Not All GPUs Are Created Equal: Characterizing Variability in Large-Scale, Accelerator-Rich Systems

Prasoon Sinha, Akhil Guliani, Rutwik Jain, Brandon Tran, Matthew D. Sinclair and Shivaram Venkataraman
Computer Sciences Department, University of Wisconsin-Madison
Madison, United States of America
Email: {psinha9, guliani, rnjain, bqtran2}@wisc.edu, {sinclair, shivaram}@cs.wisc.edu

Abstract—Scientists are increasingly exploring and utilizing the massive parallelism of general-purpose accelerators such as GPUs for scientific breakthroughs. As a result, datacenters, hyperscalers, national computing centers, and supercomputers have procured hardware to support this evolving application paradigm. These systems contain hundreds to tens of thousands of accelerators, enabling peta- and exa-scale levels of compute for scientific workloads. Recent work demonstrated that power management (PM) can impact application performance in CPU-based HPC systems, even when machines have the same architecture and SKU (stock keeping unit). This variation occurs due to manufacturing variability and the chip’s PM. However, while modern HPC systems widely employ accelerators such as GPUs, it is unclear how much this variability affects applications. Accordingly, we seek to characterize the extent of variation due to GPU PM in modern HPC and supercomputing systems. We study a variety of applications that stress different GPU components on five large-scale computing centers with modern GPUs: Oak Ridge’s Summit, Sandia’s Vortex, TACC’s Frontera and Longhorn, and Livermore’s Corona. These clusters use a variety of cooling methods and GPU vendors. In total, we collect over 18,800 hours of data across more than 90% of the GPUs in these clusters. Regardless of the application, cluster, GPU vendor, and cooling method, our results show significant performance variation even though the GPU architecture and vendor SKU are identical within each cluster, with outliers up to 1.5× slower than the median GPU. These results highlight the difficulty in efficiently using existing GPU clusters for modern HPC and scientific workloads, and the need to embrace variability in future accelerator-based systems.

Index Terms—Accelerator architectures; Dynamic voltage scaling; Power measurement; Temperature measurement; Time measurement

I. INTRODUCTION

Recently, domain scientists have leveraged the massive parallelism of accelerators for scientific discovery. Some of these discoveries use machine learning (ML) or deep learning (DL) for image recognition [1], [2], speech recognition [3–5], and machine translation [6], [7]. Scientists have also used accelerators in other areas, including molecular dynamics, material science, and quantum chemistry. Since these applications often require peta- or exascale levels of compute, running them on massively parallel systems has yielded promising results in areas including protein folding [8], plasma reactor status prediction [9], material science [10], and SARS-CoV-2 [11].

Accordingly, supercomputers, datacenters, and computing centers have procured hardware to accommodate this evolving application paradigm. To reach exascale levels of compute, many of these systems use many accelerators, which offer greater power efficiency and thus support emerging AI and HPC workloads within a constrained power budget. For example, nearly all of the top 10 supercomputers leverage GPUs [12] and the second-ranked supercomputer, Oak Ridge National Lab’s (ORNL’s) Summit, has over 27000 GPUs. Compute centers such as NCSA Delta (840 GPUs), SDSC Expanse (216 GPUs), and Texas Advanced Computing Center (TACC, 744 GPUs) also utilize many GPUs. Similarly, Microsoft, Tesla, and others have deployed accelerator-based supercomputers [13], [14]. The upcoming Aurora, El Capitan, and Frontier supercomputers [15] are expected to have even more GPUs. Thus, current exascale computing systems contain hundreds to tens of thousands of GPUs, and future systems will likely be comprised of a large variety of accelerators and customized chips [13], [14], [16]–[23].

Despite their power efficiency, exascale systems have an enormous power footprint, making it important to consider the hardware’s power management (PM) algorithms. PM can lead to power and frequency variations across nodes. Such dynamic behavior makes it challenging for repeatable, high performance and can lead to resource underutilization. For example, recent work studying CPU-based supercomputers showed that PM impacts application performance by up to 20%, even for CPUs with the same architecture and vendor SKU (Stock-Keeping Unit) [24]–[29]. This variation occurs due to the manufacturing process and the chip’s power constraints [26], [30]. However, despite their increasingly widespread use in modern HPC systems, there is limited work that examines how accelerator PM and manufacturing variability affects application performance. For example, while Scogland, et al. [30] conducted an AMD GPU variability study in 2015,
they only used one benchmark and called for a more in-depth study. We discuss related work further in Section II.

In this paper we perform a rigorous study to understand GPU variability in large scale, accelerator-rich computing clusters. While some systems [13], [14] utilize FPGAs or other accelerators, we focus on GPUs since the majority of the Top-500 systems utilize GPUs. To ensure coverage across GPU workloads that are run on modern systems and to stress different GPU components, we select five applications from ML, sparse graph analytics, and molecular dynamics. Furthermore, we study multiple computing clusters to examine how different scales (hundreds to tens of thousands of GPUs), cooling (air, mineral oil, and water), and GPU vendors (AMD and NVIDIA) impact variability. Specifically, we study five modern compute clusters: ORNL’s Summit, Sandia National Lab’s (SNL’s) Vortex, TACC’s Frontera & Longhorn, and Lawrence Livermore National Lab’s (LLNL’s) Corona. We perform measurements on over 90% of the GPUs on each cluster and verify that our methodology is statistically significant [31]. In total, we collected and analyzed over 18,800 hours of data to study variability in performance, temperature, GPU frequency, and power consumption. Our findings include:

- All clusters exhibit significant performance variability when running SGEMM (Figure 1), which is normalized to a median runtime of 1. On average, SGEMM performance varies between 7%-9% across GPUs in the same cluster, with outliers up to 1.5× slower than the median GPU.
- Performance variability is larger for compute intensive workloads such as matrix multiplication (SGEMM) or training ML models like ResNet-50 (22%, max 3.5× slower than median GPU) than memory intensive workloads like PageRank (1%). BERT pre-training, which has a mix of compute and memory-bound operations, has 8% performance variation. This variation impacts resource utilization, increases runtime, and hurts responsiveness, especially for multi-GPU experiments.
- Air cooled clusters (Longhorn, Corona) have a large temperature range ≥ 30°C and very high temperature nodes can have severe power throttling (>15% of TDP). Water and mineral oil cooling reduce temperature variation, but do not help performance and power variation.
- On the largest GPU cluster we study (Summit), there are a number of power outliers that are not correlated with temperature or frequency variation.
- Our results are consistent across different days of the week and times of day, indicating that the variability is not transient. Moreover, we also validated that similar or higher variability occurs when limiting the GPU power to be lower than the TDP (Section VI-A).

