8 Crosscuting Topics

The workshop attendees identified several crosscutting topics that transcend individual breakout groups. On the final day of the workshop, dedicated sessions focused on these overarching issues and emphasized the need for a holistic approach to energy-efficient computing. This section collates and prioritizes the most critical crosscutting themes and their corresponding recommendations.

3.8.1 C1: Testbeds and Prototypes

Discussion Summary. Testbeds and prototype systems are essential for evaluating energy-efficient computing technologies. The discussion underscored the importance of open technologies for long-term research and avoiding dependence on proprietary vendor solutions that primarily cater to commercial markets. Testbeds should facilitate co-design methodologies that integrate hardware and software development to enable the validation of energy models, real-world application testing, and performance benchmarking.

A key distinction was made between testbeds (platforms for evaluating existing and near-production technologies) and prototypes (early-stage implementations of novel hardware-software co-designs). The participants strongly recommended testbeds be made available to evaluate and develop new computing concepts from national labs, universities, and pre-competitive vendor research. Furthermore, experiences with these testbeds and prototypes can inform future system procurements and research initiatives.

Recommendations.

- Establish testbeds that support real DOE scientific workloads and include capabilities to measure traditional application performance as well as energy efficiency and power utilization.
- Integrate prototype hardware into instrumented testbeds for benchmarking, validation, and software development.
- Leverage the growing domestic infrastructure (e.g., DOD Microelectronics Commons Hubs) to support hardware prototyping for scientific computing to ensure that ASCR-funded research remains aligned with emerging trends.
- Strengthen connections between testbed development and ModSim to improve energy estimation accuracy and provide verification and validation for architectural models.

3.8.2 C2: Metrics for Energy Efficiency and Performance

Discussion Summary. Defining standardized, multilevel energy efficiency metrics is crucial to optimizing scientific computing systems. Current metrics—such as time-to-solution, utilization, average power, and throughput—provide a partial view but fail to capture the full impact of energy-aware optimizations. The workshop identified three fundamental perspectives for energy-aware metrics.

- *User-Level Metrics*: Measuring the useful work per unit of energy, where useful work is application-dependent (e.g., convergence in AI training, fidelity in simulations).
- *System-Level Metrics*: Evaluating resource utilization per unit of energy to ensure that HPC resources are optimally employed and reduce waste.
- *Facility-Level Metrics*: Capturing power usage effectiveness (PUE) and total lifecycle energy costs and integrating capital, operational, and embodied energy expenses.

A major challenge is the lack of power measurement granularity, particularly for data movement. Users need better tools to estimate which resources are most energy efficient for their workloads.

Recommendations.

- Develop standardized energy efficiency metrics across user, system, and facility levels.
- Incorporate UQ to improve the reliability of energy efficiency measurements.
- Improve energy profiling tools to provide fine-grained insights into energy costs of computation, data movement, and I/O operations.
- Ensure that power and energy become schedulable resources to incentivize efficiency in user workloads.

3.8.3 C3: Benchmarking (Including Proxy Applications and Datasets)

Discussion Summary. Accurate benchmarking is critical for understanding and improving energy efficiency in scientific computing. However, the current ecosystem lacks representative benchmarks, proxy applications, and realistic datasets tailored for DOE workloads. The discussion identified three key gaps.

- *Benchmarks for energy efficiency*: Existing benchmarks focus on performance but fail to measure energy-related parameters such as processor power settings or node temperature.
- *Proxy applications*: These are crucial for evaluating new techniques, but existing proxy applications lack documentation on which real-world behaviors they emulate.
- *Realistic datasets*: Many datasets are either "toy problems" or overly sanitized due to security concerns, making them less representative of actual scientific workloads.

Recommendations.

- Develop energy-aware benchmarks that assess energy efficiency across diverse hardware platforms.
- Improve documentation and input configurations for proxy applications to ensure they accurately reflect real scientific workloads.
- Encourage the publication of real-world datasets with representative numerical ranges and data structures for evaluation.

3.8.4 C4: Hardware-Software Integration

Discussion Summary. Effective hardware-software co-design is a moving target due to rapidly evolving heterogeneous hardware and the increasing dominance of AI-driven workloads. Traditional approaches to resource management and application optimization must adapt to emerging architectures such as PIM, in-network computing, and disaggregated memory. The discussion highlighted the importance of data movement awareness, runtime adaptability, and fine-grain task placement.

Recommendations.

- Develop hardware-aware programming models that allow applications to explicitly express data locality and energy constraints.
- Improve runtime systems to optimize data placement dynamically based on system conditions.
- Provide energy-aware job scheduling policies that consider memory hierarchy, compute-to-memory distances, and in-network processing capabilities.

3.8.5 C5: Algorithm-Hardware (Paradigm) Integration

Discussion Summary. The discussion emphasized that co-design should not predefine the solution but rather focus on solving energy efficiency challenges holistically. Too often, co-design efforts are biased toward proving a specific technology rather than objectively evaluating what works best for energy-efficient computing.

Three key barriers were identified:

- *Avoiding premature commitment to specific technologies:* Co-design efforts should explore multiple solutions rather than forcing a specific approach.
- *Faster feedback loops for progress assessment:* Current funding structures discourage rapid iteration or changes in team composition, even when new insights emerge.
- *Improving cross-disciplinary communication:* Different communities (hardware, algorithms, applications) operate on different timelines and methodologies, making effective collaboration challenging.

Recommendations.

- Develop mechanisms for iterative, objective evaluation of emerging paradigms to ensure that co-design efforts remain technology-agnostic.

- Encourage flexible funding models that allow for team restructuring if project goals evolve.
- Prioritize ModSim efforts that bridge the communication gap between hardware designers and algorithm developers.

3.8.6 Summary

Crosscutting topics such as testbeds, metrics, benchmarking, hardware-software co-design, and algorithm-hardware integration will be critical enablers of next-generation energy-efficient computing. Addressing these challenges requires a combination of fundamental research, community collaboration, and policy-driven infrastructure development. The DOE ASCR program is uniquely positioned to spearhead these efforts, ensuring that future scientific computing ecosystems are optimized for both performance and energy efficiency.

4 Priority Research Directions

The workshop participants identified five Priority Research Directions (PRDs) that provide a roadmap for achieving transformational improvements in energy efficiency for scientific computing. These PRDs emphasize a holistic approach, recognizing that gains in a single area are insufficient; instead, coordinated innovations are required across hardware devices, algorithms, software ecosystems, data management, and modeling. The following sections detail these research priorities, outlining the key questions and opportunities necessary to sustain scientific discovery.

PRD1: Co-design energy-efficient hardware devices and architectures for important workloads

Key Questions: (a) How can energy-efficient devices using innovative compute methods be designed for scientific applications at scale? (b) How can simulation and modeling tools help evaluate and develop new devices, circuits, and architectures? (c) How can architects identify key HPC application kernels for specialized hardware? (d) What methods can best support heterogeneous integration while minimizing energy loss at hardware component interfaces?

This PRD calls for a transformative, cross-disciplinary research agenda to co-design energy-efficient hardware architectures and devices tailored to critical DOE workloads. Future computing platforms must address energy consumption holistically—from the device level through architecture and system integration—while targeting the unique needs of DOE applications that span HPC, edge computing, and emerging continuum workflows.

The community must invest in a systematic evaluation of diverse emerging computing paradigms, including analog, stochastic, optical, cryogenic, neuromorphic, quantum, and biologically inspired approaches. Early-stage R&D should include device- to system-level modeling, simulation, and physical prototyping to enable informed decisions about viability, performance, and manufacturability. Emphasis should be placed on assessing integration potential and energy efficiency within realistic scientific computing contexts.

To maximize the impact of specialized hardware, this PRD recommends identifying and characterizing DOE-relevant computing kernels with significant potential for energy savings and performance acceleration. Supporting the integration of specialized components into production-scale heterogeneous systems will require research into chiplet standards, packaging technologies, and interface protocols that minimize data movement and overheads.

DOE should enable access to advanced EDA tools, open-source modeling environments, and affordable prototyping facilities to lower the barriers to entry for device-architecture co-design. These capabilities must support the collaborative development of proof-of-concept systems that run real scientific applications and enable meaningful benchmarking of performance and energy efficiency.

This PRD will catalyze a shift from general-purpose computing to energy-aware, application-driven architectures, thereby positioning DOE to lead in the development of next-generation scientific computing systems.

PRD2: Define the algorithmic foundations of energy-efficient scientific computing

Key Questions: (a) Can energy complexity measures for algorithms be used to evaluate algorithm-hardware combinations accurately? (b) How can telemetry of the current execution environment inform algorithmic choices in real time? (c) How can knowledge of the algorithm inform execution to improve energy efficiency? (d) What gains in energy efficiency are possible when focusing on scientific campaigns as a whole rather than focusing on individual tasks or operations? (e) How can data-driven algorithms or models improve over conventional simulation?

Maximizing energy efficiency requires explicit algorithmic reasoning (i.e., decisions about which algorithms to use, how to tune their parameters, and how to structure data and computation in ways that minimize energy use). This goes beyond traditional time-centric performance models because it demands a foundational theory of energy complexity that parallels asymptotic analysis in classical algorithmics. Such a theory must strike a balance between cost realism—by capturing key features of energy usage across hardware—and simplicity—by enabling widespread adoption and productivity in algorithm development.

Crucially, the most energy-efficient algorithm for a particular problem may not be the one with optimal asymptotic behavior; it may instead depend on hardware characteristics, memory hierarchies, energy costs of communication, and/or other environmental factors. Therefore, new frameworks are needed to allow algorithms to introspect and adapt to the dynamic execution environment.

Furthermore, energy efficiency must be addressed not just at the level of isolated kernels but across the scope of full scientific workflows and campaigns. For instance, the choice of surrogate model architecture in a simulation-analysis loop may depend on how often the surrogate will be used, requiring global energy optimization across training and inference phases.

Progress in this area will require new programming abstractions, system support, and performance models that enable algorithms to reason about and respond to their energetic context, laying the theoretical and practical foundation for next-generation energy-aware scientific computing.

PRD3: Reconceptualize software ecosystems for energy efficiency

Key Questions: (a) How can software ecosystems adapt for energy efficiency in innovative heterogeneous architectures? (b) How should the foundations of software development shift to enable energy-efficient computing technologies? (c) How can developers program new energy-efficient hardware productively for complex scientific workflows?

Realizing the full potential of energy-efficient computing requires a fundamental rethinking of the software ecosystem—from programming languages to operating systems—to prioritize energy as a first-class objective alongside performance and correctness. As scientific computing platforms grow more heterogeneous with the introduction of analog, neuromorphic, photonic, and chiplet-based architectures, existing software abstractions and tool chains are becoming increasingly inadequate.

This PRD calls for the design of next-generation software systems that are deeply aware of energy implications and can operate across a continuum of computing resources, from edge devices to exascale systems. A central challenge is to ensure that productivity and portability are preserved as energy-aware systems are developed.

