Understanding VASP Power Profiles on NVIDIA A100 GPUs

Zhengji Zhao
Lawrence Berkeley National Laboratory
Bekeley, USA
zzhao@lbl.gov

Brian Austin
Lawrence Berkeley National Laboratory
Bekeley, USA
baustin@lbl.gov

Ermal Rrapaj
Lawrence Berkeley National Laboratory
Bekeley, USA
ermalrrapaj@lbl.gov

Nicholas J. Wright
Lawrence Berkeley National Laboratory
Bekeley, USA
njwright@lbl.gov

***Abstract*—Power is a critical limiting factor in supercomputing as systems scale to exascale levels. To advance scientific computing, supercomputers must operate efficiently under limited power budgets. Power-aware scheduling can help by enforcing power management strategies, but this requires a deep understanding of application power behavior, especially on modern GPU-centric supercomputers. This study examines the power behavior of VASP, a leading HPC application, on the Perlmutter A100 GPU system at NERSC. We explore how VASP's power usage changes with various inputs and parallelism, and assess its response to power cappings. We find that VASP's power usage varies significantly with different workloads, more so than with parallel concurrency. Additionally, power capping GPUs to 50% of their Thermal Design Power can be applied to most VASP workloads with less than a 10% performance loss. These findings shed light on the feasibility and effectiveness of power-aware scheduling based on application power profiles on HPC systems.**

***Index Terms*—application power profile, power capping, performance, A100 GPU, VASP, computing center**

I. Introduction

As HPC enters the exascale era, power has become a critical limiting factor in supercomputing. To advance scientific computing at scale, supercomputers must operate efficiently under limited power budgets. Power-aware scheduling [1]–[3] based on application power profiles has been proposed as a crucial approach to facilitating system operations under power limits. This approach has the potential to keep the total system power within a prescribed budget and optimize performance by allocating power where demand is most critical. Despite its potential, few computing centers currently employ power-aware scheduling. The principal challenge lies in accurately predicting the power consumption of applications on rapidly evolving HPC platforms, necessitating a deep understanding of applications' power behavior.

Decade-long efforts have aimed to understand the power usage characteristics of HPC production workloads at both the system and application levels and build power and energy prediction models [5]–[8]. However, many of these studies focused on older generations of CPU and GPU architectures and have questionable relevance to today's GPU-centric HPC platforms.

Recent efforts have targeted application power behavior on modern GPU systems [9], [10], but they often examine specific instances within a broader spectrum of application power behaviors, making their findings insufficient for comprehensive power management. The most critical gap remains an in-depth understanding of applications' power variations in production settings, which is essential for accurately predicting application power usage.

Additionally, application power profiles are architecture-dependent, while cutting-edge systems often have short lifetimes. Therefore, strategies must be developed to transition power study findings into production deployment promptly.

An earlier study on NERSC's Cori system [11] implied that job's input could have a strong influence on its power use [12]. Recent analysis of NERSC's Perlmutter system [13] showed that 65% of the variation in the system power consumption was due to temporal variation in the power used by individual jobs [14]. This paper addresses both gaps through a detailed analysis of a single application: VASP [15], [16], a widely used materials science application, examining the variation in VASP job's power use and exploring how those patterns change with job input and concurrency. In addition, we evaluate the impact of GPU power caps on VASP performance to identify opportunities for efficient power management.

Our investigation shows that:

- VASP's power consumption is highly dependent on input parameters.
- VASP's power usage varies significantly with different workloads, more so than with parallel concurrency.
- Power capping GPUs to 50% of their Thermal Design Power (TDP) can be applied to most VASP workloads with less than a 10% performance loss on NVIDIA A100 GPUs.

The rest of the paper is organized as follows: Section II describes the hardware, software, and monitoring configurations. In Section III, we present VASP's power usage across

979-8-3503-5554-3/24/$31.00 ©2024 IEEE
DOI 10.1109/SCW63240.2024.00189

a diverse suite of benchmark problems. Section IV provides a systematic analysis of silicon supercells and how their power is affected by perturbing the various parameters sampled by the full benchmark suite. This is followed by a discussion of our results in Section VI. We conclude with a review of related work in Section VII.

II. System Configuration and Environment Setup

A. Perlmutter System Configuration

This work was performed on the Perlmutter [13] supercomputer at NERSC. Perlmutter is a heterogeneous system that integrates 1,792 GPU-accelerated nodes and 3,072 CPU-only nodes within a single HPE Cray Shasta platform. Each GPU-accelerated node contains one AMD EPYC 7763 "Milan" processor, 256 GB of DDR4 memory, four NVIDIA A100 Tensor Core GPUs, and four HPE Slingshot "Cassini" Network Interface Cards (NICs). Among the GPU-accelerated nodes, 256 have 80 GB of High Bandwidth Memory (HBM), and 1,536 have 40 GB of HBM. This work uses only the 40 GB GPU-accelerated nodes. More information about Perlmutter is available online [13].

Perlmutter's total system TDP, including CPU-only nodes, GPU-accelerated nodes, service nodes, network routers, and cooling distribution units is 6.9 MW. The TDP of a 40 GB GPU node is 2,350 W, which includes 280 W for the CPU, 400 W for each of the GPUs, and 470 W for peripherals (used primarily by the DDR memory and NICs).

B. Power Measurement

NERSC's Operations Monitoring and Notification Infrastructure (OMNI) [17] gathers and manages operational data from across the NERSC data center. The data include power measurements from sensors distributed throughout the system. The Cray Power Monitoring interface [18] runs on all compute nodes and provides access to power measurements for key components of the node (the CPU, each GPU, and the DDR memory), and the total node power (which includes the aforementioned components plus peripherals such as NICs). The node-level measurements are forwarded to OMNI's data store using the Lightweight Distributed Metric Service (LDMS) [19]. LDMS samples the measurements at one-second intervals, [1] but the high aggregate data rate across the system forces much of the data to be dropped, leading to an effective sampling interval of 2 seconds throughout this study. The node and GPU power were obtained from OMNI using previously-developed querying scripts [20].