Overall our results indicate that significant power and performance variation exists in GPU-rich systems. Given there are no standards to expose PM information to administrators, it is difficult today to identify aberrations within the cluster that might require maintenance, modifications, or even replacement of GPUs. Further, unlike CPU-based systems, GPU nodes cannot be assigned as a desktop or a submit node to minimize impact. Thus, our results motivate systematic benchmarking across nodes to provide an early-warning for system administrators to perform maintenance or investigate bad GPUs, without hurting long-term cluster performance. Finally, our work also highlights the need to improve visibility of accelerator variability and design mitigation techniques in runtime frameworks for both users and system administrators.

II. BACKGROUND AND RELATED WORK

A. Hardware Performance Variability

As large scale datacenters and supercomputers are typically power-constrained, manufacturing variability between two identical architectures translates to varying performance and causes load imbalance even for perfectly balanced workloads. PM and temperature variations can also cause thermal throttling, possibly degrading performance further. To mitigate variability, CPU-based HPC systems have adopted various techniques including: dynamic load balancing algorithms [32], [33], cooling mechanisms and temperature-aware job placement [34]–[36], and intelligent adaptive runtimes [24]–[26].

However, less work has examined how GPU PM and manufacturing variability affect applications. Coplin, et al. [37] and Jiao, et al. [38] demonstrated that memory- and compute-bound GPU workloads exhibit significant differences in performance and energy for a variety of configurations. Jiao, et al. also found that increasing frequency often increased both power consumption and performance in Fermi-class GPUs. Moreover, they demonstrated that GPUs utilize dynamic voltage and frequency scaling (DVFS) to help stay within their power limit. However, both studies were conducted on GPUs several generations older than the ones in use today and focus on performance variability for a single-GPU workstation. Similarly, Scogland, et al. examined how LINPACK’s performance varied for CPU clusters and AMD GPUs [30]. However, this study was done in 2015, only used a single benchmark, and motivates the need for a more in-depth study of GPU variability. Thus, while these papers are a useful foundation, characterization is needed to determine the impact of PM on modern GPU-rich systems for representative workloads.

Other recent work has also examined how the Titan supercomputer’s thousands of GPUs behaved over the machine’s lifetime [39]. However, this work focused on issues like reliability, and did not investigate the impact of PM.

B. GPU Power Management

Current many-GPU systems use a local-only PM setup: each GPU has a given thermal limit it must stay within (its thermal design power, TDP), which is 300W for the NVIDIA V100 GPUs and AMD MI60 GPUs we study [40], [41]. The GPU PM controller varies the GPU’s voltage and frequency using DVFS to avoid exceeding its TDP [42], [43]. Although GPU vendors such as AMD and NVIDIA have not disclosed details about their DVFS schemes (or PM controllers), prior work has shown that, similar to multi-core CPUs, DVFS adjusts the
GPU’s streaming multiprocessor (SM) and memory frequencies and voltages to stay within the TDP and reduce energy consumption. Sometimes DVFS inadvertently compromises performance [42]–[44]. Thus, PM can affect the performance of GPU applications. Moreover, there is no global PM strategy across GPUs in a cluster. As a result, GPUs can be also affected by the behavior of other GPUs on the same node.

### III. Methodology

To ensure that our results were representative across a variety of GPU clusters with different properties, we sought to run experiments across clusters of different sizes, cooling approaches, and GPU vendors. Moreover, we also selected benchmarks that are representative of how modern systems are used and which stress a variety of GPU components.

**Cluster Parameters:** Table I summarizes the five unique HPC clusters we studied. TACC’s air-cooled Longhorn cluster [45] has 416 GPUs split across 104 nodes, each with 4 NVIDIA V100 GPUs. TACC’s mineral oil cooled Frontera cluster [45] is similar in size to Longhorn but has NVIDIA Quadro RTX 5000 GPUs. SNL’s water-cooled Vortex cluster [46] contains 216 NVIDIA V100 GPUs split across 54 nodes. To examine variability in a larger cluster, we also studied ORNL’s water-cooled Summit supercomputer [47] (27648 NVIDIA V100 GPUs). Finally, to examine variability across GPU vendors, we studied AMD Mi60 GPUs in Livermore’s Cora cluster [48].

Although we do not have access to Vortex’s temperature setup, the Summit and TACC GPU’s shutdown, slowdown, max operating, and max memory operating temperatures are: 90°C, 87°C, 83°C, and 85°C, respectively. The Corona GPU’s shutdown, slowdown, and max memory operating temperatures: are 105°C, 100°C, and 99°C, respectively [50], [51]. Finally, the Frontera GPU’s shutdown, slowdown, and max operating temperatures are: 96°C, 93°C, and 89°C, respectively. Throughout our experiments, we did not observe GPUs exceed these thresholds.