Because the software topic is very broad, we highlight the following key research areas: (1) developing new programming languages and compiler infrastructures that generate energy-optimal code for heterogeneous architectures; (2) creating runtime systems that intelligently orchestrate workloads across diverse hardware components while minimizing energy overhead; (3) building energy-aware libraries that abstract hardware complexity without sacrificing efficiency; (4) designing operating systems that dynamically manage resources based on energy-performance tradeoffs, thermal constraints, and workload characteristics; (5) advancing software tools and telemetry systems that provide actionable insight into energy behavior for developers and system operators; and (6) streamlining abstractions, interfaces, and protocols to reduce energy costs across the software stack.

Outcomes of this research include the seamless integration of emerging architectures, automated energy-aware code generation, and new system-level policies for workload execution. A reimagined software ecosystem for energy-efficient computing will empower scientists to harness complex, evolving hardware for demanding applications while minimizing power consumption.

PRD4: Enable energy-efficient data management for data centers, instruments, and users

Key Questions: (a) What strategies can minimize energy use in data movement across applications, systems, and distributed resources? (b) How can new storage devices be designed and leveraged to lower energy consumption of I/O and data management? (c) How can system architects balance I/O performance with energy-efficient data movement?

As scientific workflows increasingly span geographically distributed facilities—combining HPC centers, experimental instruments, and edge computing devices—data management emerges as a key determinant of system energy efficiency. Today's storage systems remain largely energy-agnostic and are typically optimized for performance and capacity under the assumption of a centralized architecture. This presents a critical mismatch for modern workflows, in which the energy costs of data movement across memory hierarchies, nodes, and wide-area networks can far exceed those of computation.

This PRD calls for a fundamental rethinking of data management and storage to support energy-efficient, high-performance scientific computing across distributed environments. Energy-aware data management must become a first-class concern at all layers of the system stack—from storage devices and interconnects to file systems, data formats, and workflow orchestration frameworks.

Key research directions include (1) developing quantitative models and measurement techniques for characterizing the energy costs of data movement and storage across diverse architectures; (2) designing and evaluating energy-aware data movement strategies that operate across memory, storage, and network layers while preserving application performance; (3) leveraging emerging storage devices with rich interfaces (e.g., computational storage, zoned namespaces, and memory-semantic protocols) to reduce I/O and data motion; (4) creating energy-efficient data representations and layouts, including lower-precision formats and layout-aware data placement strategies; (5) extending techniques such as power-aware workload scheduling and migration to inherently stateful storage systems; and (6) designing storage systems around denser devices that offer lower power per byte but reduced IOPS per terabyte and exploring trade-offs between capacity, performance, and energy.

Success will enable scalable, energy-efficient data management solutions that span edge-to-facility infrastructures, empower scientific discovery at exascale data volumes, and ensure that the growth of data is not limited by energy supply.

PRD5: Develop Integrated, Scalable Energy Measurement and Modeling Capabilities for Next-Generation Computing Systems

Key Questions: (a) How should simulation and modeling infrastructures enable accurate multilevel energy modeling across devices, architectures, and systems? (b) How can multiresolution measurement frameworks attribute energy usage across end-to-end hardware and software systems at scale? (c) Which approaches balance fidelity and simulation efficiency to support early-stage design exploration? (d) What strategies ensure reproducibility and sustainability as technologies evolve?

Future computing systems will increasingly feature heterogeneous devices, novel architectures, and complex workflows that span from edge to facility. In this environment, energy consumption must be a first-class design constraint and will require early, accurate, and scalable modeling of energy consumption from individual devices to entire systems. Unfortunately, current energy modeling tools are fragmented, outdated, and often proprietary, thereby hindering co-design efforts and limiting insights into potential energy savings.

This PRD calls for the development of unified, open-source energy modeling infrastructures that support multiscale simulation and real-system measurement. To reduce development time and encourage rapid, widespread adoption, this support should be added to existing, widely used open-source frameworks. These tools must integrate models at the device, architectural, and system levels while maintaining modularity and extensibility to accommodate new technologies such as non-CMOS accelerators, memory-rich chiplets, and photonic interconnects.

Key research areas include the following:

- Unified Modeling Frameworks: Design modular, interoperable interfaces for connecting device-, architecture-, and system-level energy models. These frameworks must support mixed-fidelity modeling and plug-and-play integration of new models while abstracting internal complexity.
- Scalable and Flexible Models: Develop energy models that balance fidelity and performance, thereby enabling both high-resolution simulation and fast, exploratory co-design. Models must adapt to emerging, heterogeneous technologies and be usable from prototyping to leadership-class system scales.
- Viable Development and Maintenance: Ensure the long-term viability of modeling tools through centralized stewardship, support for community standards, integration with CI/CD pipelines, and investment in research software engineers (RSEs). Align efforts with broader open science infrastructure initiatives.
- Operational Measurement and Adaptation: Create novel methodologies for runtime energy measurement and in-situ optimization on current and future systems. These tools must support workload-aware energy adaptation within real scientific campaigns and under dynamic grid or facility constraints.
- Prototype-Based Validation: Validate energy and performance models using testbeds and hardware prototypes running full scientific applications. Collaborate with CHIPS Act–funded prototyping efforts to ensure feedback between early-stage modeling and real-world behavior.

Together, these research directions will enable early-stage energy performance co-design, support modeling practices that are built to last, and provide the DOE and broader community with robust tools for shaping the future of energy-efficient computing.

5 Enablers for Research Progress

Achieving breakthroughs in energy-efficient computing requires not only research but also the infrastructure, tools, and policies that enable technical progress. This section outlines key enablers that can accelerate innovation and address systemic challenges identified by the workshop participants.

5.1 E1: Establish Prototyping Capabilities and Testbeds

The community should invest in both virtual and physical testbeds to support the evaluation of emerging energy-efficient computing technologies. Testbeds play a crucial role in validating new architectures, devices, and system software under realistic conditions. Moreover, they also provide an essential bridge between theoretical research and practical deployment, allowing researchers to benchmark novel approaches against existing technologies. This activity should

- facilitate co-design of hardware and software for both performance and energy consumption;
- support multidisciplinary collaboration, from microelectronics to system architecture;
- provide a platform for benchmarking, energy efficiency measurements, and software development on early-stage hardware;
- establish a continuum of technologies from the benchtop to multirack prototypes that cover a range of promising technologies that require exploration; and
- integrate prototype hardware with real scientific applications to validate ModSim capabilities (Section 3.4).

5.2 E2: Ensure Viable Stewardship of IP, EDA Software, and Modeling Tools

The community should act as a steward for key simulation, modeling, and EDA tools to prevent research stagnation and tool atrophy. Simulation and modeling tools often suffer from poor long-term maintenance as students graduate and researchers move on. To help mitigate this problem, the community can

- maintain key tools at national labs or in collaboration with academics to ensure continuity across generations;
- fund RSEs dedicated to long-term tool viability;
- require integration of funded tools into mainline repositories (e.g., gem5 [168, 169], SST [170], CODES [171]);
- establish site-wide licenses for EDA tools to expand accessibility for DOE researchers; and
- support teaching and learning of popular tools to expand the community that uses, develops, and improves these tools (e.g., by hosting and/or sponsoring bootcamps, tutorials, and workshops).

5.3 E3: Strengthen Co-Design Methodologies

The community should promote cross-disciplinary co-design to align hardware, software, and algorithms for energy efficiency. Energy-efficient computing requires deep collaboration across multiple layers of the computing stack. To enable this, the community should

- encourage early-stage collaboration between hardware architects, system software developers, and domain scientists;
- establish open-source design frameworks and tools that support hardware-software co-optimization; and
- develop co-design methodologies that remain adaptable as computing paradigms evolve.

5.4 E4: Develop Standardized Metrics for Energy Efficiency

The community should define standardized energy efficiency metrics across user, system, and facility levels. To evaluate energy-aware optimizations effectively, we need clear, consistent metrics that span multiple scales.

- **User-Level Metrics**: Measure useful work per unit of energy (e.g., AI convergence, scientific simulation accuracy).

- **System-Level Metrics**: Capture resource utilization per unit of energy to minimize waste.
- **Facility-Level Metrics**: Incorporate PUE and total lifecycle energy costs.

5.5 E5: Expand Instrumentation and Telemetry in Data Centers

The community should enhance telemetry and instrumentation for real-time energy monitoring in operating HPC facilities. Current systems lack fine-grained power measurement, making energy-aware optimizations difficult. The community should

- deploy granular power and performance monitoring across computing components (CPU, memory, storage, network);
- enable real-time access to power and energy telemetry for both users and system administrators;
- standardize APIs for integrating energy measurement tools into schedulers and runtime systems; and
- provide a source of open information on real-world energy use in data centers and HPC.

5.6 E6: Create a Technology Study Group for Technology Readiness Tracking

The community should establish a technology study group to track emerging hardware and software trends, similar to the DARPA Exascale Study from 2008. Scientific and economic forces shape computing technologies, but long-term energy-efficient computing strategies require foresight. A dedicated study group should

- monitor global R&D trends in microelectronics, memory, networking, and AI accelerators;
- develop roadmaps for emerging technologies to guide DOE investments; and
- identify risks and opportunities in upcoming paradigms to ensure DOE computing remains competitive.

6 Related Topics Discussed during our Workshop

While the workshop addressed a comprehensive and forward-looking agenda across hardware, software, systems, and applications, several topics were discussed but not easily included in one of our PRDs. These topics reflect the evolving landscape of energy-efficient computing and its intersection with environmental, social, and economic systems. Other reports have partially explored these topics (e.g., the MAPT roadmap [91] and the EES2 technical workshop goal).

Energy-Aware Security and Privacy. Security and privacy constraints may conflict with energy efficiency. Encryption, access controls, and data locality policies can increase data movement or computation. Moreover, secure computing paradigms such as federated learning or homomorphic encryption may have large energy footprints. Future efforts should investigate the trade-offs between security guarantees and energy consumption in distributed and heterogeneous systems.

Life-Cycle and Sustainability Analysis of Computing Systems. The workshop focused on operational energy efficiency but did not explicitly address the life-cycle energy costs of computing infrastructure. This includes the energy and environmental impact of fabricating chips, manufacturing systems, transporting components, and end-of-life considerations such as disposal and recycling. Life-cycle analysis should be integrated into modeling tools and procurement strategies to assess trade-offs between extending hardware lifetimes and deploying newer, more efficient systems.

Usability of Energy-Aware Systems. End users, developers, and operators must make decisions based on energy telemetry and optimization tools. The effectiveness of these tools depends on how actionable and understandable the energy information is. There was limited discussion of human-computer interaction, behavioral economics, or incentive structures that encourage energy-aware behavior. Future work should explore how to design usable energy-aware abstractions and interfaces that empower both expert and non-expert users.

Economic Models and Market Mechanisms. Although some discussion touched on grid integration and load balancing, the broader role of economic incentives in energy-efficient computing was not fully explored. Dynamic pricing models, time-of-use tariffs, or cost internalization could influence job scheduling, workload placement, and user behavior. Future workshops should consider collaboration with energy economists and policy experts to co-optimize scientific productivity and energy grid stability.

