Analysis of Power Consumption and GPU Power Capping for MILC

Fatih Acun

Zhengji Zhao

Brian Austin

Electrical and Computer Engineering
Boston UniversityAdvanced Technologies Group (NERSC)
Lawrence Berkeley National Laboratory
Berkeley, USAAdvanced Technologies Group (NERSC)
Lawrence Berkeley National Laboratory
Berkeley, USA
baustin@lbl.govBoston, USA
acun@bu.eduBerkeley, USA
zzhao@lbl.govBerkeley, USA
baustin@lbl.gov

Ayse K. Coskun Electrical and Computer Engineering Boston University Boston, USA acoskun@bu.edu Nicholas J. Wright Advanced Technologies Group (NERSC) Lawrence Berkeley National Laboratory Berkeley, USA njwright@lbl.gov

Abstract—Power has been a key constraint for supercomputers, and limitations on power become increasingly noticeable through the exascale era. Limited power availability pushes the facilities to operate under power constraints and develop power management methods, making it crucial to understand applications' power consumption behavior and their performance under power constraints. In this study, we examine the power consumption of MILC, a widely used lattice quantum chromodynamics application, on the Perlmutter GPU system at NERSC. We analyze the power consumption of *Generation* and *Spectrum* applications of MILC using varying parallel concurrencies and input sizes. We then investigate the performance under GPU power caps and show that MILC is well-suited for GPU power capping. Up to 50% of GPU's TDP can be applied to MILC jobs with less than 15% of performance decrease.

Index Terms—Application power consumption, GPU power capping, power management

I. INTRODUCTION

As high-performance computing (HPC) enters the exascale, power has become a major limiting factor to continue to advance in scientific computing. Many computing centers investigate power usage of the production workloads to operate their HPC systems efficiently under a prescribed power budget, as highlighted in several recent analyses of the power use on leadership-class supercomputers [1]–[3].

A major problem arising from the increased power consumption of HPC systems is power oversubscription due to the limitations of facility power infrastructures. Next-generation systems are expected to surpass their predecessors in terms of maximum power capacity, known as thermal design power (TDP). However, it is not always feasible to scale up the power infrastructure to match the system TDP, resulting in oversubscribed systems bottlenecked by power [4]. Another critical challenge for HPC systems is regulating their power consumption to be grid-aware, helping Independent System Operators manage the high stress on the grid. The emergency demand response programs are previously used to interact with HPC facilities in critical situations such as wildfires and extreme weather events to request certain power cuts [5], [6].

Given the challenges of limited power availability that HPC systems face, the need for effective power management methods is becoming increasingly crucial. One of the most important knobs to control the power is hardware power capping to regulate the power consumption of different hardware components. To use power capping for production workloads, it is essential to analyze the power consumption of applications and their performance under power caps.

While our goal is to reduce application power consumption to stay within a specified system power budget, we also aim to minimize power fluctuations that can negatively impact electrical grids and data center infrastructure. A recent analysis of NERSC's Perlmutter system [13] showed that 65% of the system's power variation is due to temporal variation in the power used by individual jobs [2]. This underscores the need for a detailed study of applications that utilize many nodes.

In this study, we provide a comprehensive analysis of power consumption for MILC, the 2^{nd} ranking workload [7] in terms of node hours and can run with a large number of nodes, on the Perlmutter GPU system [8] at NERSC. We analyze the power consumption of MILC with different input sizes and parallel concurrencies. We then extend our analysis to investigate the effects of GPU power capping on MILC's performance and energy consumption to identify power management opportunities. Our investigation shows that:

- The power usage of MILC has a strong dependence on the changing input sizes and parallel concurrencies showing up to 40% change in power use over GPU TDP.
- A power cap of up to half of the NVIDIA A100 GPU's TDP can be applied to MILC with less than 15% performance penalty and up to 28% reduction in energy consumption.

The rest of the paper continues with an overview of the related work in Section II, the system configuration in Section III. Section IV covers our power consumption analysis, followed by our GPU power capping results in Section V. Section VI concludes the paper.