C. VASP

The Vienna Ab initio Simulation Package (VASP) [15], [16] is widely used by materials scientists to compute the electronic structure and atomic-scale properties using plane-wave density functional theory (DFT) [21], [22]. The fundamental equation solved by VASP is an eigenvalue problem given by

$$[-\tfrac{1}{2}\nabla^2 + V(\mathbf{r})]\Psi_i(\mathbf{r}) = \epsilon_i\Psi_i(\mathbf{r}) \qquad i = 1, 2, ..., N$$

where the eigenfunctions, $\Psi_i(r)$, are the "one-electon" orbitals, the eigenvalues, ϵ_i, are the orbital energies, and N is the number of eigenpairs to be solved. The potential function, $V(r)$, includes terms due to electron-ion interaction and a functional that describes electron-electron interaction. The eigen-problem is nonlinear because the inter-electronic functional depends on the orbital functions. It is therefore solved iteratively via self-consistent iteration cycle until a desired accuracy is achieved. The problem is discretized by expanding the orbitals in a plane-wave basis.

VASP predominantly utilizes Fortran 90 and relies extensively on FFTs and linear algebra libraries. VASP implements multiple levels of parallelism, expressed via a combination of MPI for distributing work across nodes, OpenMP for multi-/many-core CPUs, and OpenACC for GPUs, all within a single code-base. For GPU communications, VASP uses NVIDIA's Collective Communications Library (NCCL) [23] as an alternative to MPI.

We conducted our tests using VASP version 6.4.1. The OpenACC port [24] was built to support both OpenACC and OpenMP on Perlmutter's GPU nodes. The software stack used in this study includes: NVIDIA HPC SDK 22.7 for the NVIDIA compiler, CUDA 12.0 (CUDA driver: 525.105.17), QD, cuBLAS, cuSOLVER, and cuFFT libraries, NCCL 2.19.4, Cray MPICH 8.1.28, MKL from Intel oneAPI 22.1.0 and its FFTW3 wrappers to FFT, and HDF5 1.12.2.

VASP has three binaries: gamma-point-only (vasp_gam), standard k-points (vasp_std), and non-collinear (vasp_ncl), each designed for crystal structures with varying levels of symmetry. In our tests, we used the standard vasp_std, which is the most commonly used by users.

III. Power Profiles of VASP Benchmarks

In this section, we study the power characteristics of a diverse set of seven VASP benchmarks running on Perlmutter.

A. Benchmark Descriptions

VASP computations are categorized into two main types: basic density functional theory (DFT) calculations and more computationally intensive higher-order methods, such as hybrid functional calculations, HSE [25], and random phase approximation, RPA/ACFDTR [26]. Within each functional calculation, there are numerous variations tailored to systems with diverse chemical elements. We selected the most commonly used DFT functionals to represent these variations: LDA (CA) and GGA [27]. In this study, we used seven test cases [28], namely Si256_hse, B.hR105_hse, PdO4, PdO2, GaAsBi-64, CuC_vdw, and Si128_acfdtr. These selections represent NERSC's diverse VASP production workloads, ensuring a comprehensive coverage of various code paths, elements, and problem sizes. For example, two HSE hybrid functional calculations with different atomic configurations and problem sizes are selected in the tests: Si256_hse is a 256-atom silicon supercell with a vacancy, and B.hR105_hse is a hexa-boron structure containing 105 atoms. Additionally, we included PdO4 and PdO2, comprising PdO slabs containing 348 and

[1] There is an ongoing effort to improve the data ingest pipeline.

TABLE I
SEVEN VASP BENCHMARKS WERE CHOSEN TO COVER REPRESENTATIVE WORKLOADS AND TO EXERCISE DIFFERENT CODE PATHS.

	Si256_hse	**B.hR105_hse**	**PdO4**	**PdO2**	**GaAsBi-64**	**CuC_vdw**	**Si128_acfdtr**
Electrons (Ions)	1020 (255)	315 (105)	3288 (348)	1644 (174)	266 (64)	1064 (98)	512 (128)
Functional	HSE	HSE	DFT (LDA)	DFT (LDA)	DFT (GGA)	VDW	ACFDT/RPA
Algo	CG (Damped)	CG (Damped)	RMM (VeryFast)	RMM (VeryFast)	BD+RMM (Fast)	RMM (VeryFast)	ACFDTR
NELM (NELMDL)	41 (0)	17 (0)	60 (0)	60 (0)	60 (0)	60 (0)	
NBANDS	640	256	2048	1024	192	640	
NBANDSEXACT							23506
FFT grids	80x80x80	48x48x48	80x120x54	80x60x54	70x70x70	70x70x210	60x60x60
NPLWV	512000	110592	518400	259200	343000	1029000	216000
KPOINTS (KPAR)	1 1 1 (1)	1 1 1 (1)	1 1 1 (1)	1 1 1 (1)	4 4 4 (2)	3 3 1 (1)	1 1 1 (1)

174 atoms, respectively, to evaluate the commonly employed DFT functional calculation utilizing the RMM-DIIS iteration scheme. Furthermore, a ternary alloy structure, GaAsBi-64, was included to cover the metallic systems with the default iteration scheme, Block Davidson + RMM-DIIS algorithms. Lastly, Si128_acfdtr, a 128-atom silicon supercell, performs random phase approximation calculations (RPA/ACFDTR).

Table I details the computational specifics of these benchmarks. While the number of ions or electrons represents physical system sizes, the number of plane waves (NPLWV) or bands (NBANDS) is a computational representation of physical system sizes. It's worth mentioning that these benchmarks were meticulously designed to ensure load balancing among MPI tasks, with limited I/O to facilitate accurate benchmarking. (See the VASP Wiki page [29] for further details on the tags in Table I.)

B. Execution & Measurement Protocol

All our tests were designed to understand VASP's power characteristics and dynamics. We used the same setting as in [28] for the run-time configurations and tuning. This includes the CPU and GPU process affinity choice, run time environments, etc. For example, all tests used one OpenMP thread because, for the VASP GPU port, OpenMP threads have minor performance impacts for the majority of workloads. Four MPI tasks per node (one MPI task per GPU) were used in all tests. There is limited I/O in the tests. The VASP performance was measured with total runtime.

A few things worth mentioning:

1) Five Repeats: Each benchmark was run five times to avoid outliers and get the representative power behavior. We ran DGEMM and Stream tests before running VASP in the same job script to exclude the runs manifesting relatively larger manufactural differences in hardware devices. We selected the run with the minimum total runtime as a representative, which has a reduced chance of being affected by underperforming system components.