International Coordination and Standards. The global nature of semiconductor supply chains and scientific collaboration suggests the importance of international coordination. Opportunities include joint test beds, shared energy-efficient data center infrastructure, common standards for telemetry and modeling, and cooperative policy alignment. This is particularly relevant for distributed science workflows that span institutions and borders.

We recommend that the community consider these emerging areas as focal points for future community studies, roadmaps, or cross-agency collaborations. These topics can strengthen the foundation for performant, secure, and globally-relevant energy-efficient computing ecosystems.

7 Contributors

The following people contributed to the organization and writing of this report:

7.1 DOE Office of Advanced Scientific Computing Research

- Hal Finkel
- Marco Fornari
- David Rabson

7.2 Workshop Co-Chairs

- Robert B. Ross, Argonne National Laboratory
- Jeffrey S. Vetter, Oak Ridge National Laboratory

7.3 Authors

- Brad Aimone, Sandia National Laboratories
- George Amvrosiadis, Carnegie Mellon University
- James A. Ang, Pacific Northwest National Laboratory
- Brian Austin, Lawrence Berkeley National Laboratory
- Ryan Coffee, LCLS-SLAC National Accelerator Laboratory
- Paul D. Hovland, Argonne National Laboratory
- Hyesoon Kim, Georgia Tech
- Zhiling Lan, University of Illinois-Chicago and Argonne National Laboratory
- George Michelogiannakis, Lawrence Berkeley National Laboratory
- Kathryn Mohror, Lawrence Livermore National Laboratory
- Kevin Pedretti, Sandia National Laboratories
- John Shalf, Lawrence Berkeley National Laboratory
- Matthew D. Sinclair, University of Wisconsin-Madison
- Richard Vuduc, Georgia Tech
- Angel Yanguas-Gil, Argonne National Laboratory

Glossary

Advanced Materials and Manufacturing Technologies Office A DOE office mentioned in the context of the Energy Efficiency Scaling for Two Decades (EES2) report.

Benchmarking The process of evaluating the performance and/or energy efficiency of a computer system or algorithm, often using standardized tests or proxy applications.

Chiplet A small, modular semiconductor die that contains a subset of functionality, used to build a larger system-on-a-package through heterogeneous integration.

Co-design A methodology where multiple layers of the computing stack (e.g., hardware, software, and algorithms) are designed and optimized together, rather than sequentially, to achieve holistic system goals.

Data Movement The transfer of data between different components of a computing system (e.g., from storage to memory, or between cores and caches); often the largest source of energy consumption in modern systems.

Energy Complexity A theoretical measure for algorithms akin to *Big-O* notation, but which explicitly includes the energy cost of operations, especially data movement.

Heterogeneous Computing A system architecture that uses a diversity of specialized processing units (e.g., Central Processing Units (CPUs), Graphics Processing Units (GPUs), and Application-Specific Integrated Circuits (ASICs)) to efficiently handle different parts of a workload.

Monolithic Chip A traditional integrated circuit design where all components are fabricated onto a single, large die.

Neuromorphic Computing A computing paradigm that mimics the neural architecture and components (neurons, synapses) of the human brain, focusing on time- and energy-efficient processing.

Photonic Computing A computing paradigm that uses light (photons) instead of electrons for computation and communication, valued for high-speed and low-energy interconnects.

Proxy Application (Mini-app) A simplified, small-scale software application that captures the essential computational and communication characteristics of a full scientific workload, used primarily for co-design and benchmarking.

Resource Disaggregation The decoupling of compute, memory, and storage resources into independent pools that can be dynamically composed for a workload, often via high-speed interconnects like Compute Express Link (CXL).

SLAC National Accelerator Laboratory A Department of Energy (DOE) national laboratory managed by Stanford University.

Superconducting Computing A specific type of Cryogenic and Superconducting Computing (CSC) that utilizes superconducting materials (zero electrical resistance) to enable ultra-low-power logic and memory, often at ultra-low temperatures.

Telemetry The process of collecting and transmitting fine-grained data (e.g., power consumption, temperature, utilization) from remote sensors or system components for monitoring and optimization.

Waferscale Computing A system architecture that integrates an entire computational system onto a single silicon wafer, dramatically reducing chip-to-chip communication overhead.

Acronyms

A/D Analog-to-Digital.

AI Artificial Intelligence.

ASCR Advanced Scientific Computing Research.

ASIC Application-Specific Integrated Circuit.

AVX Advanced Vector Extensions.

BLAS Basic Linear Algebra Subroutines.

CI/CD Continuous Integration/Continuous Deployment.

CMOS Complementary Metal-Oxide-Semiconductor.

CNFET Carbon Nanotube FET.

CPU Central Processing Unit.

CSC Cryogenic and Superconducting Computing.

CXL Compute Express Link.

DARPA Defense Advanced Research Projects Agency.

DoD Department of Defense.

DOE Department of Energy.

DRAM Dynamic Random-Access Memory.

DSL Domain-Specific Language.

DVFS Dynamic Voltage and Frequency Scaling.

EDA Electronic Design Automation.

EES2 Energy Efficiency Scaling for Two Decades.

ERI Electronics Resurgence Initiative.

FFT Fast Fourier Transform.

FFTW Fastest Fourier Transform in the West.

FLOP Floating-Point Operation.

FPGA Field-Programmable Gate Array.

GPGPU General-Purpose Graphics Processing Unit.

GPU Graphics Processing Unit.

HAMR Heat-Assisted Magnetic Recording.

HPC High-Performance Computing.

I/O Input/Output.

IOPS I/O Operations Per Second.

IoT Internet of Things.

IP Intellectual Property.

IR Intermediate Representation.

JJ Josephson Junction.

LCLS Linac Coherent Light Source.

LLVM Low-Level Virtual Machine.

MAPT Microelectronics and Advanced Packaging Technologies.

MD Molecular Dynamics.

ML Machine Learning.

MLIR Multi-Level Intermediate Representation.

ModSim Modeling and Simulation.

MPI Message Passing Interface.

NNSA National Nuclear Security Administration.

NVMe Non-Volatile Memory express.

ORNL Oak Ridge National Laboratory.

PDK Process Design Kit.

PIM Processing-In-Memory.

PLC Penta-Level Cell.

PRD Priority Research Direction.

PUE Power Usage Effectiveness.

R&D Research and Development.

RSE Research Software Engineer.

SC Office of Science.

SDE Stochastic Differential Equation.

SiP System in Package.

SoC System-on-a-Chip.

SRAM Static Random-Access Memory.

SRC Semiconductor Research Corporation.

TLC Triple-Level Cell.

TPU Tensor Processing Unit.

UCIe Universal Chiplet Interconnect Express.

UQ Uncertainty Quantification.

References

[1] Semiconductor Research Corporation. Decadal plan for semiconductors. Technical report, Semiconductor Research Corporation, 2021.

[2] D. Mytton and M. Ashtine. Sources of data center energy estimates: A comprehensive review. *Joule*, 6(9):2032–2056, 2022. DOI: 10.1016/j.joule.2022.07.011.

[3] Arman Shehabi, Alex Hubbard, Alex Newkirk, Nuoa Lei, Md Abu Bakkar Siddik, Billie Holecek, Jonathan Koomey, Eric Masanet, and Dale Sartor. 2024 United States Data Center Energy Usage Report. Technical Report LBNL-2001637, Lawrence Berkeley National Laboratory, Berkeley, California, December 2024.

[4] Gordon E. Moore. Cramming More Components onto Integrated Circuits. *Electronics*, 38(8):114–117, Apr 1965.

[5] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc. Design of ion-implanted MOSFET's with very small physical dimensions. *IEEE Journal of Solid-State Circuits*, 9(5):256–268, 1974. DOI: 10.1109/jssc.1974.1050511.

[6] D. J. Frank, R. H. Dennard, E. Nowak, P. M. Solomon, Y. Taur, and W. Hon-Sum Philip. Device scaling limits of si MOSFETs and their application dependencies. *Proceedings of the IEEE*, 89(3):259–288, 2001. DOI: 10.1109/5.915374.

[7] M. Bohr. A 30 year retrospective on dennard's MOSFET scaling paper. *IEEE Solid-State Circuits Society Newsletter*, 12(1):11–13, 2007. DOI: 10.1109/N-SSC.2007.4785534.

[8] Mark A Christon, David A Crawford, Eugene S Hertel, James S Peery, and Allen C Robinson. Asci red–experiences and lessons learned with a massively parallel teraflop supercomputer. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 1997.

[9] Paul Henning and Andrew B. White. Trailblazing with roadrunner. *Computing in Science & Engineering*, 11(4):91–95, 2009. DOI: 10.1109/MCSE.2009.130.

[10] C. E. Leiserson, N. C. Thompson, J. S. Emer, B. C. Kuszmaul, B. W. Lampson, D. Sanchez, and T. B. Schardl. There's plenty of room at the top: What will drive computer performance after moore's law? *Science*, 368(6495):eaam9744, 2020. DOI: 10.1126/science.aam9744.

[11] C. Murray, S. Guha, D. Reed, G. Herrera, K. Kleese van Dam, S. Salahuddin, J. Ang, T. Conte, D. Jena, R. Kaplar, H. Atwater, R. Stevens, D. Boroyevich, W. Chappell, T.-J. K. Liu, J. Rattner, M. Witherell, K. Afridi, S. Ang, J. Bock, S. Chowdhury, S. Datta, K. Evans, J. Flicker, M. Hollis, N. Johnson, K. Jones, P. Kogge, S. Krishnamoorthy, M. Marinella, T. Monson, S. Narumanchi, P. Ohodnicki, R. Ramesh, M. Schuette, J. Shalf, S. Shahedipour-Sandvik, J. Simmons, V. Taylor, T. Theis, E. Colby, R. Pino, A. Schwartz, K. Runkles, J. Harmon, M. Nelson, and V. Skonicki. Basic research needs for microelectronics: Report of the office of science workshop on basic research needs for microelectronics, october 23 – 25, 2018. Technical report, ; USDOE Office of Science (SC) (United States), 2018. DOI: https://doi.org/10.2172/1616249.

[12] J. S. Vetter, R. Brightwell, M. Gokhale, P. McCormick, R. Ross, J. Shalf, K. Antypas, D. Donofrio, T. Humble, C. Schuman, B. Van Essen, S. Yoo, A. Aiken, D. Bernholdt, S. Byna, K. Cameron, F. Cappello, B. Chapman, A. Chien, M. Hall, R. Hartman-Baker, Z. Lan, M. Lang, J. Leidel, S. Li, R. Lucas, J. Mellor-Crummey, P. Peltz Jr., T. Peterka, M. Strout, and J. Wilke. Extreme heterogeneity 2018 - productive computational science in the era of extreme heterogeneity: Report for DOE ASCR workshop on extreme heterogeneity. Technical report, USDOE Office of Science (SC) (United States), 2018. DOI: 10.2172/1473756.

[13] James Ang, Andrew A. Chien, Simon David Hammond, Adolfy Hoisie, Ian Karlin, Scott Pakin, John Shalf, and Jeffrey S. Vetter. Reimagining codesign for advanced scientific computing: Report for the ascr workshop on reimagining codesign. Technical report, USDOE Office of Science (SC) (United States), 4 2022. URL: https://www.osti.gov/biblio/1822199, DOI: 10.2172/1822199.