II. RELATED WORK

Understanding the power consumption of HPC systems requires a multi-tier process that includes cluster-level and application-level analysis. Cluster-level analysis becomes critical for understanding the total power needs and developing cluster-level power management methods. CPU-based systems have been analyzed for their cluster, application, and userlevel power consumption [9]. Transitioning to the GPU-based computer architectures elevated the power consumption by HPC systems revealing different characteristics compared to CPU-only systems such as their lower utilization of system TDP [2].

Analyzing the application power consumption is crucial for understanding power performance relationships and implementing application-aware power management methods. Power capping of CPU workloads has been explored over a decade by analyzing the performance, and energy consumption [10], [11]. Power capping for GPU workloads become increasingly essential since computing power is now driven by GPU workloads, especially after the latest surge in generative AI. Power capping for AI workloads is explored by analyzing their performance and underlining the advantage of power capping on the reduced GPU temperatures [12]. Patel et al. analyze LLM training and inference workloads, finding that inference workloads use less peak power than training, allowing for 30% more hardware utilization within the same power budget through GPU power capping [13].

Our study thoroughly examines the power usage and variations of MILC, a key large-scale application. Combined with a recent similar analysis of VASP [14], a leading HPC application at many computing centers, we aim to establish a robust foundation for implementing power management strategies based on application power profiles.

III. SYSTEM CONFIGURATION

A. Perlmutter Supercomputer

We use Perlmutter at NERSC in this work. Perlmutter is an HPE Cray EX supercomputer with 3,072 CPU-only and 1,792 GPU-accelerated nodes. Each GPU-accelerated node contains one AMD EPYC 7763 "Milan" processor, 256 GB of DDR4 memory, four NVIDIA A100 "Ampere" GPUs, and four HPE Slingshot "Cassini" NICs. 256 of the GPU-accelerated nodes 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 for consistency over multiple experiments. Each of the GPU types has the same TDP of 400 W and the TDP of a 40 GB GPU node is 2,350 W, including 280 W for the CPU, 1600 W total for 4 GPUs, and 470 W for peripherals. Perlmutter has a total system TDP of 6.9 MW, including all the CPU and GPU nodes, service nodes, network routers, and cooling distribution units.

B. Application Level Power Measurement in Perlmutter

NERSC uses Operations Monitoring and Notification Infrastructure (OMNI) [15] for monitoring operational data related to power, performance, and other system metrics. Nodelevel power data collection is enabled through Cray's power monitoring architecture, making it possible to read power consumption by each component CPU, GPU, and memory. To collect job-specific data on Perlmutter, power consumption by the allocated nodes for the job is aggregated through Lightweight Distributed Metric Service (LDMS) [16] with a sampling rate of 1 HZ. We use the OMNI query scripts to access the job-level GPU power consumption data [17].

C. MILC NERSC-10 Benchmark

MILC is a lattice QCD application that models the strong interactions in subatomic physics with a discrete space-time model. The workflow of MILC includes two different stages, Generation and Spectrum. We refer to those stages as different applications of MILC in this work. The Generation application propagates the lattices until they sample an equilibrium distribution and the Spectrum uses the generated lattice to calculate the inversion of the staggered fermion matrix. The MILC code is implemented in C and employs MPI for parallelization. It uses the external QUDA library, which utilizes CUDA for GPU compatibility.

The NERSC-10 benchmarks are developed to estimate performance requirements for the next 5 years at NERSC based on the widely used production workloads [18]. The NERSC-10 benchmark for MILC provides a specific set of inputs and representative use cases for MILC's usage in production. In this study, we focus on analyzing the NERSC-10 benchmark for MILC.

IV. POWER CONSUMPTION ANALYSIS OF MILC

In this section, we present our analysis of power consumption for Generation and Spectrum applications of MILC. Our approach for power consumption analysis spans two aspects: (1) temporal analysis of the application power consumption to understand time-varying characteristics (e.g., fluctuations, periodicity), (2) distribution analysis to identify where the power data is concentrated.

