

# Power Consumption Trends in Supercomputers: A Study of NERSC’s Cori and Perlmutter Machines

Ermal Rrapaj

*Advanced Technologies Group (NERSC)  
Lawrence Berkeley National Laboratory*

Berkeley, USA  
ermalrrapaj@lbl.gov

Sridutt Bhalachandra

*NVIDIA*  
sriduttb@nvidia.com

Zhengji Zhao

*Advanced Technologies Group (NERSC)  
Lawrence Berkeley National Laboratory*

Berkeley, USA  
zzhao@lbl.gov

Brian Austin

*Advanced Technologies Group (NERSC)  
Lawrence Berkeley National Laboratory*

Berkeley, USA  
baustin@lbl.gov

Hai Ah Nam

*Advanced Technologies Group (NERSC)  
Lawrence Berkeley National Laboratory*

Berkeley, USA  
hnam@lbl.gov

Nicholas J. Wright

*Advanced Technologies Group (NERSC)  
Lawrence Berkeley National Laboratory*

Berkeley, USA  
njwright@lbl.gov

**Abstract**—The rising power demands of supercomputers put high importance on understanding the underlying sources of power use. We compare a comprehensive set of power measurements covering six months from two supercomputers, the Cori and Perlmutter machines at the National Energy Research Scientific Computing Center (NERSC). We show that power usage varies considerably, and is always significantly below the peak provisioned power. Several factors cause this – the machine may not be fully utilized, applications’ computational characteristics are not those which maximize power usage, and/or applications can be waiting on resources external to the node. Our analysis shows that while the power usage of applications in the same science domain is similar, the power usage of the same application run by different users is even more similar. As NERSC transitioned to GPU accelerated nodes, the peak power capabilities increased, but the production workload’s power demands did not increase at the same rate, further decreasing the fraction of thermal design power (TDP) used. These results indicate that future machines could be power capped and over-provisioned and a metric different than thermal peak design is needed for future procurement, in alignment with the actual power needs of production workloads. These results suggest that with appropriate technologies, such as power-aware scheduling or dynamic power management, future HPC systems could be operated with power caps well below TDP, avoiding the high cost of over-provisioned infrastructure.

## I. INTRODUCTION

In 2008, the exascale computing study [1] put forth a definitive challenge to deliver an exascale machine within a 20 megawatts (MWs) power budget. The current exascale machines are already over this limit. As of November 2023, the current #1 Top500 [2] listed fastest supercomputer, Frontier, consumes 22.7 MW of power at 1200 PFLOPS, while the #2 listed supercomputer Aurora, consumes around 24.7 MW at 600 PFLOPS [3]. Power usage of high performance computing (HPC) resources continues to be a concern to the community, because it translates into significant one-time fixed costs, such as the installation of a greater amount of power distribution or cooling infrastructure, as well as ongoing costs, to purchase

electricity. In addition, the rise of energy consumption by the HPC community and large machine learning models in industry has environmental impacts, e.g. carbon emission, which need to be taken into consideration as well. In short, power demands in HPC have become a key limiting factor in both current exascale system operations and the design of future supercomputers [4].

As a result, there has been much research into power and energy usage of HPC resources. However, much of this research uses benchmarks that are not always representative of the actual workloads that run on today’s supercomputers. This disparity often makes the insights presented too generic, too specific, or infeasible for practical purposes, however elegant they may be. As such, there is a need for studies that shed light on power consumption trends in modern supercomputers, even more so given the underlying challenges in collecting, monitoring and analyzing facility and systems data at the supercomputer level [5]. Anecdotally, it is often reported that production power draws from HPC resources are significantly less than the Thermal Design Power (TDP), but the reasons for this are unclear. As a result, the values quoted for TDP give an upper bound on the power consumption observed in day-to-day operations. The actual power consumed changes based on the HPC workload running at any given time on a supercomputer and could, in principle, be tuned by the respective facilities to accommodate constraints on the available power. Investigating the power characteristics of HPC production workloads can reveal potential avenues towards improving system power efficiency and help guide future system procurement. With advancements in power measurement techniques and data collection over the past years, it has become possible in most systems to observe total power usage over any time range of interest, as well as at individual nodes and their component levels [6].

We investigate such issues by studying power data collected in production, on a per-node basis, for leadership-

class HPC resources, the Cori machine [7] for the time period February-August 2019 and the Perlmutter machine [8] for the time period February-August 2023 at NERSC. In 2019 more than 4,000 unique users ran their applications on Cori, and in 2023 there were more than 9,000 unique users on Perlmutter, making these architectures perfect testing grounds for understanding the breadth of possible power usage scenarios on an HPC machine. The Cori system, comprised only by CPU partitions, was previously analyzed in [9] and the Perlmutter system, comprised of both CPU and GPU partitions, was recently analysed in [10]. Our comprehensive analysis includes a power timeline for the full system and daily variation analysis by comparing measured power to models. Furthermore, we perform set of micro-benchmarks and application-based benchmarks to gain an understanding of limiting cases and the causes of the variation in power draw. We conclude by performing a breakdown by scientific domain and application name with the aim of understanding power trends and differences across the years and HPC architectures.

The data for the Cori machine is already in the public domain [11], and the data collection for Perlmutter will also be made public. The primary contributions of this work are as follows.

- The average power of both Cori and Perlmutter systems is well below their TDPs and that the gap between average power and TDP is larger on the newer, GPU-based Perlmutter system.
- Variations in Perlmutter system power timeline are well explained by the starting and stopping of jobs with different time-averaged power demands. This contrasts with Cori, where temporal power variation within individual jobs was the largest contributor to the system’s power variation.
- Applications running on Perlmutter typically draw less power than STREAM (memory bandwidth intensive) or DGEMM (flop-intensive) microbenchmarks. The distinction between micro-benchmarks and production workloads is not apparent on Cori, where the CPUs were the only compute domain.

This paper is organized as follows. Section II covers related work, Section III describes Cori and Perlmutter systems and the data collection framework in more detail. Section IV analyses the power at system level and Section V provides our findings from a set of micro-benchmarks on a single node. Section VI breaks down the power measurements based on workload characteristics. Finally, Section VII presents our concluding remarks.

## II. RELATED WORK

With the emphasis on energy efficiency in supercomputing centers over the last decade, power management has become a crucial aspect of design and operation. Most of the HPC research in power management has centered around four key areas - operations/infrastructure, scheduling, operating systems/runtimes, and analysis/modeling/surveys.

As the infrastructure considerations for supercomputer sites and datacenters are fairly consistent, scheduling plays a pivotal

role in accommodating the differences in workloads. Consequently, a large amount of HPC research in power management has centered around scheduling. In 2010, a power-budget-guided jobs scheduling policy that maximizes overall job performance [12] was proposed and was followed by other solutions that used Dynamic Voltage and Frequency Scaling (DVFS) [13], [14]. Another solution proposed during this time tried to minimize the number of active servers of a system while still satisfying incoming application requests [15]. More recently, a data-driven scheduling approach for power management based on profiling data of production jobs runs has been proposed [16]. In [17], an approach to factor and mitigate manufacturing variability is proposed. There have been also other approaches focusing on over-provisioned HPC systems [18], [19].