[14] Rick Stevens, Valerie Taylor, Jeff Nichols, Arthur Barney Maccabe, Katherine Yelick, and David Brown. AI for Science: Report on the Department of Energy (DOE) Town Halls on Artificial Intelligence (AI) for Science. Technical report, Argonne National Lab. (ANL), Argonne, IL (United States), 02 2020. URL: https://www.osti.gov/biblio/1604756, DOI: 10.2172/1604756.

[15] A. Almgren, P. DeMar, J. Vetter, K. Riley, K. Antypas, D. Bard, R. Coffey, E. Dart, S. Dosanjh, R. Gerber, J. Hack, I. Monga, M. E. Papka, L. Rotman, T. Straatsma, J. Wells, D. E. Bernholdt, W. Bethel, G. Bosilca, F. Cappello, T. Gamblin, S. Habib, J. Hill, J. K. Hollingsworth, L. C. McInnes, K. Mohror, S. Moore, K. Moreland, R. Roser, S. Shende, G. Shipman, and S. Williams. Advanced scientific computing research exascale requirements review. an office of science review sponsored by advanced scientific computing research, september 27-29, 2016, rockville, maryland. Technical report, ; Argonne National Lab. (ANL), Argonne, IL (United States). Argonne Leadership Computing Facility, 2017. DOI: 10.2172/1375638.

[16] Deloitte Center for Energy and Industrials. Can US infrastructure keep up with the AI economy? AI infrastructure gaps. Deloitte Insights, December 2025. Eight hyperscalers expect a 44% year-over-year increase to US$371 billion in 2025 for AI data centers and computing resources; hyperscalers surpassed capital-intensive utilities in capex during 2024. URL: https://www.deloitte.com/us/en/insights/industry/power-and-utilities/data-center-infrastructure-artificial-intelligence.html.

[17] IEEE ComSoc Technology Blog. AI spending boom accelerates: Big tech to invest an aggregate of $400 billion in 2025; much more in 2026. *IEEE ComSoc Technology Blog*, October 2025. Google, Meta, Microsoft and Amazon have together spent $112 billion on capital expenditures in the past three months alone. URL: https://techblog.comsoc.org/2025/11/01/ai-spending-boom-accelerates-big-tech-to-invest-invest-an-aggregate-of-400-billion-in-2025.

[18] Trace Cohen. In Q3 2025, Amazon, Meta, Alphabet, and Microsoft invested $112 billion in AI infrastructure. LinkedIn, November 2025. Amazon: Forecasting $125B in 2025 CapEx; Alphabet: Expects $91–93B; Meta: $70–72B; Microsoft: trajectory toward ~$94B. URL: https://www.linkedin.com/posts/tracecohen_in-q3-2025-amazon-meta-alphabet-and-microsoft-activity-7391957387915730944-pUKg.

[19] J. Nickolls and W. J. Dally. The GPU computing era. *IEEE Micro*, 30(2):56–69, 2010. DOI: 10.1109/mm.2010.41.

[20] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-Datacenter Performance Analysis of a Tensor Processing Unit. In *Proceedings of the 44th Annual International Symposium on Computer Architecture*, ISCA, pages 1–12, New York, NY, USA, 2017. ACM. URL: http://doi.acm.org/10.1145/3079856.3080246, DOI: 10.1145/3079856.3080246.

[21] Jordan Aljbour, Tom Wilson, and Poorvi Patel. Powering Intelligence: Analyzing Artificial Intelligence and Data Center Energy Consumption. Technical report, Electric Power Research Institute, 2024. URL: https://www.epri.com/research/products/3002028905.

[22] J. L. Hennessy and D. A. Patterson. A new golden age for computer architecture. *Commun. ACM*, 62(2):48–60, 2019. DOI: 10.1145/3282307.

[23] G. H. Loh, M. J. Schulte, M. Ignatowski, V. Adhinarayanan, S. Aga, D. Aguren, V. Agrawal, A. M. Aji, J. Alsop, P. Bauman, B. M. Beckmann, M. V. Beigi, S. Blagodurov, T. Boraten, M. Boyer, W. C. Brantley, N. Chalmers, S. Chen, K. Cheng, M. L. Chu, D. Cownie, N. Curtis, J. D. Pino, N. Duong, A. Duțu, Y. Eckert, C. Erb, C. Freitag, J. L. Greathouse, S. Gurumurthi, A. Gutierrez, K. Hamidouche, S. Hossamani, W. Huang, M. Islam, N. Jayasena, J. Kalamatianos, O. Kayiran, J. Kotra, A. Lee, D. Lowell, N. Madan, A. Majumdar, N. Malaya, S. Manne, S. Mashimo, D. McDougall, E. Mednick, M. Mishkin, M. Nutter, I. Paul, M. Poremba, B. Potter, K. Punniyamurthy, S. Puthoor, S. E. Raasch, K. Rao, G. Rodgers, M. Scrbak, M. Seyedzadeh, J. Slice, V. Sridharan, R.v. Oostrum, E.v. Tassell, A. Vishnu, S. Wasmundt, M. Wilkening, N. Wolfe, M. Wyse, A. Yalavarti, and D. Yudanov. A research retrospective on AMD's exascale computing journey. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, page Article 81, Orlando, FL, USA, 2023. Association for Computing Machinery. DOI: 10.1145/3579371.3589349.

[24] P. M. Kogge. *The Zen of Exotic Computing*. Society for Industrial and Applied Mathematics, 2022. DOI: 10.1137/1.9781611977295.

[25] W. Chen and B. Bottoms. Heterogeneous integration roadmap: Driving force and enabling technology for systems of the future. In *2019 Symposium on VLSI Technology*, pages T50–T51, 2019. DOI: 10.23919/VLSIT.2019.8776484.

[26] International Technology Roadmap for Semiconductors. International technology roadmap for semiconductors report. Technical report, Semiconductor Research Corporation, 2013.

[27] Sean Lie. Cerebras Architecture Deep Dive: First Look Inside the Hardware/Software Co-Design for Deep Learning. *IEEE Micro*, 43(3):18–30, 2023. DOI: 10.1109/MM.2023.3256384.

[28] H. Ltaief, Y. Hong, L. Wilson, M. Jacquelin, M. Ravasi, and D. E. Keyes. Scaling the "memory wall" for multi-dimensional seismic processing with algebraic compression on cerebras CS-2 systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page Article 6, Denver, CO, USA, 2023. Association for Computing Machinery. DOI: 10.1145/3581784.3627042.

[29] J. Broz, M. Byrd, Y. Chembo, B. de Jong, E. Figueroa, T. Humble, J. Larson, P. Lougovski, O. Parekh, G. Quiroz, and K. Svore. Basic research needs in quantum computing and networking. Technical report, US Department of Energy (USDOE), Washington, DC (United States), 2023. DOI: 10.2172/2001044.

[30] R. Absar, H. Elgabra, D. Ma, Y. Zhao, and L. Wei. Cryogenic CMOS for quantum computing. In Weiqiang Liu, Jie Han, and Fabrizio Lombardi, editors, *Design and Applications of Emerging Computer Systems*, pages 591–621. Springer Nature Switzerland, Cham, 2024. DOI: 10.1007/978-3-031-42478-6_22.

[31] M. A. Manheimer. Cryogenic computing complexity program: Phase 1 introduction. *IEEE Transactions on Applied Superconductivity*, 25(3):1–4, 2015. DOI: 10.1109/TASC.2015.2399866.

[32] D. S. Holmes, A. L. Ripple, and M. A. Manheimer. Energy-efficient superconducting computing—power budgets and requirements. *IEEE Transactions on Applied Superconductivity*, 23(3):1701610–1701610, 2013. DOI: 10.1109/TASC.2013.2244634.

[33] Sapan Agarwal, Alexander Hsia, Robin Jacobs-Gedrim, David R Hughart, Steven J Plimpton, Conrad D James, and Matthew J Marinella. Designing an analog crossbar based neuromorphic accelerator. In *2017 Fifth Berkeley Symposium on Energy Efficient Electronic Systems & Steep Transistors Workshop (E3S)*, pages 1–3. IEEE, 2017.

[34] John Von Neumann and Ray Kurzweil. *The computer and the brain*. Yale university press, 2012.

[35] Carver Mead. Neuromorphic electronic systems. *Proceedings of the IEEE*, 78(10):1629–1636, 1990.

[36] Dhireesha Kudithipudi, Catherine Schuman, Craig M Vineyard, Tej Pandit, Cory Merkel, Rajkumar Kubendran, James B Aimone, Garrick Orchard, Christian Mayr, Ryad Benosman, et al. Neuromorphic computing at scale. *Nature*, 637(8047):801–812, 2025.

[37] Bhavin J Shastri, Alexander N Tait, Thomas Ferreira de Lima, Wolfram HP Pernice, Harish Bhaskaran, C David Wright, and Paul R Prucnal. Photonics for Artificial Intelligence and Neuromorphic Computing. *Nature Photonics*, 15(2):102–114, 2021.

[38] C. Huang, B. Shastri, and P. Pruncal. Photonic computing: an introduction. In Harish Bhaskaran and Wolfram H. P. Pernice, editors, *Phase Change Materials-Based Photonic Computing*, pages 37–65. Elsevier, 2024. DOI: https://doi.org/10.1016/B978-0-12-823491-4.00003-5.

[39] Z. Wu, L. Yuan Dai, Y. Wang, S. Wang, and K. Bergman. Flexible silicon photonic architecture for accelerating distributed deep learning. *Journal of Optical Communications and Networking*, 16(2):A157–A168, 2024. DOI: 10.1364/JOCN.497372.

[40] L. N. Chakrapani, P. Korkmaz, B. E. S. Akgul, and K. V. Palem. Probabilistic system-on-a-chip architectures. *ACM Transactions on Design Automation of Electronic Systems*, 12(3):29–es, 2007. DOI: 10.1145/1255456.1255466.

[41] Shashank Misra, Leslie C Bland, Suma G Cardwell, Jean Anne C Incorvia, Conrad D James, Andrew D Kent, Catherine D Schuman, J Darby Smith, and James B Aimone. Probabilistic neural computing with stochastic devices. *Advanced Materials*, 35(37):2204569, 2023.

[42] George M Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in dna. *Science*, 337(6102):1628–1628, 2012.

[43] James Bornholt, Randolph Lopez, Douglas M Carmean, Luis Ceze, Georg Seelig, and Karin Strauss. A dna-based archival storage system. *ACM SIGOPS Operating Systems Review*, 50(2):637–649, 2016.

[44] Lee Organick, Shinae Doong Ang, Yuan-Jyue Chen, et al. Random access in large-scale dna data storage. *Nature Biotechnology*, 36(3):242–248, 2018.

[45] Luis Ceze, Jeff Nivala, and Karin Strauss. Dna-based storage for the big data era. *Nature Reviews Genetics*, 20(8):456–466, 2019.

[46] Leonard M Adleman. Molecular computation of solutions to combinatorial problems. *Science*, 266(5187):1021–1024, 1994.