2) Power per Node and GPU: In this study, we focus on power usage per node and per GPU because our VASP benchmarks were designed to balance load between GPUs and nodes. However, as shown in Figure 1, individual nodes in a multi-node VASP job can have slight power variations, likely due to manufacturing differences in hardware. This is evident as identical DGEMM and STREAM runs exhibit similar power differences across nodes despite consistent performance. Even idle power varies slightly. A random check on 16 GPU nodes revealed idle power variability of up to 100 W (410 W to 510 W). These variations should be taken into account properly where applicable.

3) High Power Mode: We examined a variety of metrics to represent application power usage. In this paper, we use the high power mode (instead of mean power) and full width at half maximum (FWHM) of the high power mode to characterize application power usage and its distribution. We define the high power mode as the mode corresponding to the highest power (see Figure 2); and use the standard statistical definition of FWHM, which measures the width of the distribution at half of its maximum value. Compared to mean or maximum power, high power mode is a better metric for representing VASP's power usage, particularly for implementing power management strategies like power capping. This is because VASP's power timeline data is often multi-modal, making mean power less representative; meanwhile, maximum power may capture brief spikes that do not reflect overall usage. However, we also reference mean power where applicable, as it serves as a useful indicator of energy usage. To determine the high power mode, we utilize the kernel density estimate (KDE) plot of the power timeline data distribution.

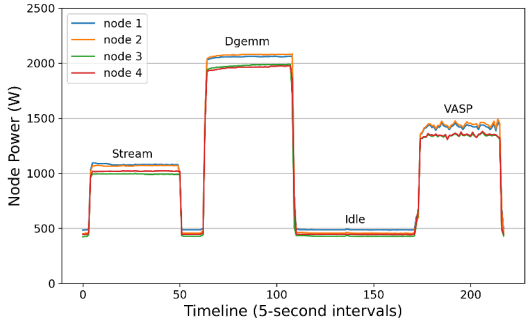

Fig. 1. The figure shows the power usage of individual nodes participating in a 4-node VASP job running Si256_hse benchmark. The job was run with Stream, Dgemm, and an idle phase before running VASP (right segment).

4) Sampling Rates: We explored the impact of sampling granularity of power measurements in our study. Figure 2 (left) illustrates the GPU power distribution and their maximum, median, and minimum power, as well as the high power mode (right), across different sampling rates. As the sampling rate increases, the full width at half maximum (FWHM) of the high power mode widens, but the high power mode itself remains unchanged. Additionally, the maximum power may slightly decrease, and some details of the timeline pattern may be lost. For instance, at a 10-second sampling rate, the second power mode is not detected, while at five seconds or finer, all three modes are visible. Figure 2 shows that any sampling rate up to 10 seconds is sufficient for capturing the high power mode and mean power. However, to effectively capture the power timeline behavior, a sampling rate of five seconds or finer is necessary, depending on the focus. Our power timeline data primarily have a 2-second intervals; even with occasional larger gaps, the interval did not exceed five seconds.

C. VASP Power Timeline Patterns

Figure 3 shows the power timelines for three selected VASP benchmarks, Si256_hse, GaAsBi-64, and Si128_acfdtr, running on a single node. Four additional benchmarks (CuC_vdw, PdO4, PdO2, B.hr_105) were omitted as their power timeline patterns are similar to one of those selected benchmarks.

As shown in Figure 3, power timeline patterns vary significantly across these benchmarks. For Si256_hse and GaAsBi-64, node power usage (represented by the black solid line) remains mostly flat with relatively small fluctuations, whereas Si128_acfdtr exhibits substantial variation. Notably, a significant portion of Si128_acfdtr's execution runs on CPUs (evidenced by the flat section in the middle of the timeline) due to VASP 6.4.1 not yet porting the exact diagonalization step to GPUs. For Si256_hse and Si128_acddtr, which consume high power, the four GPUs account for >70% of the total power usage, while the CPU and memory use less than 10%, with primarily flat power consumption during execution. GaAsBi-64, on the other hand, uses much less power, indicating

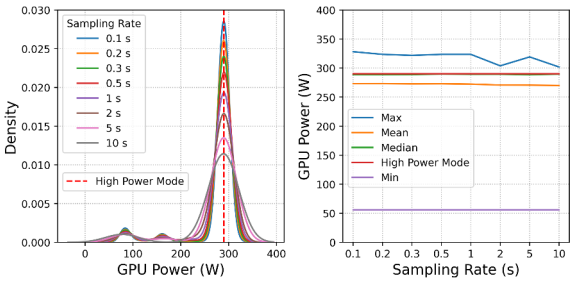

Fig. 2. Per GPU power distributions (left) and their maximum, median, minimum, and high power mode (right) at a range of sampling rates. We measured power data at a 0.1-second sampling rate and then down-sampled it to the rest of the sampling rates shown in the figure. The experiments were done with the Si256_hse benchmark using one node.

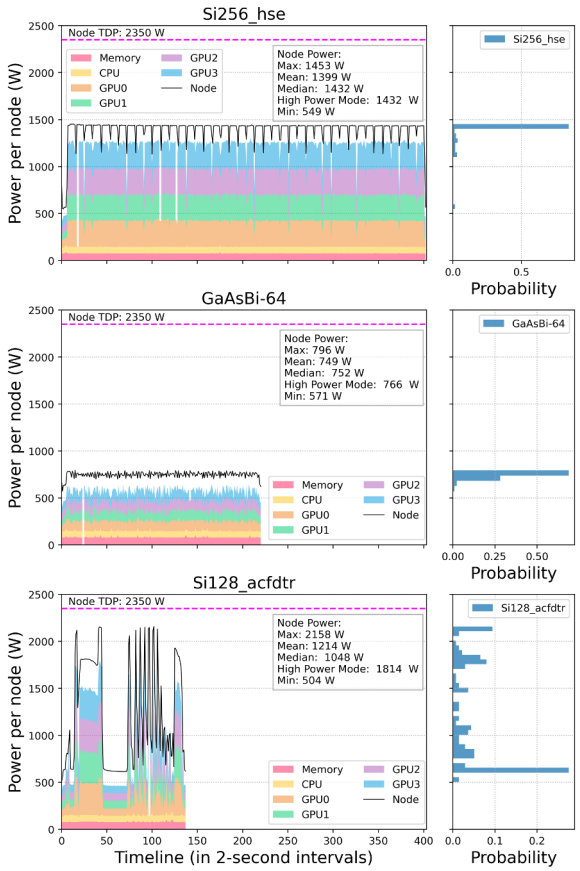

Fig. 3. Average power consumption of Perlmutter GPU node and its components when running VASP benchmarks, Si256_hse, GaAs64, and Si128_acfdtr (left panels) on a single node. The power timeline data was averaged over 2-second intervals. The gap between the total node power (black line) and the sum of the individual component powers likely arises from additional components within the node, such as network interface cards (NICs). The text box displays the maximum, median, and minimum node power and high power mode per node. A dashed magenta line indicates the node's TDP value. The histogram next to each benchmark power timeline shows the distribution of node power data.

insufficient workload to fully utilize the four GPUs in this benchmark.