**GPUs:** Although AMD and NVIDIA have recently released newer GPUs than the ones available in the systems we study, we chose to study V100’s and Mi60’s because they are widely used in modern HPC systems. Prior work recommended pinning frequency, power limits, fan speed, and voltage ID to ensure that all GPUs have the same initial state (from the software and user perspective) [31]. However, Volta V100 GPUs do not have fans [52]. Instead, like many server-class GPUs, they are attached to large heatsinks and rely on server chassis fans to pull air across them for cooling. As a result, it is not possible to adjust their fan speed (e.g., using nvidia-smi) like desktop-class GPUs. Moreover, as we do not have administrative privileges on the clusters, we could not pin the frequency, power limit, or voltage ID. Instead, we verified that all GPUs were configured to the maximum frequency and power limit: 1530MHz and 300W for the V100s and 1800MHz and 300W for the Mi60s, respectively. We believe this setup is still relevant because it demonstrates how much variability a common HPC user, who does not have these privileges, would see. Moreover, we observed similar variability on a smaller NSF CloudLab cluster where we had administrator privileges and could pin frequency and power (Section VI-B).

**Workloads:** We selected four applications from parts of CORAL-2 [53], a collection of independent benchmarks used to measure exascale workload performance. To ensure that our workloads stress different GPU components and thus provide a holistic view of variability, we chose applications that are compute-bound, memory-bound, or balanced [37], [54], [55]. Table II summarizes the applications. SGEMM is a compute-intensive matrix-multiply kernel that is widely used in a number of workloads, including ML. ResNet-50 is a popular, multi-GPU, compute-intensive ML training workload. BERT is a popular, multi-GPU Transformer-based model that is part of the of the widely used MLPerf training benchmark suite. LAMMPS is a popular scalable scientific computing application that is memory-bound for the configuration we used. Finally, we tested another memory-bound workload, PageRank, which is used in the Havoq graph analytics benchmark. To ensure we study both single GPU and multi-GPU runs, we run LAMMPS, PageRank, and SGEMM on a single GPU while ResNet and BERT are run across multiple GPUs. To evaluate variability across clusters, we ran SGEMM on all clusters; we ran all other workloads on TACC’s Longhorn cluster. We provide further details about each application’s setup in their corresponding Results & Analysis sections. Collectively, analyzing these workloads helps us gauge whether GPU variability is application-specific or not in HPC systems.

**Measurement:** We collected four metrics for the duration of each application: kernel runtime (iteration runtime for ResNet-50 and BERT) in milliseconds (ms), GPU CU/SM temperature (°C), GPU CU/SM power consumption (Watts), and GPU CU/SM frequency (MHz). To collect this data we use the GPU vendor’s profilers [61], [62]. As in prior work [31], we computed the recommended sample size (num-

| Cluster       | GPU     | # GPUs | # Nodes | Cooling     |
|---------------|---------|--------|---------|-------------|
| Longhorn [45] | V100    | 416    | 104     | air         |
| Frontera [45] | RTX 5000| 360    | 90      | mineral oil |
| Vortex [46]   | V100    | 216    | 54      | water       |
| Summit        | V100    | 27648  | 4608    | water       |
| Corona [48]   | Mi60    | 328    | 82      | air         |

**TABLE II: Summary of applications studied on HPC clusters.**

| Benchmark | Input Size                          | Clusters Observed | Collection Duration |
|-----------|-------------------------------------|-------------------|---------------------|
| SGEMM [56]| 25536 x 25536                       | Longhorn          | 6 Weeks             |
|           | 25536 x 25536                       | Summit            | 8 Weeks             |
|           | 24576 x 24576                       | Corona            | 2 Weeks             |
| ResNet-50 [57]| Train. Set: 1.2M images              | Longhorn          | 2 Weeks             |
|           | Batch size: 64                      |                   |                     |
| BERT [58] | Train. Set: 30K words               |                   |                     |
|           | Batch size: 64                      | Longhorn          | 1 Week              |
| LAMMPS [59]| x, y, z = (8, 16, 16)                | Longhorn          | 2 Weeks             |
| PageRank  | 643994 x 643994                     | Longhorn          | 2 Weeks             |

| Benchmark | Input Size                          | Clusters Observed | Collection Duration |
|-----------|-------------------------------------|-------------------|---------------------|
| SGEMM [56]| 25536 x 25536                       | Longhorn          | 6 Weeks             |
|           | 25536 x 25536                       | Summit            | 8 Weeks             |
|           | 24576 x 24576                       | Corona            | 2 Weeks             |
| ResNet-50 [57]| Train. Set: 1.2M images              | Longhorn          | 2 Weeks             |
|           | Batch size: 64                      |                   |                     |
| BERT [58] | Train. Set: 30K words               |                   |                     |
|           | Batch size: 64                      | Longhorn          | 1 Week              |
| LAMMPS [59]| x, y, z = (8, 16, 16)                | Longhorn          | 2 Weeks             |
| PageRank  | 643994 x 643994                     | Longhorn          | 2 Weeks             |
number of GPUs) for each cluster to obtain \(\lambda = 0.5\%\) accuracy for average power within a 95% confidence interval. Given that we sample measurements from almost all GPUs in each cluster, our sample size is 2.9 times larger than the worst-case recommendations. Thus, our measurements are statistically significant. Unless otherwise specified, we use the median of each measurement to avoid one-off outliers. Since 1 ms is the minimum sampling interval for these profiler, we configured our input sizes to ensure kernel durations were larger than 1 ms. To characterize applications as compute- or memory-intensive, we also collected profiler metrics related to FU utilization (the utilization level of floating-point arithmetic functional units measured by nvprof on a scale of 0 to 10), DRAM utilization, and stalls. Because all the clusters used for data collection are shared, we collected measurements while other machines were in use. However, for consistent measurement we ensured there was no timesharing of our allocated nodes or GPUs during data collection. By using exclusive allocations and staggered run times, we eliminated spatial and temporal effects on variability for all applications. We discuss spatial effects further in Section VII. Finally, to protect against transient effects we collected data for multiple runs on the same machine over multiple days or weeks.