A considerable amount of effort has focused on improving job energy efficiency, performance, or both during execution through operating system and runtime improvements. The runtime efforts have targeted mostly MPI and OpenMP with attempts to mitigate problems due to computational workload imbalance, waiting on memory, communication, and others. These works predominantly target the CPUs, but research focused on GPUs too is gaining momentum [20]–[23]. Many works focus on reclaiming slack in the presence of workload imbalance using controls like DVFS and Dynamic Duty Cycle Modulation (DDCM) [24]–[33]. The processor expends significant power while waiting on memory, so several efforts have focused on mitigating idle waiting [34]–[38]. Attempts to eliminate idle waiting during communication have also been suggested [39]–[43]. Similarly, improving resource utilization through concurrency throttling [44]–[46] and switching of components [47], [48] have too been explored.

The existing analysis/modeling/survey works form a basis for many of the above efforts, and the current work is an effort in this direction. In 2005, a framework for direct and automatic profiling of power consumption for non-interactive, parallel scientific applications was proposed [49]. Power measurements for various computational loads on large scale HPC systems has been studied showing that the Linpack benchmark consumed power very close to any subset of a typical compute-intensive scientific workload. [50]. A comprehensive definition and evaluation of memory power estimation and limiting algorithm that significantly improves sensing accuracy, power limit enforcement, and system performance has been presented [51]. The power-specific features of several architectures [52] and the interaction of supercomputing centers with their electricity service providers have also been studied [53]. There has been considerable work to leverage the fact that systems operate using less than their TDP [54]–[56]. Recent work [57] looks at the impact of power capping on the notion of application progress and proposes a model to capture the general behavior of the progress of different classes of applications under a power cap. In [58] a power analysis of one year time period of the Summit machine is provided.

With few exceptions, many of these solutions are not always representative of production workloads and considerations.

And, more comprehensive studies that show the power consumption of production supercomputers at different scales are still needed for a better understanding. To tailor future energy-efficient HPC solutions for production, studies that provide key information about the overall system, its power demand fluctuations, as well as job and application-specific trends, are necessary. Consequently, our current work hopes to fill this gap by providing a comprehensive analysis of two HPC machines and drawing trends for future machines from the combined analysis.

### III. SYSTEM CONFIGURATION

#### A. Cori

The Cori supercomputer [7] is a Cray XC40 with a peak performance of about 30 PFLOPS. Cori is comprised of 12,076 compute nodes, a 30 PB Lustre scratch filesystem, and a first-of-its kind NVRAM “burst buffer” storage system, all connected by a Cray Aries interconnect. Cori’s compute nodes are of two types. The primary compute partition has 9,688 nodes with one 68-core 1.4 GHz Intel Xeon Phi 7250 “Knight’s Landing” (KNL) processor, 96 GB of 2400 MHz DDR4 memory, and 16 GB of high-bandwidth MCDRAM and thermal design power (TDP) of 215 W. A second partition is composed of 2,388 “Haswell”-based nodes, each with two 16-core 2.3 GHz Intel Xeon E5-2698 v3 processors, 128 GB of 2133 MHz DDR4 memory, and a TDP of 135 W per processor. Cori is NERSC’s longest running machine. The supercomputer had its first users in 2015 and after 8 years it was decommissioned in 2023.

#### B. Perlmutter

The Perlmutter supercomputer is based on the HPE Cray Shasta platform with a theoretical peak performance of about 70 PFLOPS. Perlmutter is comprised of 1,792 GPU-accelerated nodes and 3,072 CPU-only nodes, and an all-flash Lustre system with 35 PB of disk space all interconnected with the HPE Slingshot network. Each GPU-accelerated node contains one 64-core 2.45 GHz AMD EPYC 7763 “Milan” processor, 256 GB DDR4 memory, four NVIDIA A100 GPUs and four HPE Cray Cassini NICs. The TDPs for the GPUs and CPUs are 400 W and 280 W, respectively yielding total node TDP of 2340 W, including all other components in the node. The CPU-only nodes have two AMD EPYC 7763 Milan processors, 512 GB DDR4 memory, one Cassini NIC, and a node TDP of 700 W. The first phase of the Perlmutter [8] installation was completed in 2021, and various hardware and network upgrades continued until February 2023.

There is a drastic change in architecture between Cori and Perlmutter. There is a dramatic increase in the node TDP, coinciding with the larger number of processors and growing TDP (and performance) per processor. System-wide, the performance and power efficiency have improved. On Cori-KNL, the high performance Linpack (HPL) benchmark achieved 14 PFLOPS using 3.94 MW. On Perlmutter-GPU, HPL reached 79 PFLOPW with only 2.95 MW.

#### C. Power Measurement Infrastructure

Power consumption can be measured from various sources throughout the systems with different levels of spatial and temporal granularity. System-level power is measured via Modbus [59], the electrical industry power reporting standard for electrical equipment. The total power is the sum of Modbus measurements from the electrical substations, and includes, besides the compute cabinets, blower cabinets and disk cabinets. Cabinet-level power data is obtained from the electrical whips that power each cabinet; these revenue-grade meters measure AC power with high resolution and accuracy. Node-level power measurements (including all peripherals) are obtained through Cray’s power monitoring (PM) architecture [60]. The Cray XC blade design used on Cori includes a microprocessor that measures the power consumption of each node on the blade. The highest resolution power measurements are obtained from Intel’s Running Average Power Limit (RAPL) counters [51]. RAPL probes a processor’s voltage regulators to determine power measurements for the processor package and its DRAM and can be sampled at very high frequencies. On Perlmutter, the highest resolution measurements are from NVIDIA’s Data Center GPU Manager (DCGM, version 3.1.6). NERSC also collects power usage data for the cooling units. However, as shown in [10], these components require much less power than the compute racks, and are not analyzed in this work.

NERSC uses the Lightweight Distributed Metric Service (LDMS) [61] to aggregate the Cray PM and DCGM counters, as well as other performance metrics. Although the underlying measurement interfaces are capable of higher sampling rates our LDMS configuration downsamples these measurements to 1 Hz to keep the total data volume and ingest rates manageable. The data are then stored by NERSC’s Operations Monitoring and Notification Infrastructure (OMNI) [6], which was developed to monitor and track operational data from NERSC systems.

#### D. Physical and Mechanical Infrastructure

Several energy-efficiency considerations allow NERSC to operate an energy efficient data center. For instance, the system dissipates HPC waste heat directly to the outside environment and avoids vapor-compression based air conditioning as much as possible. NERSC participates in the LBNL Energy and Water Management Program and tracks power usage effectiveness (PUE), IT power usage effectiveness (ITUE), and water usage effectiveness (WUE). NERSC maintained average PUE scores of 1.08 during 2019, and 1.05 during 2023.

### IV. SYSTEM POWER

Figure 1a shows the power consumption, measured through Modbus, of Cori and Perlmutter for six consecutive months, February to August, averaged over 1 hour intervals during 2019 for Cori and 2023 for Perlmutter. As we are interested in power management of *active* systems, this data excludes times when the system power is less than 0.1 MW for both systems, such as when maintenance required the system to shut

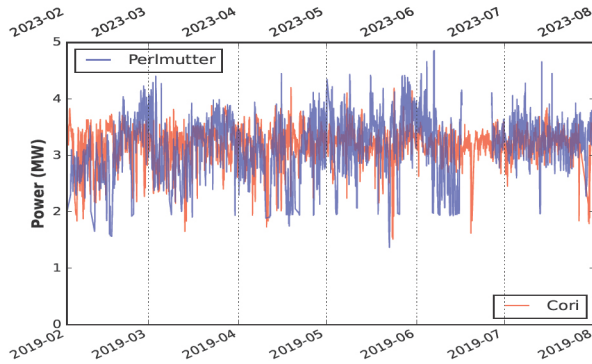
HPC System	System Power (MW)			
	Average	Standard Deviation	Maximum	TDP
Cori	3.18	0.36	4.21	5.72
Perlmutter	3.19	0.49	4.86	6.90

TABLE I: Distribution of system-level power measurements Cori and Perlmutter.