The histograms next to each power timeline in Figure 3 show the power distribution for each benchmark, revealing non-normal and at least bimodal characteristics. High power mode per node ranges from 766 to 1814 W, with maximum power exceeding 2100 W. Notably, power usage per node, as measured by the high power mode, remains well below the node's TDP (indicated by the magenta dashed line).

D. Power Usage of Representative VASP Workloads

In this subsection, we present the power consumption data for all benchmarks selected to represent NERSC's VASP workloads. For each benchmark, varied node counts were used to simulate real-world scenarios, reflecting how users run jobs across a range of nodes with varying levels of parallel

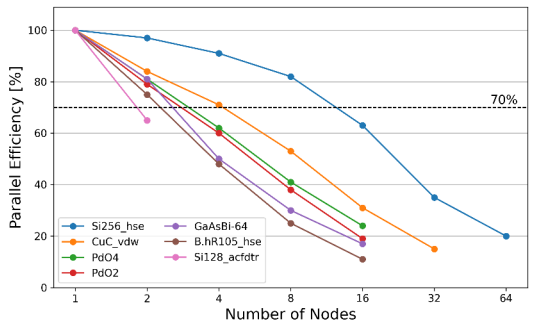

Fig. 4. Parallel efficiency of VASP.

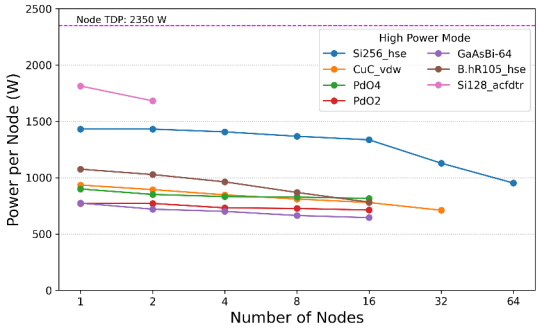

Fig. 5. Power usage of seven representative VASP workloads. The horizontal axis shows the number of nodes used, and the vertical axis shows the high power mode per node.

efficiency [2]. See Figure 4 for VASP's parallel efficiency at each node count. In general, parallel efficiency of 70% and up is recommended for optimal use of computing resources.

Figure 5 shows the high power mode per node for VASP when running each representative workload at different concurrency. We can see that the power changes with concurrency for a given workload are not significant, provided the code is run at a reasonable parallel efficiency level (70% and up); the power starts to drop visibly at parallel efficiencies below this level. However, the high power mode varies significantly with different workloads performing different types of computation (methods), ranging from 766 to 1810 W. Even within the same workloads, different sizes, and chemical elements significantly affect power consumption. For example, while the PdO4 (348 atoms) and PdO2 (174 atoms) benchmarks (the green and red lines, respectively) have identical chemical elements and unit cells, their size difference causes a power usage difference of more than 150 W per node. In another example, the B.hR105_hse benchmark is the same workload as Si256_hse but is smaller and contains different chemical elements. While both use more power than the other benchmarks performing basic DFT functional calculations (PdO2, PdO4, and GaAsBi-64), B.hR105_hse uses about 380 W less power per node than Si256_hse.

Figure 5 illustrates that many factors contribute to the power variations in VASP to varying degrees, resulting in a wide range of power usage.

IV. Decomposing VASP Power Variations

In this section, we analyze VASP's power dynamics under various conditions. We explore how system sizes, internal control parameters, concurrency, and methods influence VASP's power usage. Our experiments primarily use silicon supercells, allowing us to vary one condition at a time. While our focus is on power usage, we also present energy consumption data where relevant.

[2]Parallel efficiency is defined as S/N, where S is the speedup achieved when using N processors.

A. Power Changes with System Sizes

Figure 6 shows the high power modes for a single node and four GPUs within that node for silicon supercells with varying numbers of atoms. To isolate the effect of size, we fixed the method to the DFT functional calculation with the default iteration scheme. The results show that power usage increases with larger system sizes, reaching a plateau when the GPU power usage approaches its TDP. The figure indicates that approximately 2048 silicon atoms are needed to saturate the GPU resources for the DFT functional calculation.

As mentioned in Section III-A, the computational representations of system sizes are the number of plane waves (NPLWV) and the number of bands (NBANDS). As the number of atoms in the silicon supercells increases, both NPLWV and NBANDS also increase, with NPLWV ranging from 88,200 to 3,175,200 and NBANDS ranging from 164 to 5,764. Therefore, the power increase observed with increasing silicon supercell sizes is from the combined effect of the increased number of plane waves and bands. In the following subsection, we will examine the impact of plane waves and bands separately.

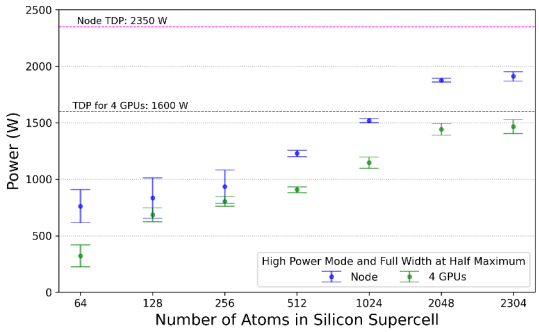

Fig. 6. VASP power consumption changes with system size. The high power mode is displayed per node (blue) and per four GPUs (green) within the node for silicon supercells with varying numbers of atoms. Error bars represent the full width at half maximum (FWHM) of the high power modes. The magenta dashed line marks the node's TDP, while the green dashed line indicates the combined TDPs of the four GPUs. The experiments used a single node.