**IQR & Variability:** We use box and whiskers plots and inter-quartile regression (IQR) to help determine variation and categorize statistically significant outliers. Each box plot represents the spread from quartile 1 to 3 \((Q1 to Q3)\). The center of the box represents the median \((Q2)\). With \(IQR = Q3 − Q1\), the upper and lower whiskers in the plot represent \(Q3 + 1.5IQR\) and \(Q1 − 1.5IQR\), respectively. The IQR captures 99.3% of the Gaussian distribution within the box-and-whisker format. We define range as the difference between the upper and lower whiskers. We denote the variation of a metric as \(\frac{range}{Q^2}\) and denote all data points that fall outside the whiskers as outliers. Thus, outliers are not included in our variance calculations.

**IV. VARIATION ACROSS CLUSTERS**

**A. Methodology for SGEMM Application**

We use SGEMM to study variability across clusters. Our SGEMM application is a single SGEMM kernel that performs matrix multiplication on two matrices containing single-precision floats. To do this, we use optimized SGEMM implementations from NVIDIA’s cuBLAS [56] and AMD’s hipBLAS [63] libraries for the respective GPUs. To study the effects of the GPU’s PM controller, it is important that all streaming multiprocessors (SMs) or compute units (CUs) are fully occupied and the work is evenly distributed across SMs/CUs. Moreover, because the PM controller relies on DVFS to maintain the power limit for safe operation [43], [64], the kernel run must be long enough for the DVFS controller to reach a stable state [37]. Thus, we carefully tuned the matrix size (Table II) for both NVIDIA V100s and AMD MI60s to (i) achieve a sufficient runtime (Section III), (ii) ensure high performance, and (iii) provide high SM/CU occupancy.

We define 1 run of our experiment as 100 repetitions of the SGEMM kernel. The repetitions help avoid statistical bias and transient effects. Before collecting data, we run one warm-up run to avoid counting cuDNN startup overheads [65], [66]. We provide the exact same matrix inputs to every GPU, since the data inputs themselves are not important in our study.

**B. SGEMM on TACC Longhorn**

Figure 2 shows aggregated box plots for kernel duration, frequency, power, and temperature for SGEMM on Longhorn. Overall, SGEMM has 9% performance variation on Longhorn. Moreover, despite verifying that the GPUs are configured to run at the highest frequency (1530 MHz), Figure 2 shows that the GPUs run at lower frequencies (1300-1440 MHz). The overall frequency variation is 140 MHz or 11%. We also observe a wide spread in temperatures (33°C between Q1 and Q3). Furthermore, certain GPUs also run at power levels far below 300W: at around 250W. This significant variation in performance, power, and temperature motivated us to further investigate the relationship between these metrics.

Figure 3 presents scatter plots between the distinct measurement pairs. In Figure 3, although sometimes the slowest kernels run on the GPUs at the highest temperatures – and the fastest kernels run on the lowest temperature GPUs – overall performance and temperature are not strongly correlated \((\rho = 0.46)\). For example, multiple GPUs running at the same temperature have up to a 200 ms (10%) performance difference. Additionally, the top-right of the performance vs temperature plot shows a cluster of GPUs with both high temperature and longer runtimes. This implies that if a GPU is running at a high temperature (>78°C), it is likely to have worse performance. However, this is not always the case, because other GPUs run at similar temperatures but complete much faster (e.g., c004). This is surprising because as frequency increases, temperature is expected to increase. Moreover, usually higher frequencies improve SGEMM’s performance [30], [43].

**Takeaway 1:** Despite running the same kernel on similarly configured GPUs, we see 9% performance variance across GPUs in the Longhorn cluster, with temperature range \(\geq 30^\circ C\) and some power outliers at 250W.

**C. SGEMM on ORNL Summit**

We next examine the larger Summit to see if similar patterns are seen. As shown in Table I, Summit contains 27648 GPUs and is water cooled. We collect the same measurements for SGEMM as Longhorn. However, since Summit has so many

\(^1\)Although not shown due to space constraints, we found that performance and frequency were strongly correlated on Longhorn \((\rho = -0.93)\).
Fig. 2: Summary results for SGEMM on Longhorn cluster presented as box plots of (a) frequency, (b) kernel duration (performance), (c) power, and (d) temperature. The color indicates the cabinet of the GPU.

Fig. 3: Scatter plots for SGEMM on Longhorn showing that (a) performance and temperature have a weak positive correlation (Pearson correlation coefficient $\rho = 0.46$) and (b) power and performance have a weak inverse correlation ($\rho = -0.35$). The color indicates the cabinet of the GPU.

Fig. 4: Summary results for SGEMM on Summit, showing variation in (a) kernel duration (performance), (b) frequency, (c) power, and (d) temperature. We breakdown box plots by row because of the large scale of the cluster. GPUs, we further breakdown our measurements based on the particular row and column where a machine resides (using node location references derived from ORNL’s layout [67]).

Figure 4 shows aggregated kernel duration, frequency, power, and temperature box plots, grouped by rows. Similar to Longhorn, Summit has 8% performance variation across
Fig. 5: Scatter plots of Summit data showing (a) performance and frequency have strong negative correlation ($\rho = -0.99$) and (b) performance and power have almost no correlation ($\rho = -0.09$). The color indicates the row for the GPU.

all rows, with rows D and F having the most outliers (Figure 4a). Likewise, frequency variation (Figure 4b) is again near 100MHz across all rows, although rows D and F have some outliers below 1300MHz. Moreover, although all row’s IQRs (Figure 4c) range from 295-300W, a number of GPUs consume less than 290W, especially in rows A and H. However, unlike Longhorn, Summit’s temperature range is narrow: 40°F-62°F. This shows the benefits of water cooling [39] compared to Longhorn, although water cooling does not significantly impact frequency or performance variation.

Figure 5 shows Summit’s scatter plots. Similar to Takeaway 1, performance and frequency were directly correlated ($\rho = -0.99$) on Summit (Figure 5a). However, unlike Longhorn, Summit has a string of power outliers below 290W with 2510ms runtime (Figure 5b). This is unexpected, since GPUs with a wide power range typically vary in their performance.