[47] L. Grigori, J. W. Demmel, and X. Hua. Communication avoiding gaussian elimination. In *High Performance Computing, Networking, Storage and Analysis, 2008. SC 2008. International Conference for*, pages 1–12, 2008. DOI: 10.1109/sc.2008.5214287.

[48] Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, Tara Vazir Ghaemmaghami, Rajendra Gottipati, William Gulland, Robert Hagmann, C. Richard Ho, Doug Hogberg, John Hu, Robert Hundt, Dan Hurt, Julian Ibarz, Aaron Jaffey, Alek Jaworski, Alexander Kaplan, Harshit Khaitan, Daniel Killebrew, Andy Koch, Naveen Kumar, Steve Lacy, James Laudon, James Law, Diemthu Le, Chris Leary, Zhuyuan Liu, Kyle Lucke, Alan Lundin, Gordon MacKean, Adriana Maggiore, Maire Mahony, Kieran Miller, Rahul Nagarajan, Ravi Narayanaswami, Ray Ni, Kathy Nix, Thomas Norrie, Mark Omernick, Narayana Penukonda, Andy Phelps, Jonathan Ross, Matt Ross, Amir Salek, Emad Samadiani, Chris Severn, Gregory Sizikov, Matthew Snelham, Jed Souter, Dan Steinberg, Andy Swing, Mercedes Tan, Gregory Thorson, Bo Tian, Horia Toma, Erick Tuttle, Vijay Vasudevan, Richard Walter, Walter Wang, Eric Wilcox, and Doe Hyun Yoon. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Comput. Archit. News*, 45(2):1–12, June 2017. DOI: 10.1145/3140659.3080246.

[49] Thomas Norrie, Nishant Patil, Doe Hyun Yoon, George Kurian, Sheng Li, James Laudon, Cliff Young, Norman Jouppi, and David Patterson. The Design Process for Google's Training Chips: TPUv2 and TPUv3. *IEEE Micro*, 41(2):56–63, 2021. DOI: 10.1109/MM.2021.3058217.

[50] Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. TPU v4: An Optically Reconfigurable Supercomputer for Machine Learning with Hardware Support for Embeddings. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, ISCA, New York, NY, USA, 2023. Association for Computing Machinery. DOI: 10.1145/3579371.3589350.

[51] Norman P. Jouppi, Doe Hyun Yoon, Matthew Ashcraft, Mark Gottscho, Thomas B. Jablin, George Kurian, James Laudon, Sheng Li, Peter Ma, Xiaoyu Ma, Thomas Norrie, Nishant Patil, Sushma Prasad, Cliff Young, Zongwei Zhou, and David Patterson. Ten lessons from three generations shaped google's tpuv4i. In *Proceedings of the 48th Annual International Symposium on Computer Architecture*, ISCA, page 1–14, Piscataway, NJ, USA, 2021. IEEE Press. DOI: 10.1109/ISCA52012.2021.00010.

[52] D. E. Shaw, M. M. Deneroff, R. O. Dror, J. S. Kuskin, R. H. Larson, J. K. Salmon, C. Young, B. Batson, K. J. Bowers, and J. C. Chao. Anton, a special-purpose machine for molecular dynamics simulation. In *34th Annual International Conference on Computer Architecture*, pages 1–12, 2007. DOI: 10.1080/08927029908022078.

[53] J. Ragan-Kelley, A. Adams, D. Sharlet, C. Barnes, S. Paris, M. Levoy, S. Amarasinghe, and F. Durand. Halide: decoupling algorithms from schedules for high-performance image processing. *Commun. ACM*, 61(1):106–115, 2017. DOI: 10.1145/3150211.

[54] Luiz André Barroso and Urs Hölzle. The case for energy-proportional computing. *Computer*, 40(12):33–37, 2007. DOI: 10.1109/MC.2007.443.

[55] Amuthan A Ramabathiran and Prabhu Ramachandran. SPINN: sparse, physics-based, and partially interpretable neural networks for PDEs. *Journal of Computational Physics*, 445:110600, 2021. DOI: 10.1016/j.jcp.2021.110600.

[56] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022. DOI: 10.1007/s10915-022-01939-z.

[57] Lorenz Kummer, Kevin Sidak, Tabea Reichmann, and Wilfried Gansterer. Adaptive precision training (AdaPT): A dynamic quantized training approach for DNNs. In *Proceedings of the 2023 SIAM International Conference on Data Mining (SDM)*, pages 559–567. SIAM, 2023. DOI: 10.1137/1.9781611977653.ch63.

[58] Jack Dongarra, Laura Grigori, and Nicholas J Higham. Numerical algorithms for high-performance computational science. *A Philosophical Transactions of the Royal Society*, 378(2166):20190066, 2020. DOI: 10.1098/rsta.2019.0066.

[59] Aydin Buluc, Tamara G. Kolda, Stefan M. Wild, Mihai Anitescu, Anthony Degennaro, John D. Jakeman, Chandrika Kamath, Ramakrishnan Ramki Kannan, Miles E. Lopes, Per-Gunnar Martinsson, et al. Randomized Algorithms for Scientific Computing (RASC). Technical report, Oak Ridge National Laboratory, 07 2021. URL: https://www.osti.gov/biblio/1807223, DOI: 10.2172/1807223.

[60] John R. Hu, Louis Liu, Shuhan Liu, Boonkhim Liew, David Guan, James Chen, Steven Jones, and William J. Dally. Co-optimization of gpu ai chip from technology, design, system and algorithms. In *2024 IEEE International Electron Devices Meeting (IEDM)*, pages 1–4, 2024. DOI: 10.1109/IEDM50854.2024.10873439.

[61] Grant Wilkins, Sheng Di, Jon C Calhoun, Robert Underwood, and Franck Cappello. To compress or not to compress: Energy trade-offs and benefits of lossy compressed I/O. In *2025 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 861–873. IEEE, 2025. DOI: 10.1109/IPDPS64566.2025.00082.

[62] Maboud F Kaloorazi, Kai Liu, Jie Chen, Rodrigo C De Lamare, and Susanto Rahardja. Randomized rank-revealing QLP for low-rank matrix decomposition. *IEEE Access*, 11:63650–63666, 2023. DOI: 10.1109/ACCESS.2023.3288889.

[63] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021. DOI: 10.1038/s42254-021-00314-5.

[64] James B Aimone, Prasanna Date, Gabriel A Fonseca-Guerra, Kathleen E Hamilton, Kyle Henke, Bill Kay, Garrett T Kenyon, Shruti R Kulkarni, Susan M Mniszewski, Maryam Parsa, et al. A review of non-cognitive applications for neuromorphic computing. *Neuromorphic Computing and Engineering*, 2(3):032003, 2022. DOI: 10.1088/2634-4386/ac889c.

[65] Serge Gratton and Ph L Toint. A note on solving nonlinear optimization problems in variable precision. *Computational Optimization and Applications*, 76:917–933, 2020. DOI: 10.1007/s10589-020-00190-2.

[66] Sven Leyffer, Stefan M Wild, Mike Fagan, Marc Snir, Krishna Palem, Kazutomo Yoshii, and Hal Finkel. Doing Moore with less–leapfrogging Moore's law with inexactness for supercomputing. *arXiv preprint arXiv:1610.02606*, 2016.

[67] Richard J. Clancy, Matt Menickelly, Jan Hückelheim, Paul Hovland, Prani Nalluri, and Rebecca Gjini. Trophy: Trust region optimization using a precision hierarchy. In Derek Groen, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot, editors, *Computational Science – ICCS 2022*, pages 445–459, Cham, 2022. Springer International Publishing. DOI: 10.1007/978-3-031-08751-6_32.

[68] Alfredo Buttari, Jack Dongarra, Julie Langou, Julien Langou, Piotr Luszczek, and Jakub Kurzak. Mixed precision iterative refinement techniques for the solution of dense linear systems. *The International Journal of High Performance Computing Applications*, 21(4):457–466, 2007. DOI: 10.1177/1094342007084026.

[69] Erin Carson and Nicholas J Higham. A new analysis of iterative refinement and its application to accurate solution of ill-conditioned sparse linear systems. *SIAM Journal on Scientific Computing*, 39(6):A2834–A2856, 2017. DOI: 10.1137/17M1122918.

[70] Erin Carson and Nicholas J Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM Journal on Scientific Computing*, 40(2):A817–A847, 2018. DOI: 10.1137/17M1140819.

[71] Erin Carson, Nicholas J Higham, and Srikara Pranesh. Three-precision GMRES-based iterative refinement for least squares problems. *SIAM Journal on Scientific Computing*, 42(6):A4063–A4083, 2020. DOI: 10.1137/20M1316822.

[72] Patrick Amestoy, Alfredo Buttari, Nicholas J Higham, Jean-Yves L'excellent, Théo Mary, and Bastien Vieublé. Five-precision GMRES-based iterative refinement. *SIAM Journal on Matrix Analysis and Applications*, 45(1):529–552, 2024. DOI: 10.1137/23M1549079.

[73] J Darby Smith, Aaron J Hill, Leah E Reeder, Brian C Franke, Richard B Lehoucq, Ojas Parekh, William Severa, and James B Aimone. Neuromorphic scaling advantages for energy-efficient random walk computations. *Nature Electronics*, 5(2):102–112, 2022. DOI: 10.1038/s41928-021-00705-7.

[74] Herbert Jaeger, Beatriz Noheda, and Wilfred G Van Der Wiel. Toward a formal theory for computing machines made out of whatever physics offers. *Nature communications*, 14(1):4911, 2023. DOI: 10.1038/s41467-023-40533-1.

[75] Herbert Jaeger. Towards a generalized theory comprising digital, neuromorphic and unconventional computing. *Neuromorphic Computing and Engineering*, 1(1):012002, 2021. DOI: 10.1088/2634-4386/abf151.

[76] Catherine D Schuman, Shruti R Kulkarni, Maryam Parsa, J Parker Mitchell, Prasanna Date, and Bill Kay. Opportunities for neuromorphic computing algorithms and applications. *Nature Computational Science*, 2(1):10–19, 2022. DOI: 10.1038/s43588-021-00184-y.

[77] USDOE Office of Energy Efficiency and Renewable Energy (EERE). Workshop on manufacturing and integration challenges for analog and neuromorphic computing. Technical report, USDOE Office of Energy Efficiency and Renewable Energy (EERE), Washington DC (United States). Advanced Materials & Manufacturing Office (AMMTO), 08 2022. URL: https://www.osti.gov/biblio/1884395, DOI: 10.2172/1884395.

[78] Bishnu Patra, Rosario M. Incandela, Jeroen P. G. van Dijk, Harald A. R. Homulle, Lin Song, Mina Shahmohammadi, Robert Bogdan Staszewski, Andrei Vladimirescu, Masoud Babaie, Fabio Sebastiano, and Edoardo Charbon. Cryo-CMOS circuits and systems for quantum computing applications. *IEEE Journal of Solid-State Circuits*, 53(1):309–321, 2018. DOI: 10.1109/JSSC.2017.2737549.

[79] K.K. Likharev and V.K. Semenov. Rsfq logic/memory family: a new josephson-junction technology for subterahertz-clock-frequency digital systems. *IEEE Transactions on Applied Superconductivity*, 1(1):3–28, 1991. DOI: 10.1109/77.80745.