B. Power Changes with Internal Parameters

VASP's internal parameters, such as the number of plane waves (NPLWV) and bands (NBANDS), can vary with system size and specific use cases. For instance, different accuracies (low, normal, or high) or cut-off energies can lead to varying numbers of plane waves for a given system. Similarly, users can select different numbers of bands to achieve better convergence or calculate optical properties.

Figure 7 shows how VASP's power usage changes with the number of plane waves (left), and the number of bands (right). The experiments were done with the Si256_hse benchmark on a single node using four GPUs. The figure shows that the high power mode remains constant when the number of bands changes but varies visibly when the number of plane waves changes. This observation aligns with VASP's parallelization strategy. VASP distributes bands across MPI processes (GPUs) while distributing plane waves to CUDA and tensor cores on each GPU. The bands assigned to each GPU are processed sequentially. Consequently, increasing the number of bands (NBANDS) per GPU extends the runtime, thereby consuming more energy (dashed line in Figure 8), but does not increase computational intensity (power). Conversely, increasing the number of plane waves (NPLWV) increases the computational intensity for each GPU due to the greater workload executed simultaneously, leading to higher power usage per GPU (and node). In this example, the GPU resources are not saturated at the reference plane-wave count (NPLWV=216000). If the GPU resources are saturated at the reference plane-wave count, no power increase would be observed, as seen in the silicon supercells with 2048 atoms and more in the previous subsection.

Notably, job concurrency remained unchanged when the number of plane waves or bands per GPU was adjusted in the above experiments.

C. Power Changes with Concurrency

Characterizing power changes with concurrency is essential for understanding VASP's power behavior. In a production setting, users may run jobs with varying node counts, either efficiently or inefficiently. VASP implements multiple levels of parallelism. The primary level distributes the number of orbitals (NBANDS) across MPI processes (GPUs), while the secondary level distributes the plane waves among the cores on each GPU. As seen in Table I, the parallelism explored in this study is well below the number of bands for each benchmark. Thus, increasing concurrency decreases the number of bands per GPU, while the number of plane waves in each band remains the same. As discussed in the previous subsection, increasing (or decreasing) the number of bands per GPU without altering the total concurrency does not impact VASP power usage. Thus, power consumption is expected to remain steady with changes in band count per GPU alone. However, as job concurrency increases, the added communication time could affect computational intensity (power) on the GPU, even though each GPU continues processing the same number of plane waves simultaneously.

Figure 8 shows the high power mode (left axis) of VASP and the total energy to the solution (right axis) at various concurrencies, using the Si256_hse benchmark as an example. The figure shows that as concurrency increases, VASP's power usage remains steady for a range of lower node concurrencies, especially when parallel efficiency stays within 70% or higher. This matches the expectation. However, with further increases in concurrency, power drops correspondingly due to increased communication time in VASP.

Figure 8 also shows that VASP's energy consumption increases monotonically with increasing concurrency.

D. Power Changes with Different Methods

Unlike some other codes, VASP includes numerous code paths within a single application binary, leading to varying power behaviors depending on the selected types of computation (methods). In this subsection, we examine the power behavior for seven distinct methods implemented in VASP. Among the seven methods, the HSE and ACFDTR methods are higher-order methods that are computationally more demanding. They also require more memory compared to their counterparts, the DFT functional calculation with various iteration schemes (those prefixed with dft in Figure 9).

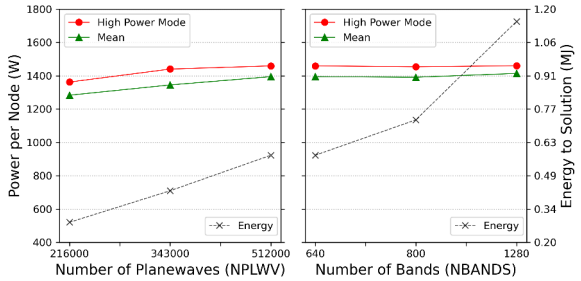

Fig. 7. VASP power consumption varies with internal parameters: the number of planewaves (left panel) and bands (right panel). The high power mode and mean power per node are shown on the left axis, while the energy to solution, measured in megajoules, is shown on the right axis. The experiments were done with Si256_hse using one node.

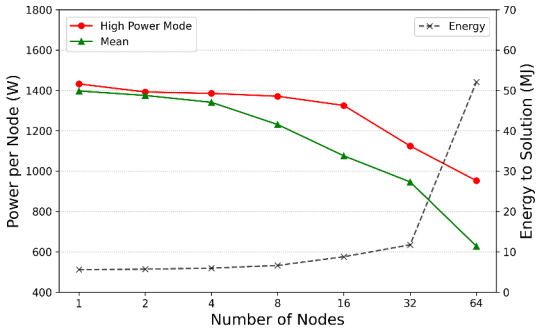

Fig. 8. VASP per node power usage (left axis) and the energy to the solution in megajoules (right axis) at various concurrency for benchmark Si256_hse.

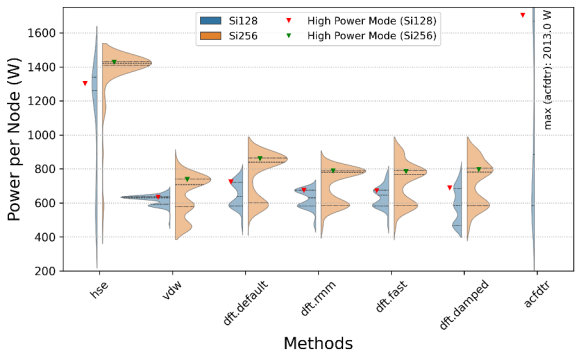

Fig. 9. VASP power usage varies with types of computations (methods). The vertical axis shows the high power mode per node. The violin plots with quartiles illustrate the multi-modal power distributions for the seven selected methods in VASP, applied to two silicon supercells containing 128 atoms (blue) and 256 atoms (orange). The tests were run on a single node.

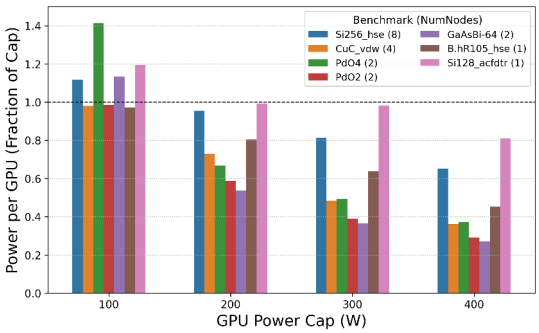

Fig. 10. Power consumed per GPU when running VASP under four different power caps: 400 W (default), 300 W, 200 W, and 100 W. The horizontal axis shows power caps applied to the GPUs, and the vertical axis shows the high power mode per GPU as a fraction of the applied cap. The dashed horizontal line represents the applied power cap. Each benchmark was run with a node count optimizing runtime while remaining above 70% parallel efficiency.