To understand these outliers, we analyzed row H, column 36 because of its variance in power consumption. We found that 7 of its nodes exhibited power outliers, with power as low as 255W, while the remaining 9 nodes did not have any outliers. However, despite the power outliers, nodes 10 and 11 did not have any temperature outliers, unlike Longhorn. Hence, these two nodes show that although water cooling decreases temperature variation, it does not prevent nodes from having large performance and power variation. Moreover, the variance in power consumption seen in row H, column 36 highlights the difficulty in drawing conclusions from cluster-wide summaries and provides another example of how our study can be used to flag underperforming nodes early for system administrators to investigate further, similar to Longhorn.

Takeaway 2: Summit and Longhorn have similar performance and frequency variation trends, but Summit has severe power outliers that are concentrated in a few rows.

Takeaway 3: While Summit and Longhorn have the same GPU temperature setups, water cooling reduces temperature variation, but does not improve performance and power variation.

D. SGEMM on Corona LLNL

Since compute clusters use GPUs from multiple vendors, it is important that we also examine how AMD GPUs vary. Corona uses air-cooling for its AMD MI60 GPUs. We collected the same measurements as Longhorn and Summit. However, unlike Longhorn and Summit we were unable to obtain a system map and thus analyze our data by node. Nevertheless, to improve readability we group nodes in “cabinets” of 12 GPUs, like the similarly-sized Longhorn cluster.

Figure 6 presents aggregated box plots for our 4 metrics. Individual data points are distinguished by our node groupings. Overall, Corona exhibits 7% runtime variation (Figure 6b), similar to Longhorn (9%) and Summit (8%). Interestingly, frequency and performance are not as strongly correlated on Corona ($\rho = -0.76$ on Corona, versus Longhorn’s −0.93 and Summit’s −0.99), as Figure 6a shows much less variability than Longhorn and Summit (Takeaways 1-2). Moreover, the MI60s have coarser frequency levels than the NVIDIA V100s, suggesting that the GPU vendor’s DVFS schemes vary significantly. Finally, the power (2%, Figure 6c) and temperature (20%, Figure 6d) variability are similar to Longhorn. However, whereas Longhorn has more power outliers consuming as little as 250W, Corona only has one outlier node (c115) which consumes 165W. Surprisingly Corona’s nodes never reach the max power of 300W. Since the temperature are as high as 99°C (near the slowdown temperature), we believe the DVFS controller heavily throttles the frequency (most frequencies also do not reach the peak 1800MHz), thereby decreasing the power since $P_{\text{dynamic}} \propto frequency$.

To further analyze the outlier node, c115, Figure 7 correlates our metrics. Corona’s kernel runtime and power relationship (Figure 7b) is similar to Longhorn and matches expectations: slower GPUs tend to consume less power. However, Summit does not exhibit this trend, as similarly performing GPUs in Summit exhibit a wide power range. The performance-power correlation coefficients are −0.48 and −0.35 on the similarly-sized Corona and Longhorn clusters, respectively, but −0.09 on Summit. However, since the cluster sizes differ, we

2We provide only a brief summary of our results here due to space constraints. Detailed analysis and graphs for this row-column pair can be found in Appendix B of our technical report [49].
Fig. 6: Summary results for SGEMM on Corona presented as box plots of (a) frequency, (b) mean kernel duration (performance), (c) power, and (d) temperature. Node c115 (green) is the outlier.

Fig. 7: Scatter plots for SGEMM on Corona showing correlations between (a) kernel duration (performance) and temperature ($\rho = 0.20$) and (b) kernel duration and power ($\rho = -0.48$). Node c115 (green) is the outlier.

also compared Summit against a scaled normal distribution of Longhorn’s performance numbers to determine cluster size’s impact. The scaled normal performance distribution projects that the Longhorn data would have 9.4% variability on a Summit-sized cluster. Since our actual Summit measurements (Section IV-C) had 8% performance variability, this suggests the cluster size may impact the severity of variability. Also, while c115 runs very hot compared to most nodes (Figure 7a), there is not a clear relationship between kernel runtime and temperature ($\rho = 0.20$). Typically higher temperatures occur when GPUs run at higher frequencies and achieve better performance. However, there are several nodes that run as hot as c115 but which do not perform worse than the median GPU, which likely rules out that this outlier was due to inadequate air cooling. Moreover, all of these GPU’s temperatures are near the slowdown temperature. Additionally, Summit nodes that perform similarly also exhibit a wide temperature range. For example, Corona GPUs performing around 2210ms have a 25°C temperature range. Thus, c115 appears to be an underperforming GPU and another candidate for further inspection and potential replacement by the system administrators.

Lastly, a single GPU exhibits little performance variation across independent application runs. Figure 8 shows the normalized per-GPU performance variance for SGEMM on various clusters. The median variance values are 0.44%, 0.12% and 6.06% on Longhorn, Summit, and Corona, respectively. Thus, our results are repeatable and any transient effects cause only minor variations. Although there are a few outliers as high as 12% (Longhorn node c002-010), we found that these do not correspond to the worst performing GPUs. Instead, they fall between the median and $Q_3$, which suggests that ill-performing GPUs are consistently ill-performing.

**Takeaway 4:** Corona’s AMD GPUs exhibit similar behavior to the like-sized Longhorn cluster with NVIDIA GPUs. However, the performance-power relationship in the larger Summit differs from Corona and Longhorn, suggesting that cluster size affects variability degree.

E. SGEMM on Vortex and Frontera

Sandia National Labs’ (SNL) Vortex cluster [46] is similar in scale to Longhorn and Corona, but is water-cooled (Table I). On the other hand, as specified in Table I, TACC’s Frontera [45] is a liquid submerged system with mineral oil-cooled Quadro RTX GPUs [45]. Due to space constraints, we only present the conclusions we drew from Vortex and Frontera.
We observed a 9% performance variation when running SGEMM on Vortex, similar to Longhorn (9%) and Summit (8%). Vortex’s GPUs exhibited a narrower temperature variation, reinforcing **Takeaway 3** – water cooling reduces temperature variation but does not diminish performance or power variability. Interestingly, on Frontera we observed relatively lesser runtime variation (5%) but Frontera also had some significant outliers. Frontera’s temperature and power findings relative to Longhorn and Vortex suggest that mineral-oil cooling lies between air and water-cooling in how effectively it reduces thermal variation.