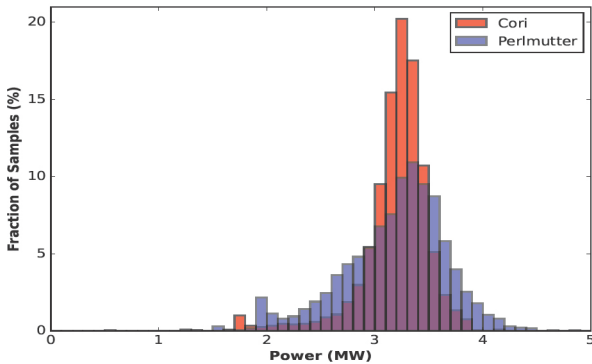
down or be idle. There is a conspicuous gap visible during the month of June 2023 on Perlmutter which corresponds to an infrastructure upgrade period for OMNI. Some understanding of breadth of the power distribution can be gleaned by examining the timeline in Figure 1a. Rapid (hour-to-hour) fluctuations with swings up to 1.89 MW for Cori and 2.13 MW for Perlmutter are typical, but slower (e.g. seasonal) variations are not evident.

The distribution of the one hour power samples for both supercomputers is illustrated in Figure 1b and summarized in Table I. Despite a 20% increase in TDP, the average power on Perlmutter is the same as Cori. Thus, the average fraction of TDP used dropped- from 56% on Cori to 46% on Perlmutter, as did the maximum fraction of TDP- from 73% to 70%. However, the power *variation* increased by 36%.

One possible explanation for the low power demands (relative to TDP) is low system utilization, but our data do not



(a) Timeline



(b) Distribution

Fig. 1: Power consumption of NERSC’s Cori and Perlmutter systems over a six-month period.

support this. In Figure 2 we display the two dimensional distribution of the total system power and the percentage of the compute nodes allocated by the job scheduler at the time of the power measurement. On both systems, the system utilization is typically close to 80% (and higher on Perlmutter), well above the 46-56% power utilization. Other features of Figure 2 point to sources of power variation. The perceptible slopes of the distributions suggest that utilization may be a contributing factor, while scatter around the (imagined) trendlines indicates the importance of other factors, such as workload-dependent power draw.

To estimate the importance of different sources of Cori’s power fluctuation over time, Bhalachandra [9] compared a series of approximate model power timelines. All of the models were constructed from the same set of measured power data, but each model used a different averaging scheme to control which sources of variation it included. Here, we use the same approach to understand power fluctuation on Perlmutter, and begin by providing a brief description of each model.

The utilization power model ( $P_{util}$ ) includes only the effects of system utilization. (See the preceding discussion of Figure 2.) It combines an average active power value and idle power for active and idle nodes respectively,

$$P_{util}(t) = P_{idle} N_{idle}(t) + P_{active} N_{active}(t). \quad (1)$$

The active power is the average of all Cray-PM power records from nodes that were active at the time of measurement, and the measurement of idle power is described in detail in Section V.

The job-mix power model ( $P_{mix}$ ) includes the effects of system utilization and adds the variation that occurs when a low-power job is replaced by a high-power job in the scheduling and vice versa,

$$P_{mix}(t) = P_{idle} N_{idle}(t) + \sum_{j \in jobs(t)} \bar{P}_j \quad (2)$$

$\bar{P}_j$  is the (time-averaged) job-level power usage, which we derived by integrating the Cray-PM records over the node-lists and wall-times recorded in the Slurm [62] jobs database. The sum includes only jobs that are running at time  $t$ .

The augmented job-mix power model ( $P_{aug}$ ) takes into consideration the effects of large jobs on the system’s power fluctuation by substituting the large jobs’ average power with their actual timelines,  $P_k(t)$ ,

$$P_{Aug} = P_{idle} N_{idle}(t) + \sum_{k \in large\ jobs(t)} P_k(t) + \sum_{j \in other\ jobs(t)} \bar{P}_j \quad (3)$$

The emphasis on large jobs is related to the prevalence of bulk synchronous programming models, which can cause large power swings if, for example, all of the nodes in the job simultaneously stop computing as they enter I/O phases. In [9], large jobs used at least 1,024 of Cori’s KNL nodes (10.5% of that partition). In our analysis, large jobs are those that use at least 224 of Perlmutter’s GPU-accelerated nodes (12.5% of that partition). If the large job threshold was eliminated, then

all the  $\bar{P}_j$  would be replaced by  $P_k(t)$ , and the augmented job-mix power model would reproduce the measured power timeline exactly.

The modeled power timelines for Perlmutter’s GPU partition are shown in Figure 3. The utilization model is shown in red, the job-mix model in green, and the augmented job-mix model in yellow. In Table II, we report the accuracy of the power models using the metric  $1 - \text{RMSE}/\text{STD}$ , where RMSE is the root mean squared error, and STD is the standard deviation of the measured power variation. We have also included the results of the analysis on the Cori machine [9] for comparison. On both systems, the utilization model accounts for only a modest fraction (7-8%) of the systems total variation, which matches the consistently high utilization shown in Figure 2. However, the job-mix model accounts for an additional 28% of Perlmutter’s variation, but only 6% of Cori’s, implying that the jobs on Perlmutter have a broader distribution of average power per node than those on Cori. The causes of this will be investigated in Section V. A noteworthy observation of [9], was that power variation on Cori was dominated by the transient power fluctuations within large jobs, accounting for 60% of the total variation. This effect is much less pronounced on Perlmutter, where it accounts for only 12%. The difference is explained, in part, by the fraction of node hours used by large jobs on the selected days: 43% on Cori and 26% on Perlmutter.

The power fluctuations within individual jobs, which are described by the augmented job-mix model, have both temporal and spatial components. In figure 4a we plot the timeline of one large-scale application running on Perlmutter: the X-point Gyrokinetic Code (XGC) [63] is a whole-volume, total-f gyrokinetic particle-in-cell code developed for modelling tokamak fusion reactors. The timeline is marked by rapid power spikes of up to 125 kW. To better understand the distribution among the 224 GPU-accelerated nodes active during its run, we select multiple time snapshots of the power distribution among nodes in figure 4b. The bimodal distributions and long tails are indications of load imbalance as some nodes are performing high power computation while others are close to

Model	HPC Machine			
	Cori		Perlmutter	
	Accuracy (%)	$\Delta$ (%)	Accuracy (%)	$\Delta$ (%)
Utilization	8	-	7	-
Job – Mix	14	6	35	28
Augmented	74	60	47	12

TABLE II: Summary of power variation explainable by the Utilization, Job-Mix, and Augmented Job-Mix models for Cori and Perlmutter. The higher the value, the better the model is at explaining the measured variation.  $\Delta$  is the improvement of each model with respect to the previous one in terms of the difference of the respective values.

idle power levels. Similar behavior was also present among the applications studied on Cori [9].