Figure 9 shows the high power modes for the seven selected methods. To minimize the moving parts in the comparison, we applied all these methods to the two silicon supercells with 128 and 256 atoms separately. The VDW method adds the van der Waals interaction corrections to the eigenenergy calculations, adding minor computational costs to the basic DFT calculations. So, we will treat it like other DFT methods in this subsection. The figure shows significant differences in power usage between the higher-order methods (HSE and ACFDTR) and the other DFT methods for both systems. Notably, the high power mode varies by more than 600 W per node on average.

Figure 9 also shows that when the supercell size increases, the power usage increases for all methods. This observation aligns with the power usage trend when increasing system sizes as discussed in Section IV-A.

V. Power Capping and Performance

To regulate VASP's power usage, we investigate VASP performance under varied power capping. This information is essential to optimize power allocation among jobs and distribute power where demand is more critical. While the DVFS [30] method is commonly employed for its ease of use, we chose to use power capping to control the device power, which is more efficient and accurate [31] in power control. Since the four GPUs on each Perlmutter node consume more than 70% of node power in VASP and are responsible for the power fluctuations during execution, we applied power capping on the GPUs. We used the NVIDIA System Management Interface (nvidia-smi) [32] tool to set various GPU power limits (using the -pl option) on the nodes.

A. Efficacy of Power Capping

The power range of the A100 40 GB GPU spans from 100 W to 400 W. We applied four different power caps, 400 W, 300 W, 200 W, and 100 W, to the GPUs allocated for VASP jobs. The default power limit on Perlmutter GPU nodes is 400 W, which is the A100 40 GB GPU's TDP. Figure 10 shows the high power mode per GPU as a fraction of the applied cap for seven representative VASP workloads. The dashed horizontal line represents the applied power cap. Bars falling below this line indicate that power usage remains within the applied power cap.

The efficacy of power capping in reducing power usage within the capped value is evident, except at the lowest allowed power cap of 100 W. At this cap, a larger error is observed, but it is not likely to be deployed in practice due to substantial performance degradation (see Section V-C).

B. Effect of Power Capping

Figure 11 illustrates the impact of power capping using the Si128_acfdtr benchmark as an example. The peak power is reduced by about 50%, while the troughs remain unchanged. This indicates power capping not only reduces power but also mitigates power variations within a job. Notably, the execution required higher power at 400 W is now visibly slowed down under the 200 W power cap.

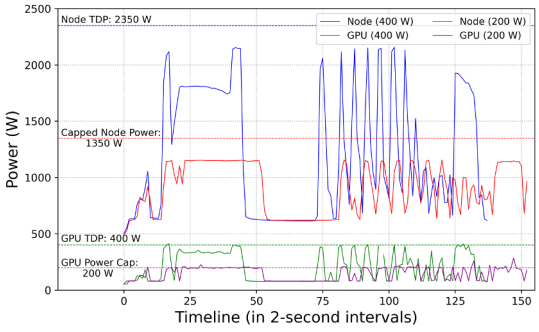

Fig. 11. Effect of GPU power capping on VASP. The power usage timeline for node and GPU 0, with and without a 200 W power cap, is shown on the vertical axis. Power timeline data is averaged over 2-second intervals. The experiment used the Si128_acfdtr benchmark on a single node.

C. Performance Response

Figure 12 shows the VASP performance response to different power caps. As shown in the figure, the performance of VASP is not affected when applying a 300 W power cap. However, at a 200 W power cap, we start to see a noticeable performance slowdown with the two most power-hungry benchmarks: Si256_hse and Si128_scfdtr (each by 9%). When the power limit is further reduced to 100 W, the slowdown is drastically increased, about 60% for both Si256_hse and Si128_acfdtr. Notably, the GaAsBi-64 and PdO2 benchmarks still have insignificant performance loss (<5%) even at the 100 W power cap.

We observed similar performance responses to power caps when running these benchmarks at different node counts, with the 1-node run being the most power-demanding. Figure 13 shows the VASP performance under different power caps when running Si256_hse at varied node counts. The performance is normalized at each node count relative to the default power limit. At all node counts, VASP responds to power caps similarly to its optimal node count: performance is unaffected at 300 W but drops 9% at 200 W, and decreases by over 60% at 100 W.

VI. Discussion

A. Implications

Further work is needed to accurately predict the power usage of VASP jobs. However, some of our findings can be used in power-aware scheduling to regulate this usage.

For example, VASP can run at only 50% of TDP with a less than 10% performance decrease, and the lower power-demanding jobs, DFT functional calculations, can run without visible performance loss at this power limit. The batch system can utilize this information to apply a 50% TDP power cap to VASP jobs and reallocate the spared power where the demand is more critical. The batch system can determine the workload type of VASP jobs in the queue without costly computation. Therefore, a scheduler can apply power capping decisions to VASP jobs within each scheduling cycle, usually 30 seconds.

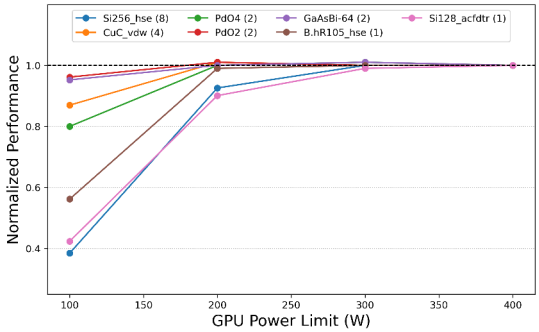

Fig. 12. VASP performance under GPU power caps. The horizontal axis shows the applied GPU power caps, and the vertical axis shows the performance normalized over the default power limit, 400 W, for the seven representative benchmarks. he number next to each benchmark name in the legend is the node count used to run the benchmark.