[80] Teng Wang, Yanlan Hu, Peng Fu, and Huajun Liu. Quench detection method for superconducting magnets with a phase difference measurement system based on multiple-correlation. *Fusion Engineering and De-*

sign, 170:112658, 2021. URL: https://www.sciencedirect.com/science/article/pii/S0920379621004348, DOI: https://doi.org/10.1016/j.fusengdes.2021.112658.

[81] Fabio Sebastiano, Harald Homulle, Bishnu Patra, Rosario Incandela, Jeroen van Dijk, Lin Song, Masoud Babaie, Andrei Vladimirescu, and Edoardo Charbon. Cryo-CMOS electronic control for scalable quantum computing. In *2017 54th ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6, 2017. DOI: 10.1145/3061639.3072948.

[82] George Michelogiannakis, Yehia Arafa, Brandon Cook, Liang Yuan Dai, Abdel-Hameed Hameed Badawy, Madeleine Glick, Yuyang Wang, Keren Bergman, and John Shalf. Efficient intra-rack resource disaggregation for hpc using co-packaged dwdm photonics. In *2023 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 158–172, 2023. DOI: 10.1109/CLUSTER52292.2023.00021.

[83] Ahmad Hassan, Sreenil Saha, and Anthony Chan Carusone. Fully integrated photonic dot-product engine in 45-nm soi cmos for photonic computing. In *2023 IEEE Silicon Photonics Conference (SiPhotonics)*, pages 1–2, 2023. DOI: 10.1109/SiPhotonics55903.2023.10141931.

[84] Chengeng Li, Fan Jiang, Shixi Chen, Xianbin Li, Jiaqi Liu, Wei Zhang, and Jiang Xu. Towards scalable gpu system with silicon photonic chiplet. In *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6, 2024. DOI: 10.23919/DATE58400.2024.10546733.

[85] IEEE. International Roadmap for Devices and Systems. *IEEE IRDS*, 2022. DOI: 10.60627/c13z-v363.

[86] Kanika Monga, Nitin Chaturvedi, and S. Gurunarayanan. Design of a stt-mtj based random-access memory with in-situ processing for data-intensive applications. *IEEE Transactions on Nanotechnology*, 21:455–465, 2022. DOI: 10.1109/TNANO.2022.3199230.

[87] Yao-Hung Huang, Yu-Cheng Hsieh, Yu-Cheng Lin, Yue-Der Chih, Eric Wang, Jonathan Chang, Ya-Chin King, and Chrong Jung Lin. High density embedded 3d stackable via rram in advanced mcu applications. In *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*, pages 1–2, 2023. DOI: 10.23919/VLSITechnologyandCir57934.2023.10185230.

[88] Michèle Weiland, Holger Brunst, Tiago Quintino, Nick Johnson, Olivier Iffrig, Simon Smart, Christian Herold, Antonino Bonanni, Adrian Jackson, and Mark Parsons. An Early Evaluation of Intel's Optane DC Persistent Memory Module and its Impact on High-performance Scientific Applications. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3295500.3356159.

[89] Yinjun Wu, Kwanghyun Park, Rathijit Sen, Brian Kroth, and Jaeyoung Do. Lessons Learned from the Early Performance Evaluation of Intel Optane DC Persistent Memory in DBMS. In *Proceedings of the 16th International Workshop on Data Management on New Hardware*, DaMoN, New York, NY, USA, 2020. Association for Computing Machinery. DOI: 10.1145/3399666.3399898.

[90] Pantea Zardoshti, Michael Spear, Aida Vosoughi, and Garret Swart. Understanding and Improving Persistent Transactions on Optane DC Memory. In *IEEE International Parallel and Distributed Processing Symposium*, IPDPS, pages 348–357, 2020. DOI: 10.1109/IPDPS47924.2020.00044.

[91] Semiconductor Research Corporation. Microelectronics and advanced packaging technologies roadmap. Technical report, Semiconductor Research Corporation, 2023.

[92] Debendra Das Sharma. Compute express link (cxl): Enabling heterogeneous data-centric computing with heterogeneous memory hierarchy. *IEEE Micro*, 43(2):99–109, 2023. DOI: 10.1109/MM.2022.3228561.

[93] Peter Onufryk and Swadesh Choudhary. UCIe: Standard for an Open Chiplet Ecosystem. *IEEE Micro*, 45(1):16–25, sep 2025. DOI: 10.1109/MM.2024.3451532.

[94] Hanchen Wang, Tianfan Fu, Yuanqi Du, Wenhao Gao, Kexin Huang, Ziming Liu, Payal Chandak, Shengchao Liu, Peter Van Katwyk, Andreea Deac, Anima Anandkumar, Karianne Bergen, Carla P. Gomes, Shirley Ho, Pushmeet Kohli, Joan Lasenby, Jure Leskovec, Tie-Yan Liu, Arjun Manrai, Debora Marks, Bharath Ramsundar,

Le Song, Jimeng Sun, Jian Tang, Petar Veličković, Max Welling, Linfeng Zhang, Connor W. Coley, Yoshua Bengio, and Marinka Zitnik. Scientific discovery in the age of artificial intelligence. *Nature*, 620(7972):47–60, Aug 2023. DOI: 10.1038/s41586-023-06221-2.

[95] Sara McAllister, Fiodar Kazhamiaka, Daniel S. Berger, Rodrigo Fonseca, Kali Frost, Aaron Ogus, Maneesh Sah, Ricardo Bianchini, George Amvrosiadis, Nathan Beckmann, and Gregory R. Ganger. A Call for Research on Storage Emissions. In *Proceedings of the 3rd Workshop on Sustainable Computer Systems*. ACM, July 2024. DOI: 10.1145/3727200.3727211.

[96] Suren Byna, Stratos Idreos, Terry Jones, Kathryn Mohror, Rob Ross, and Florin Rusu. Report for the ASCR Workshop on the Management and Storage of Scientific Data. Technical report, US Department of Energy (USDOE), Washington, DC (United States). Office of Science, 01 2022. URL: https://www.osti.gov/biblio/1845707, DOI: 10.2172/1845707.

[97] Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. Measuring the carbon intensity of ai in cloud instances. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, page 1877–1894, New York, NY, USA, 2022. Association for Computing Machinery. DOI: 10.1145/3531146.3533234.

[98] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training, 2021. URL: https://arxiv.org/abs/2104.10350, arXiv:2104.10350.

[99] Marion Dörrich, Mingcheng Fan, and Andreas M. Kist. Impact of Mixed Precision Techniques on Training and Inference Efficiency of Deep Neural Networks. *IEEE Access*, 11:57627–57634, 2023. DOI: 10.1109/ACCESS.2023.3284388.

[100] John Shalf. The future of computing beyond Moore's law. *A Philosophical Transactions of the Royal Society*, 2020. DOI: https://doi.org/10.1098/rsta.2019.0061.

[101] Peter Kogge and John Shalf. Exascale computing trends: Adjusting to the "new normal"' for computer architecture. *Computing in Science & Engineering*, 15(6):16–26, 2013. DOI: 10.1109/MCSE.2013.95.

[102] Matias Bjørling, Abutalib Aghayev, Hans Holmberg, Aravind Ramesh, Damien Le Moal, Gregory R. Ganger, and George Amvrosiadis. ZNS: Avoiding the block interface tax for flash-based SSDs. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 689–703. USENIX Association, July 2021. URL: https://www.usenix.org/conference/atc21/presentation/bjorling.

[103] Bill Martin, Yoni Shternhell, Mike James, Yeong-Jae Woo, Hyunmo Kang, Anu Murthy, Erich Haratsch, Kwok Kong, Andres Baez, Santosh Kumar, and et al. NVM Express Technical Proposal 4146 Flexible Data Placement, Nov 2022.

[104] Rekha Pitchumani and Yang-Suk Kee. Hybrid data reliability for emerging Key-Value storage devices. In *18th USENIX Conference on File and Storage Technologies (FAST 20)*, pages 309–322, Santa Clara, CA, February 2020. USENIX Association. URL: https://www.usenix.org/conference/fast20/presentation/pitchumani.

[105] Ethernet. Ieee p802.3df 200gb/s, 400gb/s, 800gb/s, and 1.6tb/s ethernet task force. URL: https://www.ieee802.org/3/df/index.html.

[106] Nik Sultana, John Sonchack, Hans Giesen, Isaac Pedisich, Zhaoyang Han, Nishanth Shyamkumar, Shivani Burad, André DeHon, and Boon Thau Loo. Flightplan: Dataplane disaggregation and placement for p4 programs. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 571–592. USENIX Association, April 2021. URL: https://www.usenix.org/conference/nsdi21/presentation/sultana.

[107] S. Ghose, A. Boroumand, J. S. Kim, J. Gómez-Luna, and O. Mutlu. Processing-in-memory: A workload-driven perspective. *IBM Journal of Research and Development*, 63(6):3:1–3:19, 2019. DOI: 10.1147/JRD.2019.2934048.

[108] CRAM. Computational ram. URL: https://www.eecg.utoronto.ca/~dunc/cram/.

[109] Zhenyuan Ruan, Tong He, and Jason Cong. INSIDER: Designing In-Storage computing system for emerging High-Performance drive. In *2019 USENIX Annual Technical Conference*, USENIX ATC, pages 379–394, Renton, WA, July 2019. USENIX Association. URL: https://www.usenix.org/conference/atc19/presentation/ruan.

[110] Storage Networking Industry Association. Snia iotta trace repository. URL: http://iotta.snia.org/traces/block-io/388?only=386.

[111] Devesh Tiwari, Simona Boboila, Sudharshan Vazhkudai, Youngjae Kim, Xiaosong Ma, Peter Desnoyers, and Yan Solihin. Active flash: Towards Energy-Efficient, In-Situ data analytics on Extreme-Scale machines. In *11th USENIX Conference on File and Storage Technologies (FAST 13)*, pages 119–132, San Jose, CA, February 2013. USENIX Association. URL: https://www.usenix.org/conference/fast13/technical-sessions/presentation/tiwari.

[112] Yangwook Kang, Yang-suk Kee, Ethan L. Miller, and Chanik Park. Enabling cost-effective data processing with smart ssd. In *IEEE 29th Symposium on Mass Storage Systems and Technologies*, MSST, pages 1–12, 2013. DOI: 10.1109/MSST.2013.6558444.

[113] Akshitha Sriraman and Thomas F Wenisch. μTune: Auto-Tuned Threading for OLDI Microservices. In *Proceedings of the 13th USENIX Conference on Operating Systems Design and Implementation*, OSDI, page 177–194, USA, 2018. USENIX Association. URL: https://www.usenix.org/conference/osdi18/presentation/sriraman.

[114] Yu Gan, Yanqi Zhang, Dailun Cheng, Ankitha Shetty, Priyal Rathi, Nayan Katarki, Ariana Bruno, Justin Hu, Brian Ritchken, Brendon Jackson, Kelvin Hu, Meghna Pancholi, Yuan He, Brett Clancy, Chris Colen, Fukang Wen, Catherine Leung, Siyuan Wang, Leon Zaruvinsky, Mateo Espinosa, Rick Lin, Zhongling Liu, Jake Padilla, and Christina Delimitrou. An Open-Source Benchmark Suite for Microservices and Their Hardware-Software Implications for Cloud & Edge Systems. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 3–18, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3297858.3304013.