## V. SINGLE-NODE POWER CONSUMPTION

The effectiveness of the Job-Mix power model suggests that the power requirements for each job can vary significantly. This section examines the application characteristics that contribute to these difference. We begin our analysis by measuring the power used by four simple single-node activity patterns (Idle, STREAM, DGEMM and Firestarter) that represent limiting cases for application behavior. The Idle test is no-op and represents the baseline power requirement of an inactive node. The STREAM [64] kernel, and its GPU version BabelStream [65], measure memory bandwidth and stress the memory subsystem. The DGEMM kernel exercises the node’s floating-point performance by performing double-precision matrix-matrix multiplication. Firestarter [66] is a power virus designed to drive all of the subsystems simultaneously and determine a node’s maximum achievable power ( $P_{max}$ ). All benchmarks were compiled with CUDA [67] for the GPU-accelerated nodes and the benchmarks on the KNL nodes were compiled with Intel Math Kernel Library [68]. Each application was run on 50 unique nodes, and the power draw was averaged over the execution time. To avoid any temperature effects on power measurement each run was at least 30 minutes long [69]. The total power of each node was obtained from the Cray PM counters.

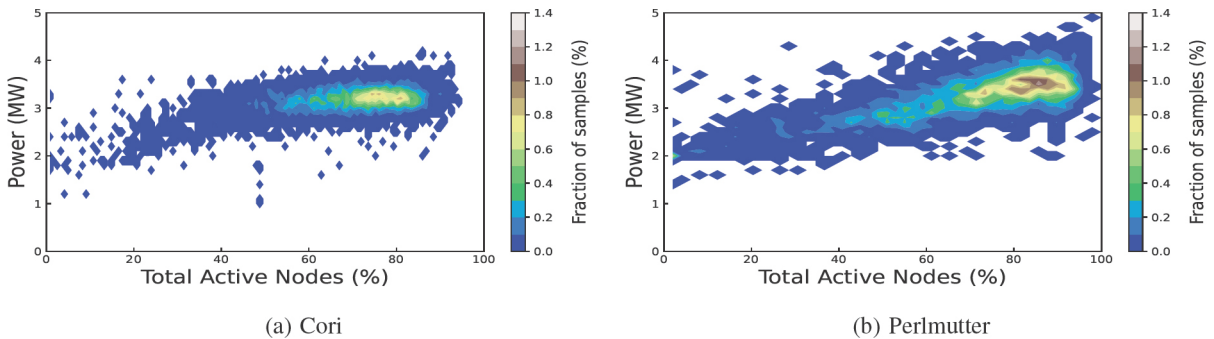


Fig. 2: Distribution of the correlation between system utilization and system power. Cori has more than twice the number of nodes in comparison to Perlmutter, but a lower range of system power values. Both systems have a high concentration of nodes for system power values close to 3 MW.



Figure 5 shows the fraction of TDP used by the two node types when running these kernels. For all benchmarks studied here, the variation in power measured about the mean is less

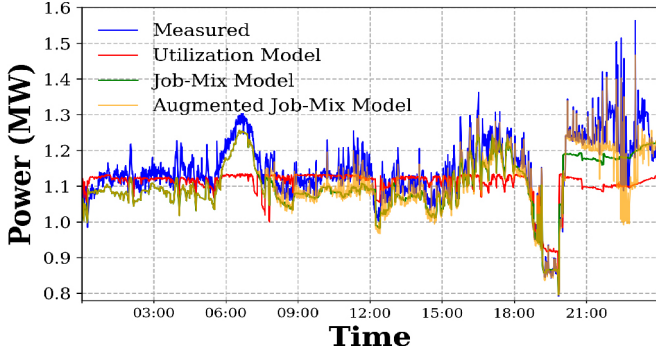
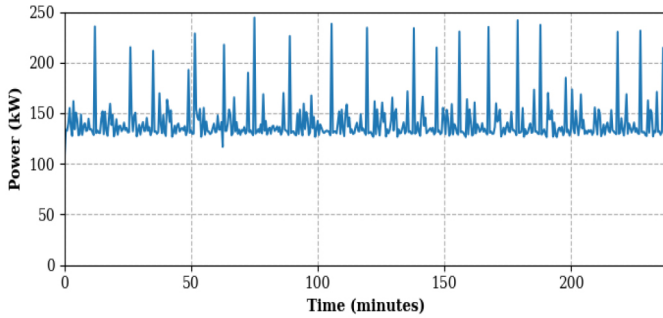
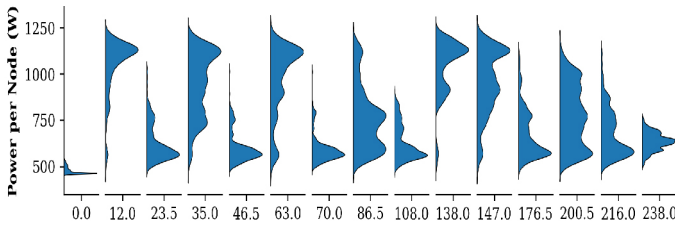


Fig. 3: Utilization Model for the GPU partition on Perlmutter: a two-parameter model reflecting only whether a node is idle or active. (All jobs use the same average power.) Job-Mix Model: adds job-specific power values to the utilization model. Errors result from each job’s power changing over time. Augmented Job-Mix Model: small jobs are treated by the Job-Mix model, and large jobs are represented by their measured time-dependent power values. Residual errors result from temporal fluctuations in small jobs’ power.



(a) Timeline of the total power of the XGC application



(b) Node power distribution for XGC application as function of time, in minutes, since the start of the run. The selected times show both power fluctuations and load imbalance.

Fig. 4: Execution snapshots of the power consumption, time intervals in minutes, for one of the large jobs from figure 3 on the GPU partition of Perlmutter. The bimodal distributions in some of the snapshots is likely due to load imbalance during the run.

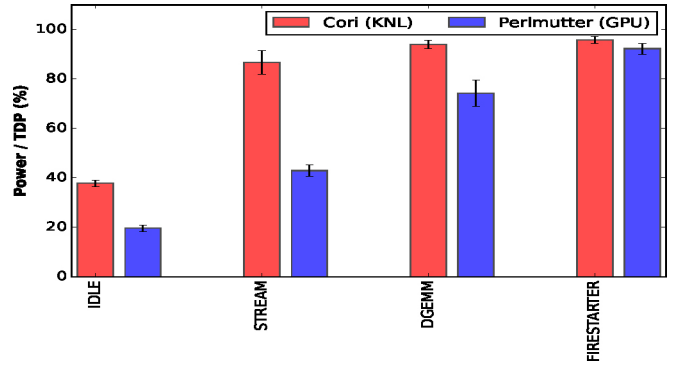


Fig. 5: Power consumption of several microbenchmarks. Total node power was measured using Cray PM counters. Error bars denote the standard deviation of the total power across 50 runs.

than 8% on both machines. As there are four GPUs and one CPU per Perlmutter node, the absolute power demands are naturally higher than Cori’s single-CPU nodes. The idle power of the single-socket KNL nodes is 110 W, which is about 37% of the TDP, while the idle power of the GPU nodes is around 460 W, quite a bit higher than KNL, but only about 20% of the respective TDP. A sharp difference between the two architectures is evident in the memory bandwidth kernels: on Cori, the STREAM benchmark reaches 90% of the  $P_{max}$ , while on Perlmutter, BabelStream reaches only 40%. The power for DGEMM (with random input [70]) is 90% of  $P_{max}$  on Cori’s KNL nodes and 80% of  $P_{max}$  on Perlmutter’s GPU nodes. When running Firestarter, Cori’s KNL nodes use 90% of TDP and Perlmutter’s GPU nodes use 90%.