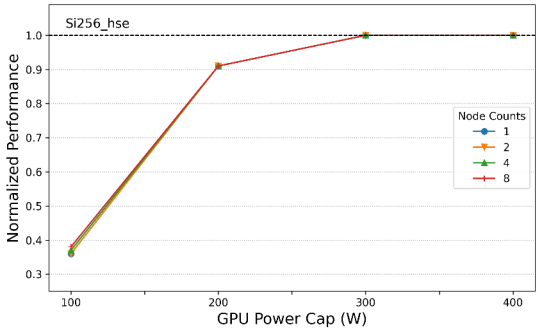

Fig. 13. VASP performance under GPU power caps when running Si256_hse at varied node counts. The horizontal axis shows the applied power caps, and the vertical axis shows the performance normalized at each concurrency relative to the default power limit, 400 W

B. Deployment strategies

Our deep dive into VASP power dynamics can extend to other applications. We plan to incrementally include additional prominent applications running at NERSC, especially those running at larger scales. Notably, the top 10 applications consume over 60% of the computing cycles, with VASP alone accounting for more than 15% of NERSC's computing cycles (as depicted in [33], [34]). Our approach has been recently applied to NERSC's second top application, MILC [35], which uses more than 12% of NERSC computing cycles by one of NERSC's summer students. By adopting this strategic approach, we aim to regulate power for a significant portion of the Perlmutter system. This will be achieved through the gradual implementation of power-aware scheduling, leveraging insights from application power profile analyses.

NERSC's workload is immensely diverse. While it is doable to deep-dive into a small number of top applications, this level of detailed study is not practical for all applications running at NERSC. These other workloads will necessitate a more statistical approach. To this end, in addition to this bottom-up approach (application by application), we also plan to explore top-down methods, such as deep learning techniques, to address the diverse workloads at NERSC. Ultimately, we expect to integrate these two approaches to achieve our power management goals with the desired accuracy.

C. Next Step - Predicting VASP Power

Our in-depth study on VASP power characteristics provides the basis for developing power prediction models. We have identified several key contributors to power variations, including system sizes (number of plane waves and bands), methods, and concurrency, and have assessed their varying impacts on power consumption. Moving forward, we will integrate our current findings into a comprehensive power model by quantifying each variation and extending our analysis to explore additional factors, such as chemical elements.

VII. Related Work

Several recent analyses of power use by leadership-class supercomputers have highlighted the cost of power for ever-growing HPC systems [14], [36], [37]. These observational studies belie a history of research into energy-efficient computing. A common focus is the use of Dynamic Voltage and Frequency Scaling (DVFS) to reduce the energy used by a job [4], [30], [38]–[45]. Others have examined and modeled the behavior of jobs running under a power cap [46]. While these studies mostly focused on the previous generation CPU and GPU systems, there are more recent studies investigating the power cap effect on modern GPUs at the system level [47]. However, system-level power caps leave optimization opportunities unexplored. Recent studies [48] comparing DVFS and power capping in the LLM space demonstrated best-use scenarios for each. Both DVFS and power-capping have the potential to impact performance negatively, and a variety of metrics have been proposed to quantify the energy/performance trade-off [49]–[51].

Power-aware scheduling based on application power profiles has been proposed for system-level power management [1]–[3]. Accurate estimates of the workload's power use are needed for these methods to be effective [7], [8], and approaches range from prediction-based [1], [52], to online measurement-based [2], to data-driven [53], [54].

Machine learning approaches for dynamic power management show promise, e.g., [55], but lack evaluation for production deployment. These methods must incorporate application-specific power behavior to accurately predict power usage.

This paper thoroughly investigates the power variations of an important HPC application in a production environment, demonstrating that application power consumption can strongly depend on input data that is not readily available to the scheduler. Our study aims to build a solid foundation for implementing power management strategies through power-aware scheduling based on application power profiles.

Acknowledgment

This work was supported by the Office of Advanced Scientific Computing Research in the Department of Energy Office of Science under contract number DE-AC02-05CH11231. This work used the resources of the National Energy Scientific Computing Center (NERSC) at the Lawrence Berkeley National Laboratory.

References

[1] M. Etinski, J. Corbalan, J. Labarta, and M. R. Valero, 'Optimizing job performance under a given power constraint in HPC centers', in Green Computing Conference, 2010 International, 2010, pp. 257–267.

[2] 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.

[3] M. Maiterth et al., 'Energy and Power Aware Job Scheduling and Resource Management: Global Survey — Initial Analysis', in 2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW), 2018, pp. 685–693.

[4] 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, 2005.

[5] Mantovani, F., & Calore, E. (2018). Performance and Power Analysis of HPC Workloads on Heterogeneous Multi-Node Clusters. Journal of Low Power Electronics and Applications, 8(2), 13. https://doi.org/10.3390/jlpea8020013

[6] T. Patel, A. Wagenhäuser, C. Eibel, T. Hönig, T. Zeiser and D. Tiwari, "What does Power Consumption Behavior of HPC Jobs Reveal? : Demystifying, Quantifying, and Predicting Power Consumption Characteristics," 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, 2020, pp. 799-809, doi: 10.1109/IPDPS47924.2020.00087.

[7] C. -H. Hsu et al., "Application Power Signature Analysis," 2014 IEEE International Parallel & Distributed Processing Symposium Workshops, Phoenix, AZ, USA, 2014, pp. 782-789, doi: 10.1109/IPDPSW.2014.90.

[8] Kenneth O'Brien, Ilia Pietri, Ravi Reddy, Alexey Lastovetsky, and Rizos Sakellariou. 2017. A survey of power and energy predictive models in HPC systems and applications. ACM Comput. Surv. 50, 3, Article 37 (June 2017), 38 pages. DOI: http://dx.doi.org/10.1145/3078811

[9] Zhengji Zhao, Ermal Rrapaj, Sridutt Bhalachandra, Brian Austin, Hai Ah Nam, and Nicholas Wright. 2023. Power Analysis of NERSC Production Workloads. In Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis (SC-W 2023), November 12–17, 2023, Denver, CO, USA. ACM, New York, NY, USA 9 Pages. https://doi.org/10.1145/3624062.3624200

[10] Anish Govind, Sridutt Bhalachandra, Zhengji Zhao, Ermal Rrapaj, Brian Austin, Hai Ah Nam, Comparing Power Signatures of HPC Workloads: Machine Learning vs Simulation, Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, November 2023, Pages 1890–1893, https://doi.org/10.1145/3624062.3624274

[11] 'Cori, a Cray XC40 system (decommissioned in May 2023)'. [Online]. Available: https://www.top500.org/system/178924/.