We execute Generation and Spectrum application benchmarks with four inputs of varying sizes and three different parallel concurrencies for each input to capture the changing power behaviors for different configurations. Table I shows our experiment configurations. We repeat the run for each configuration three times to generate power data and use the one with the shortest execution time when reporting performance. To provide insight into power and energy consumption results in Section V, we present the parallel efficiency results upfront in Figure 1, which are calculated relative to the experiments with the smallest node allocation for each input category. In general, a parallel efficiency of 70% and up is considered a good use of computing resources.

 TABLE I

 EXPERIMENT CONFIGURATIONS FOR MILC GENERATION AND SPECTRUM



Fig. 1. Parallel efficiency results. The horizontal dashed line represents the 70% threshold for recommended parallel efficiency. Parallel efficiency is calculated as S/N, where S is the speedup achieved when using N processors.

A. Temporal Analysis of Power Consumption

In this section, we analyze the power consumption timelines for the application executions. Figure 2 shows the timelines of GPU, CPU, memory, and node power for Generation and Spectrum executions with tiny input on one node. By investigating timelines for all experiment configurations, we see similar behavior exists among our experiments with different input sizes and parallel concurrencies and provide these as representative examples.

We observe low power consumption periods, around 500 W node power corresponding to 20% of the node TDP, at the beginning and the end of each application. By profiling the application runs, we identify that low power periods refer to I/O operations for loading the input and writing the output. For the core calculation phases that start approximately after the first 15 seconds, node power reaches 1500-1600 W driven by the increase in the GPU power. For Generation, we observe a fluctuating behavior on node power, between 1650 W and 1250 W, that repeats approximately within 15 seconds. Profiling results show that those power drops correspond to CUDA memory operations and file writes. Due to these power drops, Generation jobs with large node allocations can cause cluster-



Fig. 2. Timeline of power consumption for Generation and Spectrum applications with tiny input executed on 1 node. The GPU timeline represents the total combined power consumption of all four GPUs within the node.

level power fluctuations. Spectrum applications demonstrate a steady power consumption at around 1600 W node power in the core calculation phase. For both applications, we see the total GPU power consumption by the four GPUs within the node partakes the most significant portion of the node power. Therefore, we focus on the GPU power consumption for the rest of our analysis.

B. GPU Power Distributions

A comprehensive analysis of GPU power distribution is essential for identifying key characteristics to see where the power data is located over the GPU power domain. We provide the GPU power distributions for all experiments with violin plots in Figure 3. We show the number of nodes used on the x-axis, and per GPU power on the y-axis, and use colors to represent different input sizes. The violins show the Kernel Density Estimation (KDE) of each distribution. We show the power value with the highest density with a diamond inside each violin to explicitly indicate the mode of the power distribution for the core phase execution (high power region) and refer to it as 'high power mode' for the rest of this paper.

In Figure 3, we see the power distributions show multimodal characteristics for each experiment configuration, pointing out the I/O operations for lower power regions and powerintensive calculations for high power regions. In addition, we see a decreasing trend for the GPU power distributions and their corresponding modes with the increased number of nodes within each input size, as expected due to decreased computation intensity per GPU.

Another significant takeaway from the power distributions is that power consumption does not reach the GPU TDP of 400 W. We observe that high power modes for all experiments are always below 320 W, which corresponds to 80% of the GPU TDP. This observation is useful for power provisioning to inform the power management methods to expect the per GPU power for MILC applications to be less likely to exceed 350 W.



Fig. 3. GPU power distribution for all experiments with Generation and Spectrum applications. The horizontal dashed lines show the GPU TDP. The diamonds in each violin show the high power mode value for each distribution.

V. GPU POWER CAPPING FOR MILC

Hardware power capping has various system-level and application-level impacts. Power capping affects application performance depending on the restrictiveness of the power budget and the application's sensitivity to power limits. However, there are multiple advantages such as improved energy and power efficiency and lower hardware temperatures [12].

In this section, we analyze the impact of power capping on the power and energy consumption of MILC. We also examine the performance trade-offs associated with different power capping levels. We use the NVIDIA System Management Interface tool (*nvidia-smi*) [19] to apply various power caps to GPUs using its *-pl* option. *Nvidia-smi*'s *-pl* option requires root privilege. NERSC implemented a local Slum plugin that allows end users to apply power limits using an sbatch flag, *--power-limit*, conveniently.