The difference in power consumption between these two benchmarks on Cori and Perlmutter is due to architectural differences. On Cori, the CPU is the only compute available; actively used by both. On Perlmutter, DGEMM primarily uses the GPUs, while Firestarter utilizes the whole compute power of the node (both CPU and GPU are actively performing operations at the same time). This distinction becomes visible when we compare the power draw of the individual components. For DGEMM the average power draw of each GPU is about 83% of its TDP, and for the CPU it is 23% of its TDP. For Firestarter the respective values are 72% and 56%. In both cases most of the power needs are due to the activity of the GPUs with the increased CPU activity for Firestarter likely being the reason behind the difference observed at node power level. We also checked the power draw for the memory in the node, and it was 76 W for DGEMM and 96 W for Firestarter.

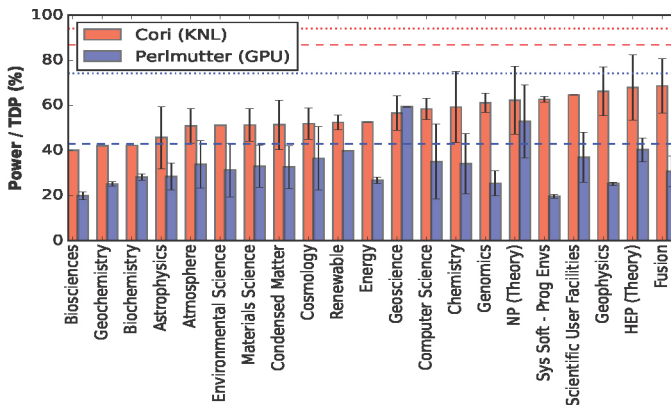
As we are interested in both power and energy efficiency, we dive deeper into the compute capabilities achieved by these benchmarks. In table III we display the bandwidth and compute performance per unit of energy for STREAM and DGEMM. It is easy to notice that while STREAM energy efficiency is essentially the same in both machines, DGEMM has vastly improved (over six times) due to the presence of four GPUs per node on Perlmutter. For STREAM the average power draw of each GPU was about 49% of its TDP, and for

HPC Machine	STREAM Efficiency [GB/J]	DGEMM Efficiency [Gflop/J]
Cori	1.22	6.85
Perlmutter	1.36	42.10

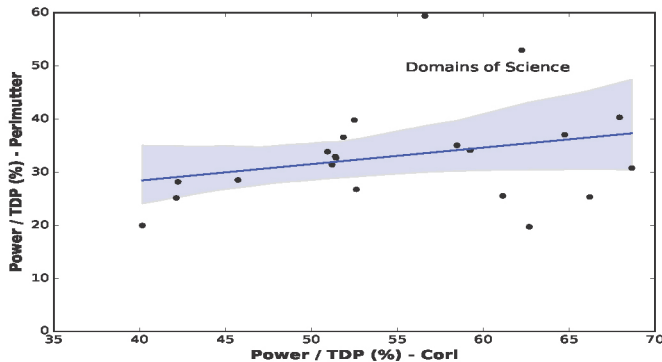
TABLE III: STREAM and DGEMM node energy efficiency in terms of bandwidth and compute performance per unit of energy per node on the Cori and Perlmutter machines.

the CPU it 25%. This is a similar power consumption with respect to DGEMM for the CPU, but much lower with regard to the GPU. The memory power consumption for STREAM was 75 W on average, almost the same as DGEMM. While power demand has increased from Cori to Perlmutter, energy efficiency has increased as well.

## VI. DOMAIN AND WORKLOAD ANALYSIS FOR CAPABILITY JOBS



(a) Domains of science workloads power for the two supercomputers. Horizontal dashed (dotted) line is the fraction of TDP by the STREAM (DGEMM) benchmark; Cori-KNL in red and Perlmutter-GPU in blue.



(b) Correlation between the power consumption by domains of science workloads for the two supercomputers. The solid line is a linear fit and 95% confidence region is in blue.

Fig. 6: Node power consumption as fraction of TDP for capability jobs for various domains of science. For the Cori KNL nodes, the TDP is 300 W and for the Perlmutter GPU nodes it is 2.34 kW.

NERSC’s workload is extremely diverse, with contributions from more than 9,000 active users and over 11 million

application runs for the year 2023. In this section, we broaden our scope by analyzing the production workload. We focus on long-running large jobs as defined in the previous section. Jobs at this scale are particularly relevant for power management. On Cori they have the greatest potential to modulate the total system power on practical time-scales [9] and on Perlmutter they can have rather large power variation. As in previous section, the results in this section are focused on the KNL partition on Cori and GPU partition on Perlmutter.

Figures 6 and 7 show the average power consumption per node as fraction of the TDP, as measured using Cray PM counters, for different science domains and also for the most used production applications, respectively. The power consumed by the GPU node on Perlmutter is a lower fraction of TDP than the KNL node on Cori. However, applications on Perlmutter nodes consistently require more raw power due to the high compute capabilities of the four GPUs.

In Figure 6a, these jobs are classified by science domain and ordered in increasing power on Cori. As part of the NERSC allocation request process, every project must self-select a science domain. Columns represent the average over jobs within each domain and the error bars represent one standard deviation from the mean. In horizontal dotted lines we show the power draw from STREAM and in dashed lines the power draw from DGEMM applications from the previous section for comparison. The majority of the codes on Cori require less power draw than these microbenchmarks, while on Perlmutter many applications’ power needs are much closer to STREAM. As the figure shows, the power variation within a domain is generally less than the differences between domains, but the jobs within the same domain of science can also have high variation; about 30% for Astrophysics on Cori and 39% for Chemistry on Perlmutter. On Cori, Fusion codes seem to be the most power-hungry, as they consume about 70% of the TDP on average, while on Perlmutter the highest power consumption is about 59%, reached by Geoscience. In figure 6b we plot the average values of the measured power for the domains of science for Cori versus Perlmutter. Then, we proceed to perform linear fits to see how strong correlations can be. The offsets differ by less than 9% from value obtained by subtracting the idle power measurements from figure 5. The slope is rather small at around 0.31, an indication of small correlation between the power draw from domains of science at the two supercomputers.

Application names identify workloads more specifically than science domains and should be more reliable predictors of power use. In Figure 7a we focus on a few of the most heavily used applications of the jobs from Figure 6a. The range of power measurements is comparable to that of science domain classification, but the variation is lower. There are variations in power consumption across multiple invocations of the same code (the error bars show standard deviation). Likely explanations for these differences include running at different scales, performance variation due to interactions with the network, the storage subsystem and/or thermal/manufacturing variability between nodes. Apart from the machine attributes,

many codes contain more than one algorithm with different power profiles selected via different inputs leading to the variations in power. A few applications, notably VASP, show significantly more power variation across runs than others; likely because the power draw of these codes is more strongly dependent on their input options and data sets. Overall, we observe a higher power variation of workloads on Cori as can be seen from the error bars in Figure 7a. As this plot suggests, applications do not stress the floating point units of the CPU/GPU as much as the DGEMM and Firestarter micro-benchmarks. In figure 7b we plot the average values of the measured power for the selected workloads for Cori versus Perlmutter and perform a linear fit to study correlations between the two systems. The offsets differs by less 5% from value obtained by subtracting the idle power measurements from figure 5. The slope is even smaller than the previous case at 0.26, primarily due to the similar average power draw from the workloads on Perlmutter. Further disentangling the

sources of variation will require further data collection for the workload which delves deeper into the compute and bandwidth requirements, in addition to power. In future research we plan to include a series of experiments to better understand the roots of variation and expand the data collection.