### V. VARIATION ACROSS APPLICATIONS

We next study different applications to determine whether variability is application-specific or not.

#### A. ResNet-50 on TACC Longhorn

Given their widespread use on GPU-rich clusters, it is important to understand performance variability for ML applications. Thus, we studied the ResNet-50 CNN for the most intensive training phase. We chose the 50-layer version because it is a stable, commonly used benchmark in the HPC community [68], [69]. In general, ResNet-50 is computationally intensive, especially for its kernels that perform convolution. However, ResNet-50 also performs other, more memory intensive operations, which reduce its overall computational intensity compared to SGEMM. This is confirmed by examining FU utilization: averaging across kernels, ResNet-50’s was 5.4 (on a 1–10 scale) while SGEMM’s FU utilization was 10. This also aligns with observations from prior work on classifying GPU applications [70]. Our training set uses 1.2 million images from ImageNet with batch size 64. We define one training run as 500 iterations. Note that we did not complete training on our entire training set; 500 training iterations was sufficiently long to collect profiling data while training was stable. Finally, since ResNet-50 is commonly trained using multiple GPUs, we trained across four GPUs on one node and trained 3–4 times per node. Like SGEMM, before collecting data we perform one warm-up run.

Unlike SGEMM, ResNet-50 has ≈85 unique kernels and over 1.3 million kernels per run. Since 75% of these kernels run for less than 2ms, accurately measuring them is challenging (Section III). Thus, averaging kernel duration would not fairly characterize its overall performance. Hence, for ResNet-50 we use iteration duration instead of kernel duration as our performance metric; iteration duration is also more informative for HPC users training ML models. All other metrics are the same as SGEMM. However, we ignore the initialization kernels (e.g., NCCL) to avoid one-time startup costs.

Figure 9 presents ResNet-50’s aggregated box plots on Longhorn for our 4 metrics. Unlike our SGEMM experiments on Longhorn (Section IV-B), ResNet-50 has little frequency variation and most nodes run at the max 1530MHz. However, ResNet-50 has 22% performance variation (Figure 9b), our largest observed variation. The temperature variation (Figure 9d) is similar to SGEMM’s: a 30°C range. Surprisingly, unlike SGEMM there is significant power variation (104%,
Figure 9c). We believe this reflects ResNet-50’s more varied behavior across its different kernel types – although its convolution kernels are similar to SGEMM, ResNet-50’s other, less computationally intense kernels have less onerous power demands. Consequently, the DVFS algorithm does not need to reduce voltage or frequency to remain beneath the TDP.

Figure 10 presents scatter plots of our distinct metrics. Figure 10a shows that most iterations complete within 100-150ms and GPUs have frequencies around 1530MHz. However, surprisingly, there are tails on both axes. In particular, several c008 runs have much lower frequency yet perform similarly to those running at the max 1530MHz. Meanwhile, several GPUs in c002 run at 1530MHz (Figure 10a), but perform poorly and consume much less power (Figure 10b). We expect c002’s stragglers to consume more power than the tail of c008’s runs, since \(P_{\text{dynamic}} \propto \text{frequency}\). However, we observe the opposite: c002’s worst performing runs consume much less power (as low as 76W) than c008 and other cabinets. This differs from Longhorn’s SGEMM runs. Thus, ResNet-50 and SGEMM’s variability and correlation differences on Longhorn suggest that variability is application-specific. Moreover, 8 of the 10 worst-performing GPUs for SGEMM were also outliers for ResNet, highlighting that variability is not transient and some GPUs consistently do not perform well.

We also analyzed ResNet-50 when running as a single GPU application. However, due to space constraints we include its analysis and results in our extended technical report [49].

**Takeaway 5:** ResNet-50 exhibits the highest performance variability (22%) across all our benchmarks. Moreover, the difference in ResNet-50’s and SGEMM’s compute intensities, and the significant difference in performance-frequency correlations, suggests that variability is application-specific.

### B. BERT on TACC Longhorn

Given the increasing adoption of Transformer-based models [7], [19], [71]–[78], we also study another multi-GPU ML workload: pre-training of Bidirectional Encoder Representations from Transformers (BERT) [58]. BERT is widely used in Natural Language Processing (NLP) and is in the popular MLPerf suite [68], [69]. We used BERT\_LARGE, which has 24 encoders with 16 bidirectional self-attention heads. Our training set was 30522 words and uses a batch size of 64. Like ResNet, we limit each training run to 250 iterations and run across all 4 GPUs in a node. We performed one warm up run and then performed measurements 5 times on each node. Overall we observed 53 nodes in Longhorn for BERT and collect the same metrics as ResNet. Similar to ResNet, we omit the initialization kernels to avoid one-time startup costs.

Figure 11 shows aggregated box plots on Longhorn for our 4 metrics. Compared to ResNet, BERT has less overall power consumption (Figure 11c): BERT’s median power consumption is around 40W lower. This is expected because ResNet uses more compute intensive GEMMs, which increase its power consumption. In comparison, BERT’s GEMMs are much less computationally intensive: although GEMMs make up 30-65% of its total runtime, they only utilize 40-50% of the GPU [77], [79]–[81] – which decreases BERT’s overall power consumption. Profiling FU utilization confirms this: BERT’s FU utilization was 1.2 on average, much less than ResNet’s 5.4 and SGEMM’s 10. Nonetheless, like ResNet there is large power variability (~87%). However, because BERT is less computationally intensive than ResNet, it has less performance variability (~8%) and fewer frequency outliers (Figure 11b and Figure 11a, respectively). Perhaps unsurprisingly, the performance outliers in Figure 11 in cabinet c002 on Longhorn are also outlier nodes for ResNet in Figure 9b. This suggests that the same GPUs perform poorly for both ML applications.