A. Power Usage Under Capping

The permissible range for power capping on A100 40 GB GPUs is 100 W to 400 W, with 400 W being the default setting on Perlmutter. Our experiments on power capping tested values of 400 W, 300 W, 250 W, 200 W, and 150 W.

Figure 4 shows the high power mode per GPU under each applied power cap at various node counts for Generation (top) and Spectrum (bottom). One can see the power requirements for these runs at the 400 W power limit. The efficacy of the power capping is evident, as shown in the figure that all measured power values are below their respective power caps. Figure 5 shows the power usage timeline for a Generation run with two nodes with and without 200 W power cap, sampled



Fig. 4. High power modes per GPU under GPU power caps.

among these runs. As noted in Section IV-A, those drops of the power timeline correspond to the memory operations in Generation, which are not affected by the applied power cap because their required power is below the applied cap 200 W (as shown in the GPU power timelines in Figure 5). Consequently, the troughs of the power timeline stay unchanged while the peaks in the power timeline are capped below the set limit. As a result, power capping not only reduces overall power consumption but also smooths out power variations.

B. Performance Under Power Capping

In this section, we analyze the performance of MILC applications and present our insights for performance-aware power caps. Figure 6 shows the performance sensitivities of applications for power caps relative to their uncapped execution times. Our takeaways for performance sensitivities under power caps are as follows:

Power Caps Above 250 W: For power caps 250 W or above, the performance slowdowns are considerably minor, rarely reaching up to 10%. Our power consumption distribution analysis in Figure 3 shows a significant portion of the application executions are not reaching 250 W and not getting affected by



Fig. 5. Effect of GPU power capping on MILC Generation. 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 small input of Generation running on two nodes.

the power cap. Even for those applications with GPU power distributions exceeding 250 W (e.g., tiny and small inputs up to 4 nodes), it is possible to reduce the power consumption using a 250 W power cap with a minimal performance penalty. **Significant Performance Slowdown at 150 W:** There is a significant performance slowdown at 150 W. While the performance impacts are relatively small at 200 W, power caps result in significant slowdowns up to 85% at 150 W.

Scaling up with Fixed Problem Size: Since per-GPU power consumption decreases with increased concurrency for a given input size, applications running with higher concurrency experience less performance loss under power caps than those with fewer nodes. These runs are indicated by green lines in Figure 6.

C. Energy Consumption Under Power Capping

Energy consumption is a crucial metric to analyze under power caps along with power and performance. We present the total GPU energy consumption by applications in Figure 7. Our insights for energy consumption are as follows:

Scaling up with Fixed Problem Size: In each input size for Generation and Spectrum, runs with the largest node allocations (shown in green lines in Figure 7) have the highest energy consumption due to their low parallel efficiency as presented in Figure 1. However, it is possible to save substantial amounts of energy up to 23% with a 150 W power cap since large runs of each input exhibit the least performance slowdowns.

Increased Energy at 150 W: We see the most significant reductions in energy consumption up to 28% with a 200 W power cap. Applying a 150 W power cap, especially for the runs with fewer nodes (e.g., less than 4 nodes) increases the energy consumption compared to a 200 W power cap. As discussed in the performance analysis, a 150 W power cap leads to significant slowdowns, which in turn result in higher energy consumption due to longer execution times.

VI. DISCUSSION

In this study, we present a comprehensive analysis of the power consumption and GPU power capping for MILC using various input sizes and parallel concurrencies. Our investigation shows that applying a power cap of up to 50% of the A100 GPU's TDP to MILC jobs results in less than 15% performance penalty. Parallel to our work, a deep-dive study on VASP revealed quite different power characteristics but also showed that up to 50% of A100 GPU's TDP can be applied to most VASP workloads with less than 10% performance loss [14]. These findings can be used in the power-aware scheduler to regulate the power consumption of a significant portion of the Perlmutter system. Moving forward, we plan to build a comprehensive power prediction model for MILC based on our study to deploy in power-aware scheduling along with VASP and other prominent applications on Perlmutter.