## VII. CONCLUSIONS & FUTURE WORK

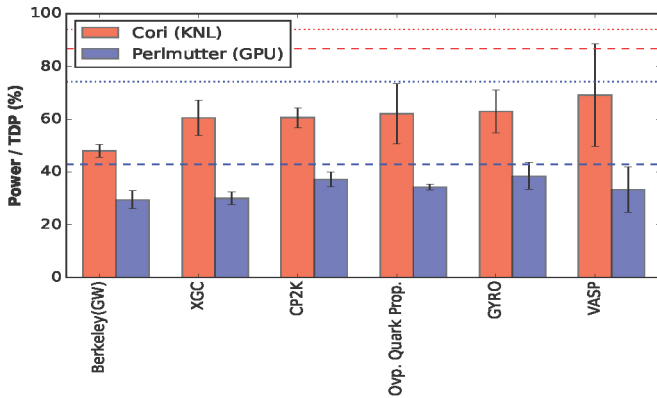
Our study of the power usage of the Cori and Perlmutter supercomputers over the course of six consecutive months on each system explored the reasons for the anecdotal observations that supercomputers' power draws are often significantly less than their TDPs. Not only did we confirm such anecdotal reports with empirical data, but we also note an increased divide between the maximal and practical power draw values. This gap, coupled with the high capital cost of electrical infrastructure, suggests that future HPC systems might be deployed more economically if the infrastructure was sized according to estimates of the systems' operational power draw rather than their TDPs. However, in this scenario, more emphasis would need to be placed on power aware scheduling and power capping to avoid power consumption instances that exceed the infrastructure capabilities.

Our analysis shows that the overall power draw of the machine can be understood by considering the total number of nodes in use and the types of applications running on those nodes. Categorizing jobs by their science domains or application names provides rudimentary power estimates that could allow for example, a very simple power capping scheduling algorithm to be implemented on the time-scale of job durations. However, our post-hoc analysis of the system power timeline revealed that more than half of Perlmutter's power variation occurred on faster time-scales. As machine learning workloads become more common, we expect the intra-job variation to increase due to the compute idle time between training epochs [71]. Further analysis of fine-grained power measurements is clearly needed in order to explain the application behaviors that contribute to their power use.

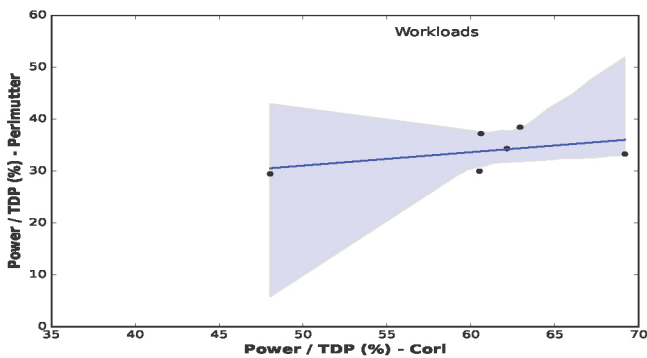
Near-term future work calls for detailed analysis of the data set to further understand the dependency of the power draw upon science domain and application, potentially involving more sophisticated statistical analysis techniques such as machine learning. In the longer term we plan to develop methods to simulate the ability of the scheduler to cap the overall power draw of the system through the use of application and user dependent power signatures to deploy such techniques in production. In addition, TDP greatly affects hardware procurement, but is not representative of workload applications, and we plan to conduct further analysis to devise a more suitable metric.

## REFERENCES

- [1] P. Kogge, K. Bergman, S. Borkar, D. Campbell, W. Carson, W. Dally, M. Denneau, P. Franzone, W. Harrod, K. Hill *et al.*, "Exascale computing study: Technology challenges in achieving exascale systems," 2008.
- [2] (2023). [Online]. Available: <https://top500.org/>
- [3] (2023). [Online]. Available: <https://shorturl.at/bhvW8>



(a) Prevalent workloads power for the two supercomputers. Horizontal dashed (dotted) line is the fraction of TDP by the STREAM (DGEMM) benchmark; Cori-KNL in red and Perlmutter-GPU in blue.



(b) Correlation between the power consumption of the workloads for the two supercomputers. The solid line is a linear fit and 95% confidence region is in blue.

Fig. 7: Node power consumption as fraction of TDP for capability jobs for various workloads. For the Cori KNL nodes, the TDP is 300 kW and for the Perlmutter GPU nodes it is 2.34 kW.