[12] Internal communication with Bhalachandra at NERSC (2021).

[13] 'Perlmutter, a HPE Cray EX system', 2023. [Online]. Available: https://docs.nersc.gov/systems/perlmutter/architecture/.

[14] E. Rrapaj, S. Bhalachandra, Z. Zhao, B. Austin, H. Nam and N. J. Wright, 'Power Consumption Trends in Supercomputers: A Study of NERSC's Cori and Perlmutter Machines', in ISC High Performance 2024 Research Paper Proceedings, 2024

[15] G. Kresse, G. Kresse, and J. Furthm, 'Efficiency of it ab initio total energy calculations for metals and semiconductors using a plane-wave basis set', Comput. Mater. Sci., vol. 6, 1996.

[16] VASP, an Ab initio electronic structure calculation code, https://www.vasp.at/wiki/index.php/The_VASP_Manual

[17] 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, 2019, p. 10.

[18] S. J. Martin and M. Kappel, 'Cray XC30 power monitoring and management', in Proceedings of Cray User Group Meeting, 2014, Lugano, Switzerland. Available Online:, 2014.

[19] A. Agelastos 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, 2014, pp. 154–165.

[20] S. Bhalachandra, 'Perlmutter OMNI Power Analysis Scripts', 2023. [Online]. Available: https://gitlab.com/NERSC/perlmutter-omni-analysis/.

[21] P. Hohenberg and W. Kohn, "Inhomogeneous Electron Gas" by (1964).

[22] W. Kohn and L. J. Sham, "Self-Consistent Equations Including Exchange and Correlation Effects" (1965).

[23] 'NVIDIA Collective Communications Library (NCCL)', NVIDIA Developer. [Online]. Available: https://developer.nvidia.com/nccl. [Accessed: 27-Mar-2024].

[24] M. Marsman, S. Maintz, A. Romanenko, M. Wetzstein, and G. Kresse, 'Porting VASP to GPU using OpenACC: exploiting the asynchronous execution modeli', in OpenACC Annual Meeting, 2020.

[25] Hybrid functionals https://www.vasp.at/wiki/index.php/Category:Hybrid_functionals#

[26] https://www.vasp.at/wiki/index.php/ACFDT/RPA_calculations

[27] https://www.vasp.at/wiki/index.php/Category:Exchange-correlation_functionals
[28] Z. Zhao, B. Austin, S. Maintz, and M. Mashman, 'VASP Performance on Cray EX Based on NVIDIA A100 GPUs and AMD Milan CPUs"', in Cray User Group meeting, 2023.
[29] 'Category:INCAR tag'. [Online]. Available: https://www.vasp.at/wiki/index.php/Category:INCAR_tag. [Accessed: 27-Mar-2024].
[30] 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, 2005.
[31] C. Imes and H. Zhang, 'Handing DVFS to Hardware: Using Power Capping to Control Software Performance', 2018.
[32] , NVIDIA System Management Interface, https://developer.nvidia.com/system-management-interface
[33] Brian Austin, *et. al* https://portal.nersc.gov/project/m888/nersc10/workload/N10_Workload_Analysis.latest.pdf
[34] Brian Austin, Perlmutter machine time breakdown by applications, 2024. https://portal.nersc.gov/project/m888/shared/perlmutter_machine_time_break_down_by_application_2024.pdf
[35] Fatih Acun, Zhengji Zhao, Brian Austin, and Nicholas Wright, 'Analysis of Power Consumption and GPU Power Capping for MILC,' to appear in the SC24 workshop proceedings of Sustainable Computing.
[36] 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.
[37] 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, 2021, pp. 1–14.
[38] 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, 2017, pp. 947–956.
[39] 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, 2015, pp. 911–918.
[40] 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.
[41] 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, 2005, p. 33.
[42] H. Kimura, M. Sato, Y. Hotta, T. Boku, and D. Takahashi, 'Emprical 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, 2006.
[43] 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, 2015, p. 3.
[44] 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.
[45] A. Tiwari, M. Laurenzano, J. Peraza, L. Carrington, and A. Snavely, 'Green Queue: Customized Large-scale Clock Frequency Scaling', in CGC '12: Proc. of the 2nd Intl. Conference on Cloud and Green Computing, 2012.
[46] 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), 2019, pp. 793–804.
[47] Dan Zhao, Siddharth Samsi, Joseph McDonald, Baolin Li, David Bestor, Michael Jones, Devesh Tiwari, and Vijay Gadepally. 2023. Sustainable Supercomputing for AI: GPU Power Capping at HPC Scale. In Proceedings of the 2023 ACM Symposium on Cloud Computing (SoCC '23). Association for Computing Machinery, New York, NY, USA, 588–596. https://doi.org/10.1145/3620678.3624793
[48] Pratyush Patel, Esha Choukse, Chaojie Zhang, Íñigo Goiri, Brijesh Warrier, Nithish Mahalingam, and Ricardo Bianchini. 2024. 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 '24), Vol. 3. Association for Computing Machinery, New York, NY, USA, 207–222. https://doi.org/10.1145/3620666.3651329
[49] R. Gonzalez and M. Horowitz, 'Energy dissipation in general purpose microprocessors', IEEE Journal of Solid-State Circuits, vol. 31, no. 9, pp. 1277–1284, 1996.
[50] A. J. Martin, M. Nyström, and P. I. Pénzes, 'ET 2: a metric for time and energy efficiency of computation', in Power aware computing, Springer-Verlag, 2002, pp. 293–315.
[51] S. I. Roberts, S. A. Wright, S. A. Fahmy, and S. A. Jarvis, 'Metrics for Energy-Aware Software Optimisation', in High Performance Computing, 2017, pp. 413–430.
[52] 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.
[53] 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.
[54] S. Wallace et al., '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, 2016, p. 56.
[55] A. M. Karimi, N. S. Sattar, W. Shin, and F. Wang, 'Profiling and Modeling of Power Characteristics of Leadership-Scale HPC System Workloads', Feb-2024. [Online]. Available: https://arxiv.org/pdf/2402.00729.pdf.

VIII. Appendix: Artifact Description/Artifact Evaluation

The Artifact Description materials for this paper, including the data and scripts used to generate the figures, are available at https://zenodo.org/records/13853298. Please note that VASP is licensed software, and therefore we are unable to provide the associated Artifact Evaluation materials.