[115] Zhipeng Jia and Emmett Witchel. Nightcore: Efficient and Scalable Serverless Computing for Latency-Sensitive, Interactive Microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 152–166, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3445814.3446701.

[116] Yu Gan, Mingyu Liang, Sundar Dev, David Lo, and Christina Delimitrou. Sage: Practical and Scalable ML-Driven Performance Debugging in Microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 135–151, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3445814.3446700.

[117] Shutian Luo, Huanle Xu, Chengzhi Lu, Kejiang Ye, Guoyao Xu, Liping Zhang, Yu Ding, Jian He, and Chengzhong Xu. Characterizing Microservice Dependency and Performance: Alibaba Trace Analysis. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC '21, page 412–426, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3472883.3487003.

[118] Shuang Chen, Christina Delimitrou, and José F. Martínez. PARTIES: QoS-Aware Resource Partitioning for Multiple Interactive Services. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 107–120, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3297858.3304005.

[119] Yanqi Zhang, Weizhe Hua, Zhuangzhuang Zhou, G. Edward Suh, and Christina Delimitrou. Sinan: ML-based and QoS-aware resource management for cloud microservices. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 167–181, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3445814.3446693.

[120] Shutian Luo, Huanle Xu, Kejiang Ye, Guoyao Xu, Liping Zhang, Jian He, Guodong Yang, and Chengzhong Xu. Erms: Efficient Resource Management for Shared Microservices with SLA Guarantees. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1*, ASPLOS 2023, page 62–77, New York, NY, USA, 2022. Association for Computing Machinery. DOI: 10.1145/3567955.3567964.

[121] Kaihua Fu, Wei Zhang, Quan Chen, Deze Zeng, and Minyi Guo. Adaptive Resource Efficient Microservice Deployment in Cloud-Edge Continuum. *International Parallel and Distributed Processing Symposium*, 33(8):1825–1840, August 2022. DOI: 10.1109/TPDS.2021.3128037.

[122] Haoran Qiu, Subho S. Banerjee, Saurabh Jha, Zbigniew T. Kalbarczyk, and Ravishankar K. Iyer. FIRM: An Intelligent Fine-grained Resource Management Framework for SLO-Oriented Microservices. In *Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation*, OSDI, USA, 2020. USENIX Association. URL: https://www.usenix.org/conference/osdi20/presentation/qiu.

[123] Amirhossein Mirhosseini, Sameh Elnikety, and Thomas F. Wenisch. Parslo: A Gradient Descent-based Approach for Near-optimal Partial SLO Allotment in Microservices. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC ’21, page 442–457, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3472883.3486985.

[124] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS 2023, page 118–132, New York, NY, USA, 2023. Association for Computing Machinery. DOI: 10.1145/3575693.3575754.

[125] Noman Bashir, Tian Guo, Mohammad Hajiesmaili, David Irwin, Prashant Shenoy, Ramesh Sitaraman, Abel Souza, and Adam Wierman. Enabling Sustainable Clouds: The Case for Virtualizing the Energy System. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC ’21, page 350–358, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3472883.3487009.

[126] Abel Souza, Noman Bashir, Jorge Murillo, Walid Hanafy, Qianlin Liang, David Irwin, and Prashant Shenoy. Ecovisor: A Virtual Energy System for Carbon-Efficient Applications. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, ASPLOS 2023, page 252–265, New York, NY, USA, 2023. Association for Computing Machinery. DOI: 10.1145/3575693.3575709.

[127] Helping you pick the greenest region for your Google Cloud resources. https://cloud.google.com/blog/topics/sustainability/pick-the-google-cloud-region-with-the-lowest-co2. (Accessed on 04/26/2024).

[128] Ana Radovanović, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, and Nick Care. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems*, 38(2):1270–1280, 2022. DOI: 10.1109/TPWRS.2022.3173250.

[129] Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. Let’s Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud. In *Proceedings of the 22nd International Middleware Conference*, Middleware ’21, page 260–272, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3464298.3493399.

[130] Saurabh Kadekodi, Francisco Maturana, Sanjith Athlur, Arif Merchant, K. V. Rashmi, and Gregory R. Ganger. Tiger: Disk-Adaptive redundancy without placement restrictions. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 413–429, Carlsbad, CA, July 2022. USENIX Association. URL: https://www.usenix.org/conference/osdi22/presentation/kadekodi.

[131] Saurabh Kadekodi, Shashwat Silas, David Clausen, and Arif Merchant. Practical design considerations for wide locally recoverable codes (LRCs). In *21st USENIX Conference on File and Storage Technologies (FAST*

23), pages 1–16, Santa Clara, CA, February 2023. USENIX Association. URL: https://www.usenix.org/conference/fast23/presentation/kadekodi.

[132] Eduardo Pinheiro, Ricardo Bianchini, and Cezary Dubnicki. Exploiting redundancy to conserve energy in storage systems. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '06/Performance '06, page 15–26, New York, NY, USA, 2006. Association for Computing Machinery. DOI: 10.1145/1140277.1140281.

[133] Junaid Shuja, Kashif Bilal, Sajjad A. Madani, Mazliza Othman, Rajiv Ranjan, Pavan Balaji, and Samee U. Khan. Survey of Techniques and Architectures for Designing Energy-Efficient Data Centers. *IEEE Systems Journal*, 10(2):507–519, 2016. DOI: 10.1109/JSYST.2014.2315823.

[134] Jiechao Gao, Haoyu Wang, and Haiying Shen. Smartly Handling Renewable Energy Instability in Supporting A Cloud Datacenter. In *2020 IEEE International Parallel and Distributed Processing Symposium*, IPDPS, pages 769–778, 2020. DOI: 10.1109/IPDPS47924.2020.00084.

[135] Gong Chen, Wenbo He, Jie Liu, Suman Nath, Leonidas Rigas, Lin Xiao, and Feng Zhao. Energy-Aware Server Provisioning and Load Dispatching for Connection-Intensive Internet Services. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, NSDI, page 337–350, USA, 2008. USENIX Association. URL: https://www.usenix.org/legacy/event/nsdi08/tech/full_papers/chen/chen.pdf.

[136] Christina Delimitrou and Christos Kozyrakis. Paragon: QoS-aware Scheduling for Heterogeneous Datacenters. In *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 77–88, New York, NY, USA, 2013. Association for Computing Machinery. DOI: 10.1145/2451116.2451125.

[137] Christina Delimitrou and Christos Kozyrakis. Quasar: Resource-Efficient and QoS-Aware Cluster Management. In *Proceedings of the 19th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS, page 127–144, New York, NY, USA, 2014. Association for Computing Machinery. DOI: 10.1145/2541940.2541941.

[138] Exos x18 data sheet. https://www.seagate.com/content/dam/seagate/migrated-assets/www-content/datasheets/pdfs/exos-x18-channel-DS2045-4-2106US-en_US.pdf, 2021.

[139] Exos x10 data sheet. https://www.seagate.com/files/www-content/datasheets/pdfs/exos-x-10DS1948-1-1709-GB-en_GB.pdf, 2021.

[140] Chris Mellor. WD and Tosh talk up penta-level cell flash. https://blocksandfiles.com/2019/08/07/penta-level-cell-flash/ 5/17/22.

[141] Bichlien Nguyen, Julie Sinistore, Jake Smith, Praneet S. Arshi, Lauren M. Johnson, Tim Kidman, T.J. diCaprio, Doug Carmean, and Karin Strauss. Architecting datacenters for sustainability: Greener data storage using synthetic dna. In *Electronics Goes Green 2020*. Fraunhofer IZM, IEEE, September 2020. URL: https://www.microsoft.com/en-us/research/publication/architecting-datacenters-for-sustainability-greener-data-storage-using-synthetic-dna/.

[142] Patrick Anderson, Erika Blancada Aranas, Youssef Assaf, Raphael Behrendt, Richard Black, Marco Caballero, Pashmina Cameron, Burcu Canakci, Thales De Carvalho, Andromachi Chatzieleftheriou, Rebekah Storan Clarke, James Clegg, Daniel Cletheroe, Bridgette Cooper, Tim Deegan, Austin Donnelly, Rokas Drevinskas, Alexander Gaunt, Christos Gkantsidis, Ariel Gomez Diaz, Istvan Haller, Freddie Hong, Teodora Ilieva, Shashidhar Joshi, Russell Joyce, Mint Kunkel, David Lara, Sergey Legtchenko, Fanglin Linda Liu, Bruno Magalhaes, Alana Marzoev, Marvin Mcnett, Jayashree Mohan, Michael Myrah, Trong Nguyen, Sebastian Nowozin, Aaron Ogus, Hiske Overweg, Antony Rowstron, Maneesh Sah, Masaaki Sakakura, Peter Scholtz, Nina Schreiner, Omer Sella, Adam Smith, Ioan Stefanovici, David Sweeney, Benn Thomsen, Govert Verkes, Phil Wainman, Jonathan Westcott, Luke Weston, Charles Whittaker, Pablo Wilke Berenguer, Hugh

Williams, Thomas Winkler, and Stefan Winzeck. Project Silica: Towards Sustainable Cloud Archival Storage in Glass. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, 2023. DOI: 10.1145/3600006.3613208.

[143] M. Umar, S. V. Moore, J. S. Meredith, J. S. Vetter, and K. W. Cameron. Aspen-based performance and energy modeling frameworks. *Journal of Parallel and Distributed Computing*, 2017. DOI: 10.1016/j.jpdc.2017.11.005.

[144] Naveen Muralimanohar, Rajeev Balasubramonian, and Norman P Jouppi. CACTI 6.0: A tool to model large caches. *HP Laboratories*, 27:28, 2009.

[145] Sheng Li, Jung Ho Ahn, Richard D. Strong, Jay B. Brockman, Dean M. Tullsen, and Norman P. Jouppi. The McPAT Framework for Multicore and Manycore Architectures: Simultaneously Modeling Power, Area, and Timing. *ACM Transactions on Architecture & Code Optimization*, 10(1):5:1–5:29, April 2013. URL: http://doi.acm.org/10.1145/2445572.2445577, DOI: 10.1145/2445572.2445577.

[146] David Brooks, Vivek Tiwari, and Margaret Martonosi. Wattch: a Framework for Architectural-level Power Analysis and Optimizations. In *Proceedings of the 27th Annual International Symposium on Computer Architecture*, ISCA, page 83–94, New York, NY, USA, 2000. Association for Computing Machinery. DOI: 10.1145/339647.339657.

[147] Vijay Kandiah, Scott Peverelle, Mahmoud Khairy, Junrui Pan, Amogh Manjunath, Timothy G. Rogers, Tor M. Aamodt, and Nikos Hardavellas. AccelWattch: A Power Modeling Framework for Modern GPUs. In *Proceedings of the 54th IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, page 738–753, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3466752.3480063.