- [4] K. Bergman, S. Borkar, D. Campbell, W. Carlson, W. Dally, M. Denneau, P. Franzon, W. Harrod, K. Hill, J. Hiller *et al.*, “Exascale computing study: Technology challenges in achieving exascale systems,” *Defense Advanced Research Projects Agency Information Processing Techniques Office (DARPA IPTO), Tech. Rep.*, vol. 15, p. 181, 2008.
- [5] E. Bautista, C. Whitney, and T. Davis, “Big data behind big data,” in *Conquering Big Data with High Performance Computing*. Springer, 2016, pp. 163–189.
- [6] E. Bautista, M. Romanus, T. Davis, C. Whitney, and T. Kubaska, “Collecting, monitoring, and analyzing facility and systems data at the national energy research scientific computing center,” in *Proceedings of the 48th International Conference on Parallel Processing: Workshops*. ACM, 2019, p. 10.
- [7] K. Antypas, N. Wright, N. P. Cardo, A. Andrews, and M. Cordery, “Cori: a cray xc pre-exascale system for nersc,” *Cray User Group Proceedings*. Cray, 2014.
- [8] (2023) Perlmutter, a hpe cray ex system. NERSC. [Online]. Available: <https://docs.nersc.gov/systems/perlmutter/architecture/>
- [9] S. Bhalachandra, B. Austin, and N. J. Wright, “Understanding power variation and its implications on performance optimization on the cori supercomputer,” in *2021 International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, 2021, pp. 51–62.
- [10] Z. Zhao, E. Rrapaj, S. Bhalachandra, B. Austin, H. A. Nam, and N. 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*, ser. SC-W ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1279–1287. [Online]. Available: <https://doi.org/10.1145/3624062.3624200>
- [11] S. Bhalachandra, B. Austin, and N. Wright, “NERSC cori power consumption during 2019,” 2020.
- [12] M. Etinski, J. Corbalan, J. Labarta, and M. Valero, “Optimizing job performance under a given power constraint in hpc centers,” in *Green Computing Conference, 2010 International*. IEEE, 2010, pp. 257–267.
- [13] M. Etinski, J. Corbalan, J. Labarta, and M. Valero, “Utilization driven power-aware parallel job scheduling,” *Computer Science-Research and Development*, vol. 25, no. 3-4, pp. 207–216, 2010.
- [14] M. Etinski, J. Corbalan, J. Labarta, and M. Valero, “Parallel job scheduling for power constrained hpc systems,” *Parallel Computing*, vol. 38, no. 12, pp. 615–630, 2012.
- [15] O. Mämmelä, M. Majanen, R. Basmadjian, H. De Meer, A. Giesler, and W. Homberg, “Energy-aware job scheduler for high-performance computing,” *Computer Science-Research and Development*, vol. 27, no. 4, pp. 265–275, 2012.
- [16] S. Wallace, X. Yang, V. Vishwanath, W. E. Allcock, S. Coghlan, M. E. Papka, and Z. 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*. IEEE Press, 2016, p. 56.
- [17] Y. Inadomi, T. Patki, K. Inoue, M. Aoyagi, B. Rountree, M. Schulz, D. Lowenthal, Y. Wada, K. Fukazawa, M. Ueda *et al.*, “Analyzing and mitigating the impact of manufacturing variability in power-constrained supercomputing,” in *SC’15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2015, pp. 1–12.
- [18] O. Sarood, A. Langer, A. Gupta, and L. Kale, “Maximizing throughput of overprovisioned hpc data centers under a strict power budget,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE Press, 2014, pp. 807–818.
- [19] R. Sakamoto, T. Cao, M. Kondo, K. Inoue, M. Ueda, T. Patki, D. Ellsworth, B. Rountree, and M. Schulz, “Production hardware over-provisioning: Real-world performance optimization using an extensible power-aware resource management framework,” in *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2017, pp. 957–966.
- [20] Y. Jiao, H. Lin, P. Balaji, and W.-c. Feng, “Power and performance characterization of computational kernels on the gpu,” in *Proceedings of the 2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010, pp. 221–228.
- [21] D. Rohr, S. Kalcher, M. Bach, A. A. Alaqeily, H. M. Alzaidy, D. Eschweiler, V. Lindenstruth, S. B. Alkhereyfy, A. Alharthiy, A. Al-mubarak *et al.*, “An energy-efficient multi-gpu supercomputer,” in *2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on CyberSpace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC, CSS, ICESS)*. IEEE, 2014, pp. 42–45.
- [22] B. Dutta, V. Adhinarayanan, and W.-c. Feng, “Gpu power prediction via ensemble machine learning for dvfs space exploration,” in *Proceedings of the 15th ACM International Conference on Computing Frontiers*, 2018, pp. 240–243.
- [23] V. Adhinarayanan, B. Dutta, and W.-c. Feng, “Making a case for green high-performance visualization via embedded graphics processors,” in *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*. IEEE, 2018, pp. 721–724.
- [24] V. W. Freeh and D. K. Lowenthal, “Using multiple energy gears in MPI programs on a power-scalable cluster,” in *PPoPP 2005: Proc. of the ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2005.
- [25] N. Kappiah, V. W. Freeh, and D. K. Lowenthal, “Just in time dynamic voltage scaling: Exploiting inter-node slack to save energy in mpi programs,” in *Proceedings of the 2005 ACM/IEEE conference on Supercomputing*. IEEE Computer Society, 2005, p. 33.
- [26] C. Hsu and W. Feng, “A power-aware run-time system for high-performance computing,” in *SC05: Proc. of the 2005 ACM/IEEE Conference on High Performance Networking and Computing*. IEEE Computer Society, 2005.
- [27] R. Ge, X. Feng, and K. W. Cameron, “Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters,” in *SC05: Proc. of the 2005 ACM/IEEE Conference on High Performance Networking and Computing*. IEEE Computer Society, 2005.
- [28] H. Kimura, M. Sato, Y. Hotta, T. Boku, and D. Takahashi, “Empirical Study on Reducing Energy of Parallel Programs Using Slack Reclamation by DVFS in a Power-scalable High Performance Cluster,” in *CLUSTER 2006: Proc. of the 2006 IEEE Intl. Conference on Cluster Computing*. IEEE, 2006.
- [29] B. Rountree, D. K. Lowenthal, B. R. de Supinski, M. Schulz, V. W. Freeh, and T. K. Bletsch, “Adagio: Making DVS practical for complex HPC applications,” in *ICS ’09: Proc. of the 23rd Intl. Conference on Supercomputing*, 2009.
- [30] A. Tiwari, M. Laurenzano, J. Peraza, L. Carrington, and A. Snively, “Green queue: Customized large-scale clock frequency scaling,” in *CGC ’12: Proc. of the 2nd Intl. Conference on Cloud and Green Computing*, Nov. 2012.
- [31] S. Bhalachandra, A. Porterfield, and J. F. Prins, “Using dynamic duty cycle modulation to improve energy efficiency in high performance computing,” in *Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International*. IEEE, 2015, pp. 911–918.
- [32] A. Porterfield, R. Fowler, S. Bhalachandra, B. Rountree, D. Deb, and R. Lewis, “Application runtime variability and power optimization for exascale computers,” in *Proceedings of the 5th International Workshop on Runtime and Operating Systems for Supercomputers*. ACM, 2015, p. 3.
- [33] S. Bhalachandra, A. Porterfield, S. L. Olivier, and J. F. Prins, “An adaptive core-specific runtime for energy efficiency,” in *Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International*. IEEE, 2017, pp. 947–956.
- [34] S. Huang and W. Feng, “Energy-efficient cluster computing via accurate workload characterization,” in *CCGrid 2009: Proc. of the 9th IEEE/ACM Intl. Symposium on Cluster Computing and the Grid*. IEEE Computer Society, 2009.
- [35] S. Eyerhan and L. Eeckhout, “Fine-grained dvfs using on-chip regulators,” *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 8, no. 1, p. 1, 2011.
- [36] K. Livingston, N. Triquenau, T. Fighiera, J. C. Beyler, and W. Jalby, “Computer using too much power? give it a rest (runtime energy saving technology),” *Computer Science-Research and Development*, vol. 29, no. 2, pp. 123–130, 2014.
- [37] W. Wang, A. Porterfield, J. Cavazos, and S. Bhalachandra, “Using Per-Loop CPU Clock Modulation for Energy Efficiency in OpenMP Applications,” in *Proceedings of the 2015 44th International Conference on Parallel Processing (ICPP)*. IEEE Computer Society, 2015, pp. 629–638.
- [38] S. Bhalachandra, A. Porterfield, S. L. Olivier, J. F. Prins, and R. J. Fowler, “Improving energy efficiency in memory-constrained