**Takeaway 6:** Like ResNet-50, BERT exhibits large power variability. However, BERT’s less computationally intensive GEMMs reduce performance variability (8%). Moreover, BERT’s and ResNet-50’s outlier nodes are the same.

### C. LAMMPS on TACC Longhorn

LAMMPS is a popular molecular dynamics (MD) application that simulates a variety of atomic and molecular systems [59]. Whether LAMMPS is compute or memory-bound depends on the selected settings and hardware [53], [82], [83]. We chose to use the REAXC setting [84] to simulate a chemical reaction. When REAXC is run in a distributed multi-GPU setup with large simulation sizes, LAMMPS is compute-intensive [53]. However, to compare LAMMPS variability with SGEMM, we ran it as a single-GPU experiment, with input configuration parameters \((x, y, z)\) that determine GPU occupancy and are limited by the 16GB device memory on NVIDIA V100s. After careful tuning, we selected a size of \((8,16,16)\) which lead to high GPU utilization, while still remaining within device memory limits. Profiling shows that LAMMPS has 42\(\times\) higher DRAM utilization than ResNet, while ResNet kernels utilized FUs 4.3\(\times\) more than LAMMPS. Thus, LAMMPS is memory-bound in our experiments.

Each LAMMPS run is composed broadly of 2 types of kernels, short-running (\(\leq 60\mu s\) long) and long-running kernels (4 unique kernels, 20-200ms long). Long-running kernels make up 98% of the total runtime of a LAMMPS job but there are 4 unique long kernels interspersed with short ones. Thus, similar to ResNet-50, median kernel duration is not a good performance measure. We therefore use the sum of all large kernel durations as our performance metric. All other metrics are measured as specified in Section III.

Figure 12 presents LAMMPS’ aggregated box plots on Longhorn for our 4 metrics. Interestingly, median power for all LAMMPS jobs was \(\leq 180\) W (Figure 12c), in contrast to more compute-intensive applications such as SGEMM, which often touched V100 TDP. Moreover, similar to BERT and ResNet-50, frequency quickly saturates to the maximum value of 1530MHz (Figure 12a) and does not change throughout the course of the job. Additionally, performance varies by less than 1% in Figure 12b, which completely differs from BERT (8%), SGEMM (9%), and ResNet-50 (22%). However, we still observe power variability of 20% and temperature variability of 8°C between Q1 and Q3 (Figure 12d). These observations are in line with prior work [82] and emphasize that high energy
consumption is undesirable in a memory-bound application because it is not accompanied by any significant increase in performance. In part, this happens because the GPU’s memory frequency is lower than the compute frequency, and even if the application is memory-bound, this does not stress the TDP as much as the compute-heavy applications. Overall, this suggests that (i) SM frequency gets pinned in applications that are less compute-intensive (ii) performance of such applications is relatively more predictable with very low variability in application runtime and (iii) significant temperature and power variability are still observed across GPUs.

**Takeaway 7:** Similar to **Takeaway 5**, LAMMPS’ results show that variability is application-specific with memory-intensive applications seeing lower performance variance, but still having significant power and temperature variability.

### D. PageRank on TACC Longhorn

PageRank is a popular graph analytics algorithm that is used in recommender systems, search engines, social network analysis, and bibliometrics [85]. Although neither push- nor pull-based graph analytics algorithms always provide the best performance on GPUs [86]–[88], we focus on pull-based algorithms because they are more widely used. The most common PageRank algorithm utilizes Sparse matrix-Vector (SPMV) computation [60], [85]. Since PageRank’s sparsity is input graph dependent, its memory access pattern can be highly irregular and often dwarfs the amount of compute, making it both memory bandwidth-bound and highly irregular. We chose an input graph that fully utilizes the SMs of a V100 GPU and provides sufficiently long kernels runtimes (Section III). We ran PageRank with rajat30, an undirected graph for circuit simulation [89]. The other configuration parameters are the same as SGEMM on Longhorn. Like LAMMPS, SPMV computations are also memory bound, but irregular. Consequently, they do not stress memory as much: LAMMPS has 4.24× higher DRAM utilization than PageRank. However, PageRank kernels had 61% memory dependency stalls, versus 7% for LAMMPS and 3% for SGEMM. PageRank also had negligible FU execution dependency stalls (12× less than SGEMM), showing that PageRank is not compute-bound.

Figure 13 summarizes Longhorn variability for PageRank. Similar to LAMMPS (Section V-C), PageRank has little fre-
Fig. 14: Day of the week summary results for Summit. We see more power outliers on Mondays, Wednesdays, and Fridays.

VI. VARIATION ACROSS TIME, POWER BUDGET

Next, we analyze how variability changes across days of the week and how the GPU power limit affects variability.

A. Variation Across Days of the Week

To determine if the variations we observed on these clusters hold over time, we ran SGEMM on Summit on each day of the week across a period of eight weeks. Figure 14 shows a consistent trend across each day of the week: around 8% variation. On Mondays, Wednesdays, and Fridays there is a higher concentration of GPUs which consume below 290W. However, even with more power outliers, performance on these days is very similar to the rest of the week and Takeaways 1-3, as is also shown in Figure 5. Longhorn also has similar results, but we omit detailed analysis due to space constraints.

Takeaway 9: The variability we observe is consistent throughout the week, suggesting that regardless of when the experiments are run, our observations hold.

B. Varying Power Limit

We also studied how varying a GPU’s power limit affects variability. Varying the power limit (e.g., using nvidia-smi) requires administrative access. Since we did not have administrator (root) access on the large compute clusters, we used the smaller CloudLab [90], which has 12 NVIDIA V100 GPUs. Figure 15 shows the performance variation as the power limit varies from 100-300W while running SGEMM.

Fig. 15: Performance variation of SGEMM on NSF CloudLab GPUs when varying the power limit.