[148] Gene Wu, Joseph L. Greathouse, Alexander Lyashevsky, Nuwan Jayasena, and Derek Chiou. GPGPU Performance and Power Estimation Using Machine Learning. In *IEEE 21st International Symposium on High Performance Computer Architecture*, HPCA, pages 564–576, 2015. DOI: 10.1109/HPCA.2015.7056063.

[149] Wesley Brewer, Matthias Maiterth, Vineet Kumar, Rafal Wojda, Sedrick Bouknight, Jesse Hines, Woong Shin, Scott Greenwood, Wesley Williams, David Grant, and Feiyi Wang. A Digital Twin Framework for Liquid-cooled Supercomputers as Demonstrated at Exascale. In *The International Conference for High Performance Computing, Networking, Storage, and Analysis*, SC, November 2024. DOI: 10.1109/SC41406.2024.00029.

[150] Jude Alnas, Muneer Alshowkan, Nageswara S. V. Rao, Nicholas A. Peters, and Joseph M. Lukens. Optimal resource allocation for flexible-grid entanglement distribution networks. *Opt. Express*, 30(14):24375–24393, Jul 2022. URL: https://opg.optica.org/oe/abstract.cfm?URI=oe-30-14-24375, DOI: 10.1364/OE.458358.

[151] Hsuan-Hao Lu, Muneer Alshowkan, Jude Alnas, Joseph M. Lukens, and Nicholas A. Peters. Procrustean entanglement concentration in quantum-classical networking. *Phys. Rev. Appl.*, 21:044027, Apr 2024. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.21.044027, DOI: 10.1103/PhysRevApplied.21.044027.

[152] William L. Miller, Deborah Bard, Amber Boehnlein, Kjiersten Fagnan, Chin Guok, Eric Lançon, Sreeranjani Ramprakash, Mallikarjun Shankar, Nicholas Schwarz, and Benjamin L. Brown. Integrated research infrastructure architecture blueprint activity (final report 2023). Technical report, US Department of Energy (USDOE), Washington, DC (United States). Office of Science; Lawrence Berkeley National Laboratory (LBNL), Berkeley, CA (United States), 07 2023. URL: https://www.osti.gov/biblio/1984466, DOI: 10.2172/1984466.

[153] Lisa Su and Sam Naffziger. 1.1 Innovation For the Next Decade of Compute Efficiency. In *2023 IEEE International Solid-State Circuits Conference*, ISSCC, 2023. DOI: 10.1109/ISSCC42615.2023.10067810.

[154] Hongzi Mao, Malte Schwarzkopf, Shaileshh Bojja Venkatakrishnan, Zili Meng, and Mohammad Alizadeh. Learning Scheduling Algorithms for Data Processing Clusters. In *Proceedings of the ACM Special Interest Group on Data Communication*, SIGCOMM '19, page 270–288, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3341302.3342080.

[155] Yuping Fan, Boyang Li, Dustin Favorite, Naunidh Singh, Taylor Childers, Paul Rich, William Allcock, Michael E. Papka, and Zhiling Lan. Dras: Deep reinforcement learning for cluster scheduling in high per-

formance computing. *IEEE Transactions on Parallel and Distributed Systems*, 33(12):4903–4917, 2022. DOI: 10.1109/TPDS.2022.3205325.

[156] A. Mu'alem and D. Feitelson. Utilization, Predictability, Workloads, and User Runtime Estimates in Scheduling the IBM SP2 with backfilling. *IEEE Trans on Parallel and Distributed Systems*, 2001. DOI: 10.1109/71.932708.

[157] Bartłomiej Kocot, Paweł Czarnul, and Jerzy Proficz. Energy-aware scheduling for high-performance computing systems: A survey. *Energies*, 16(2), 2023. URL: https://www.mdpi.com/1996-1073/16/2/890, DOI: 10.3390/en16020890.

[158] Sean Wallace, Xu Yang, Venkat Vishwanath, William Allcock, Susan Coghlan, Michael E. Papka, and Zhiling Lan. A Data Driven Scheduling Approach for Power Management on HPC Systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, SC, 2016. DOI: 10.1109/SC.2016.55.

[159] Jie Li, George Michelogiannakis, Brandon Cook, Dulanya Cooray, and Yong Chen. Analyzing Resource Utilization in an HPC System: A Case Study of NERSC's Perlmutter. In *High Performance Computing: 38th International Conference, ISC High Performance 2023, Hamburg, Germany, May 21–25, 2023, Proceedings*, 2023. DOI: 10.1007/978-3-031-32041-5_16.

[160] Baolin Li, Rohin Arora, Siddharth Samsi, Tirthak Patel, William Arcand, David Bestor, Chansup Byun, Rohan Basu Roy, Bill Bergeron, John Holodnak, Michael Houle, Matthew Hubbell, Michael Jones, Jeremy Kepner, Anna Klein, Peter Michaleas, Joseph McDonald, Lauren Milechin, Julie Mullen, Andrew Prout, Benjamin Price, Albert Reuther, Antonio Rosa, Matthew Weiss, Charles Yee, Daniel Edelman, Allan Vanterpool, Anson Cheng, Vijay Gadepally, and Devesh Tiwari. AI-Enabling Workloads on Large-Scale GPU-Accelerated System: Characterization, Opportunities, and Implications. In *2022 IEEE International Symposium on High-Performance Computer Architecture*, HPCA, 2022. DOI: 10.1109/HPCA53966.2022.00093.

[161] Zhengji Zhao, Ermal Rrapaj, Sridutt Bhalachandra, Brian Austin, Hai Ah Nam, and Nicholas Wright. Power Analysis of NERSC Production Workloads. In *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, SC-W '23, page 1279–1287, New York, NY, USA, 2023. Association for Computing Machinery. DOI: 10.1145/3624062.3624200.

[162] Akshaya Jagannadharao, Nicole Beckage, Sovan Biswas, Hilary Egan, Jamil Gafur, Thijs Metsch, Dawn Nafus, Giuseppe Raffa, and Charles Tripp. A beginner's guide to power and energy measurement and estimation for computing and machine learning, 2024. URL: https://arxiv.org/abs/2412.17830, arXiv:2412.17830.

[163] Walid A. Hanafy, Qianlin Liang, Noman Bashir, Abel Souza, David Irwin, and Prashant Shenoy. Going Green for Less Green: Optimizing the Cost of Reducing Cloud Carbon Emissions. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS '24, page 479–496, New York, NY, USA, 2024. Association for Computing Machinery. DOI: 10.1145/3620666.3651374.

[164] Noman Bashir, Tian Guo, Mohammad Hajiesmaili, David Irwin, Prashant Shenoy, Ramesh Sitaraman, Abel Souza, and Adam Wierman. Enabling sustainable clouds: The case for virtualizing the energy system. In *Proceedings of the ACM Symposium on Cloud Computing*, SoCC '21, page 350–358, New York, NY, USA, 2021. Association for Computing Machinery. DOI: 10.1145/3472883.3487009.

[165] Liuzixuan Lin and Andrew A Chien. Adapting Datacenter Capacity for Greener Datacenters and Grid. In *Proceedings of the 14th ACM International Conference on Future Energy Systems*, e-Energy '23, page 200–213. ACM, June 2023. URL: http://dx.doi.org/10.1145/3575813.3595197, DOI: 10.1145/3575813.3595197.

[166] Pratyush Patel, Esha Choukse, Chaojie Zhang, Íñigo Goiri, Brijesh Warrier, Nithish Mahalingam, and Ricardo Bianchini. Characterizing Power Management Opportunities for LLMs in the Cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS, page 207–222, New York, NY, USA, 2024. Association for Computing Machinery. DOI: 10.1145/3620666.3651329.

[167] Grant L. Stewart, Gregory A. Koenig, Jingjing Liu, Anders Clausen, Sonja Klingert, and Natalie Bates. Grid Accommodation of Dynamic HPC Demand. In *Workshop Proceedings of the 48th International Conference on Parallel Processing*, ICPP Workshops '19, New York, NY, USA, 2019. Association for Computing Machinery. DOI: 10.1145/3339186.3339214.

[168] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood. The gem5 simulator. *SIGARCH Comput. Archit. News*, 39(2):1–7, 2011. DOI: 10.1145/2024716.2024718.

[169] Jason Lowe-Power, Abdul Mutaal Ahmad, Ayaz Akram, Mohammad Alian, Rico Amslinger, Matteo Andreozzi, Adrià Armejach, Nils Asmussen, Brad Beckmann, Srikant Bharadwaj, Gabe Black, Gedare Bloom, Bobby R. Bruce, Daniel Rodrigues Carvalho, Jeronimo Castrillon, Lizhong Chen, Nicolas Derumigny, Stephan Diestelhorst, Wendy Elsasser, Carlos Escuin, Marjan Fariborz, Amin Farmahini-Farahani, Pouya Fotouhi, Ryan Gambord, Jayneel Gandhi, Dibakar Gope, Thomas Grass, Anthony Gutierrez, Bagus Hanindhito, Andreas Hansson, Swapnil Haria, Austin Harris, Timothy Hayes, Adrian Herrera, Matthew Horsnell, Syed Ali Raza Jafri, Radhika Jagtap, Hanhwi Jang, Reiley Jeyapaul, Timothy M. Jones, Matthias Jung, Subash Kannoth, Hamidreza Khaleghzadeh, Yuetsu Kodama, Tushar Krishna, Tommaso Marinelli, Christian Menard, Andrea Mondelli, Miquel Moreto, Tiago Mück, Omar Naji, Krishnendra Nathella, Hoa Nguyen, Nikos Nikoleris, Lena E. Olson, Marc Orr, Binh Pham, Pablo Prieto, Trivikram Reddy, Alec Roelke, Mahyar Samani, Andreas Sandberg, Javier Setoain, Boris Shingarov, Matthew D. Sinclair, Tuan Ta, Rahul Thakur, Giacomo Travaglini, Michael Upton, Nilay Vaish, Ilias Vougioukas, William Wang, Zhengrong Wang, Norbert Wehn, Christian Weis, David A. Wood, Hongil Yoon, and Éder F. Zulian. The gem5 simulator: Version 20.0+, 2020. URL: https://arxiv.org/abs/2007.03152, DOI: 10.48550/ARXIV.2007.03152.

[170] A. F. Rodrigues, K. S. Hemmert, B. W. Barrett, C. Kersey, R. Oldfield, M. Weston, R. Risen, J. Cook, P. Rosenfeld, E. Cooper-Balis, and B. Jacob. The structural simulation toolkit. *SIGMETRICS Perform. Eval. Rev.*, 38(4):37–42, 2011. DOI: 10.1145/1964218.1964225.

[171] Misbah Mubarak, Christopher D Carothers, Robert B Ross, and Philip Carns. Enabling Parallel Simulation of Large-scale HPC Network Systems. *IEEE Transactions on Parallel and Distributed Systems*, 28(1):87–100, 2017. DOI: 10.1109/TPDS.2016.2543725.

Disclaimer: This report (HTTPS://DOI.ORG/10.2172/2476961) was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government.

U.S. DEPARTMENT of ENERGY | Office of Science