ACKNOWLEDGMENT

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a Department of Energy Office of Science User Facility using NERSC awards ASCR-ERCAP0026875 and ERCAP0026397.

REFERENCES

- A. M. Karimi, N. S. Sattar, W. Shin, and F. Wang, "Profiling and modeling of power characteristics ofleadership-scale hpc system workloads," Feb. 2024.
- [2] E. Rrapaj, S. Bhalachandra, Z. Zhao, B. Austin, H. A. 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 (39th International Conference), pp. 1–10, 2024.
- [3] Z. Zhao, E. Rrapaj, S. Bhalachandra, B. Austin, H. A. Nam, and N. Wright, "Power analysis of NERSC production workloads," in *Proceedings of PMBS23*, 2023.
- [4] 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*, p. 173–182, 2013.
- [5] J. Kwan, "Climate change threatens supercomputers," Science (New York, NY), vol. 378, no. 6616, pp. 124–124, 2022.
- [6] V. Mehra and R. Hasegawa, "Using demand response to reduce data center power consumption — google cloud blog," Oct 2023.
- [7] B. Austin, "Perlmutter machine time breakdown by applications (2023)," https://ur0.jp/EuRqo, 2023. Accessed: 2024-08-12.
- [8] "Perlmutter architecture documentation." https://docs.nersc.gov/systems/ perlmutter/architecture/. Accessed: 2024-08-05.
- [9] 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," in 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 799–809, 2020.
- [10] 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), pp. 793–804, 2019.
- [11] O. Sarood, A. Langer, L. Kalé, B. Rountree, and B. de Supinski, "Optimizing power allocation to cpu and memory subsystems in overprovisioned hpc systems," in 2013 IEEE International Conference on Cluster Computing (CLUSTER), pp. 1–8, 2013.
- [12] D. Zhao, S. Samsi, J. McDonald, B. Li, D. Bestor, M. Jones, D. Tiwari, and V. Gadepally, "Sustainable supercomputing for ai: Gpu power capping at hpc scale," in *Proceedings of the 2023 ACM Symposium* on Cloud Computing, SoCC '23, (New York, NY, USA), p. 588–596, Association for Computing Machinery, 2023.
- [13] P. Patel, E. Choukse, C. Zhang, I. n. Goiri, B. Warrier, N. Mahalingam, and R. Bianchini, "Characterizing power management opportunities for llms in the cloud," in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, ASPLOS '24, (New York, NY, USA), p. 207–222, Association for Computing Machinery, 2024.
- [14] Z. Zhao, E. Rrapaj, S. Bhalachandra, B. Austin, and N. Wright, "Understanding VASP power profiles on NVIDIA A100 GPUs," in *Proceedings* of *PMBS24*, 2024.
- [15] 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 *Workshop Proceedings of the 48th International Conference on Parallel Processing*, ICPP Workshops '19, (New York, NY, USA), Association for Computing Machinery, 2019.
- [16] A. Agelastos and et. al, "The lightweight distributed metric service: A scalable infrastructure for continuous monitoring of large scale computing systems and applications," in *Proceedings of SC'14*, pp. 154– 165, 2014.
- [17] NERSC, "Perlmutter omni-path analysis." https://gitlab.com/NERSC/ perlmutter-omni-analysis, 2024. Accessed: 2024-08-05.
- [18] NERSC, "NERSC-10 benchmarks." https://www.nersc.gov/systems/ nersc-10/benchmarks, 2024. Accessed: 2024-08-05.
- [19] NVIDIA Corporation, NVIDIA System Management Interface. NVIDIA Corporation, 2023. Version 12.535.0.



Fig. 6. Power sensitivity for the performance of Generation and Spectrum applications with different input sizes and parallel concurrencies. Execution time is measured relative to the uncapped time at 400 W.



Fig. 7. Total GPU energy consumption for Generation and Spectrum applications with different input sizes and parallel concurrencies.