- applications using core-specific power control,” in *Proceedings of the 5th International Workshop on Energy Efficient Supercomputing*, ser. E2SC’17. New York, NY, USA: ACM, 2017, pp. 6:1–6:8. [Online]. Available: <http://doi.acm.org/10.1145/3149412.3149418>
- [39] K. Kandalla, E. P. Mancini, S. Sur, and D. K. Panda, “Designing power-aware collective communication algorithms for infiniband clusters,” in *Parallel Processing (ICPP), 2010 39th International Conference on*. IEEE, 2010, pp. 218–227.
- [40] A. Vishnu, S. Song, A. Marquez, K. Barker, D. Kerbyson, K. Cameron, and P. Balaji, “Designing energy efficient communication runtime systems for data centric programming models,” in *Proceedings of the 2010 IEEE/ACM Int’l Conference on Green Computing and Communications & Int’l Conference on Cyber, Physical and Social Computing*. IEEE Computer Society, 2010, pp. 229–236.
- [41] V. Sundriyal and M. Sosonkina, “Per-call energy saving strategies in all-to-all communications,” in *Recent Advances in the Message Passing Interface*. Springer, 2011, pp. 188–197.
- [42] T. Hoeffler and D. Moor, “Energy, memory, and runtime tradeoffs for implementing collective communication operations,” *Journal of Supercomputing Frontiers and Innovations*, vol. 1, no. 2, pp. 58–75, 2014.
- [43] A. Venkatesh, A. Vishnu, K. Hamidouche, N. Tallent, D. D. Panda, D. Kerbyson, and A. Hoisie, “A case for application-oblivious energy-efficient mpi runtime,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. ACM, 2015, p. 29.
- [44] M. Curtis-Maury, J. Dzierwa, C. D. Antonopoulos, and D. S. Nikolopoulos, “Online power-performance adaptation of multithreaded programs using hardware event-based prediction,” in *Proceedings of the 20th annual international conference on Supercomputing*. ACM, 2006, pp. 157–166.
- [45] D. Li, B. R. De Supinski, M. Schulz, K. Cameron, and D. S. Nikolopoulos, “Hybrid mpi/openmp power-aware computing,” 2010.
- [46] A. K. Porterfield, S. L. Olivier, S. Bhalachandra, and J. F. Prins, “Power measurement and concurrency throttling for energy reduction in openmp programs,” in *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*. IEEE, 2013, pp. 884–891.
- [47] A. Youssef, M. Anis, and M. Elmasry, “Dynamic standby prediction for leakage tolerant microprocessor functional units,” in *Microarchitecture, 2006. MICRO-39. 39th Annual IEEE/ACM International Symposium on*. IEEE, 2006, pp. 371–384.
- [48] J. Leverich, M. Monchiero, V. Talwar, P. Ranganathan, and C. Kozyrakis, “Power management of datacenter workloads using per-core power gating,” *Computer Architecture Letters*, vol. 8, no. 2, pp. 48–51, 2009.
- [49] X. Feng, R. Ge, and K. W. Cameron, “Power and energy profiling of scientific applications on distributed systems,” in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. IEEE, 2005, pp. 34–34.
- [50] S. Kamil, J. Shalf, and E. Strohmaier, “Power efficiency in high performance computing,” in *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*. IEEE, 2008, pp. 1–8.
- [51] H. David, E. Gorbato, U. R. Hanebutte, R. Khanna, and C. Le, “Rapl: memory power estimation and capping,” in *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*. IEEE, 2010, pp. 189–194.
- [52] D. Hackenberg, R. Schöne, T. Ilsche, D. Molka, J. Schuchart, and R. Geyer, “An energy efficiency feature survey of the intel haswell processor,” 2015.
- [53] N. Bates, G. Ghatikar, G. Abdulla, G. A. Koenig, S. Bhalachandra, M. Sheikhalishahi, T. Patki, B. Rountree, and S. Poole, “Electrical grid and supercomputing centers: An investigative analysis of emerging opportunities and challenges,” *Informatik-Spektrum*, vol. 38, no. 2, pp. 111–127, 2015.
- [54] R. Ge and K. W. Cameron, “Power-aware speedup,” in *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2007, pp. 1–10.
- [55] S. Song, C.-Y. Su, R. Ge, A. Vishnu, and K. W. Cameron, “Iso-energy-efficiency: An approach to power-constrained parallel computation,” in *2011 IEEE International Parallel & Distributed Processing Symposium*. IEEE, 2011, pp. 128–139.
- [56] T. Patki, D. K. Lowenthal, B. Rountree, M. Schulz, and B. R. de Supinski, “Exploring hardware overprovisioning in power-constrained, high performance computing,” in *Proceedings of the 27th international ACM conference on International conference on supercomputing*. ACM, 2013, pp. 173–182.
- [57] S. Ramesh, S. Perarnau, S. Bhalachandra, A. D. Malony, and P. Beckman, “Understanding the impact of dynamic power capping on application progress,” in *2019 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 2019, pp. 793–804.
- [58] W. Shin, V. Oles, A. M. Karimi, J. A. Ellis, and F. Wang, “Revealing power, energy and thermal dynamics of a 200pf pre-exascale supercomputer,” in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC ’21. New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/10.1145/3458817.3476188>
- [59] “Modbus application protocol,” 2012. [Online]. Available: [http://modbus.org/docs/Modbus\\_Application\\_Protocol\\_V1\\_1b3.pdf](http://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf)
- [60] S. J. Martin and M. Kappel, “Cray xc30 power monitoring and management,” in *Cray User Group Conference Proceedings*, 2014.
- [61] A. Agelastos, B. Allan, J. Brandt, P. Cassella, J. Enos, J. Fullop, A. Gentile, S. Monk, N. Naksinehaboon, J. Ogden *et al.*, “The lightweight distributed metric service: a scalable infrastructure for continuous monitoring of large scale computing systems and applications,” in *SC’14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2014, pp. 154–165.
- [62] A. B. Yoo, M. A. Jette, and M. Grondona, “Slurm: Simple linux utility for resource management,” in *Workshop on Job Scheduling Strategies for Parallel Processing*. Springer, 2003, pp. 44–60.
- [63] M. D. J. Cole, R. Hager, T. Moritaka, J. Dominski, R. Kleiber, S. Ku, S. Lazerson, J. Riemann, and C. S. Chang, “Verification of the global gyrokinetic stellarator code XGC-S for linear ion temperature gradient driven modes,” *Physics of Plasmas*, vol. 26, no. 8, p. 082501, 08 2019. [Online]. Available: <https://doi.org/10.1063/1.5109259>
- [64] J. D. McCalpin, “Stream benchmark,” *Link: www.cs.virginia.edu/stream/ref.html#what*, vol. 22, 1995.
- [65] “Evaluating attainable memory bandwidth of parallel programming models via babelstream,” *Int. J. Comput. Sci. Eng.*, vol. 17, no. 3, p. 247–262, jan 2018.
- [66] D. Hackenberg, R. Oldenburg, D. Molka, and R. Schöne, “Introducing firestarter: A processor stress test utility,” in *2013 International Green Computing Conference Proceedings*. IEEE, 2013, pp. 1–9.
- [67] NVIDIA, P. Vingelmann, and F. H. Fitzek, “Cuda, release: 10.2.89,” 2020. [Online]. Available: <https://developer.nvidia.com/cuda-toolkit>
- [68] E. Wang, Q. Zhang, B. Shen, G. Zhang, X. Lu, Q. Wu, and Y. Wang, *Intel Math Kernel Library*. Cham: Springer International Publishing, 2014, pp. 167–188. [Online]. Available: [https://doi.org/10.1007/978-3-319-06486-4\\_7](https://doi.org/10.1007/978-3-319-06486-4_7)
- [69] A. Porterfield, R. Fowler, S. Bhalachandra, and W. Wang, “Openmp and mpi application energy measurement variation,” in *Proceedings of the 1st International Workshop on Energy Efficient Supercomputing*. ACM, 2013, p. 7.
- [70] S. Bhalachandra, B. Austin, S. Williams, and N. J. Wright, “Understanding the impact of input entropy on fpu, cpu, and gpu power,” 2022.
- [71] A. Govind, S. Bhalachandra, Z. Zhao, E. Rrapaj, B. Austin, and H. A. Nam, “Comparing power signatures of hpc workloads: Machine learning vs simulation,” in *Proceedings of the SC ’23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*, ser. SC-W ’23. New York, NY, USA: Association for Computing Machinery, 2023, p. 1890–1893. [Online]. Available: <https://doi.org/10.1145/3624062.3624274>