Interestingly, unlike prior work [30] performance variation was similar when pinning and not pinning (Section IV-B). Rather, our variability results for a 300W limit are in line with those from the large clusters: 9% variation. Moreover, as expected, the kernel durations increase with lower power limits. However, variability and the number of outliers also increase with lower power limits. For example, at a 150W limit there is 18% variability versus 9% at a 300W limit. Our conversations with GPU manufacturers indicate that this is potentially occurring because GPU DVFS algorithms are less optimized for extremely low power budgets, but variability under power limits would become important if future exascale machines are operating under a varying power budget [91].

VII. CONCLUSIONS AND TAKEAWAYS

Modern compute clusters are embracing accelerator-rich systems, especially GPUs. Although prior work has identified how variability in CPUs affects these systems, it is unclear how much GPU variability affects these systems. Thus, we conducted a detailed study and analysis of how GPU variability affects a wide range of modern HPC and scientific workloads across five computing centers of varying scales, cooling methods, and GPU vendors. Our results show that there is significant variability in these systems: up to 22% for the applications, with outliers up to $1.5 \times$ slower than...
the median GPU. This demonstrates the need to embrace variability in future accelerator-based systems. We conclude by highlighting several implications and mitigation strategies that practitioners and researchers can pursue in the future:

**Impact on Users:** In addition to the performance variation, inter-experiment variability is also important. For example, when running SGEMM on Longhorn, 18% of the GPUs are 6-7% (or about 150ms) slower than the fastest GPUs. Thus, users running SGEMM-like single GPU experiments would have an 18% chance of being assigned a slower GPU. In Summit, again with SGEMM, 9% of the GPUs are 6-7% (or about 160ms) slower than the fastest GPUs. Thus, users running a single GPU experiment on Summit have a 9% chance of being assigned a slower GPU. Like Longhorn, Summit also has variance within specific GPUs. Users running multi-GPU experiments are even more likely to be assigned a slower GPU: if a user requests 4 GPUs on Longhorn, 40%-50% of the time they will be assigned a slower GPU. We also observed this in our multi-GPU BERT and ResNet-50 experiments.

**Spatial Effects:** For all of our experiments, we obtained exclusive access to a machine and all its GPUs. This eliminated any spatial effects from jobs running on neighboring GPUs. While this is the typical allocation mode in modern supercomputing clusters, spatial effects would be relevant for other scenarios like cloud computing or enterprise clusters [92] where GPUs are allocated individually. We plan to study both spatial and temporal (i.e., variability due to a preceding job run on the same GPU) effects in the future.

**Blacklisting, Maintenance:** Cluster operators can use our study to improve the cluster’s operation and help develop strategies for better maintenance. For example, our study has already helped TACC’s operators identify and perform targeted maintenance on problematic nodes with clearly underperforming GPUs in Frontera and Longhorn. Performing periodic variability benchmarking can help automate this.

**Application-aware Frameworks:** Since performance variation is application dependent, the next generation of HPC allocation frameworks should take application properties into account to mitigate variance. Similar to prior work [70], our profiling indicates that metrics like FU utilization, DRAM utilization, and memory stalls can be used by operators to classify applications and modify schedulers to assign medium- and high-compute-intensive workloads on nodes with less variation. Similarly, memory-bound applications can be run on higher-variability nodes without incurring significant performance loss. However, this may change in future as thermal performance degrades below 14nm [93].

**New Hardware and System Design:** A major limiter to further improving efficiency is the lack of standards for exposing power information in modern accelerators. Thus, for future systems, designers can build on the insights generated by our benchmarks for current systems and apply co-design that makes the hardware, software, and runtime layers aware of the variance in the systems. To do this, we will need to design a standard for accelerators to expose PM information from the hardware to the software and runtime. Using this information, we can develop techniques for global power management that can enable optimal PM decisions across accelerators and further reduce performance variability.

**ACKNOWLEDGMENTS**

We thank the anonymous shepherd and the SC reviewers for their constructive comments and suggestions that improved this work. We also thank Zhao Zhang (TACC), Clayton Hughes (SNL), Oscar Hernandez and David Bernholdt (ORNL), Ian Karlin and Scott Futral (LLNL) for providing us machine access on the clusters we used and helping us run some of the experiments. Additional thanks Ian and Scott for providing information about slowdown temperatures for the Corona GPUs. We extend our gratitude to Stan Moore and Christian Trott (SNL) for their help with using Kokkos library for LAMMPS. We also acknowledge Scott Groel at Cloudlab Clemson for clarification on V100 fans. Finally, we thank Joseph Greathouse (AMD) and Mike Sabotta (NVIDIA) for helping us understand GPU power management. This work is supported in part by a University of Wisconsin Fall Research Competition grant, a LLNL allocation, ORNL grant GEN010wisc, and a TACC allocation. Sinclair has an affiliate appointment with AMD Research.

**REFERENCES**

[1] J. Deng, W. Dong et al., “Imagenet: A large-scale hierarchical image database,” in IEEE Conference on Computer Vision and Pattern Recognition, ser. CVPR. IEEE, 2009, pp. 248–255.
[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in neural information processing systems, 2012, pp. 1097–1105.
[3] D. Amodei, R. Anubhai et al., “Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin,” in Proceedings of the 33nd International Conference on Machine Learning. ser. ICML, 2016, pp. 173–182.
[4] A. Hannun, C. Case et al., “Deep speech: Scaling up end-to-end speech recognition,” arXiv preprint arXiv:1412.5567, 2014.
[5] Y. He, T. N. Sainath et al., “Streaming end-to-end speech recognition for mobile devices,” 2018.
[6] M. Ott, S. Edunov et al., “Scaling Neural Machine Translation,” 2018.
[7] Y. Wu, M. Schuster et al., “Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation,” CoRR, vol. abs/1609.08144, 2016. [Online]. Available: http://arxiv.org/abs/1609.08144
[8] J. Junpner, R. Evans et al., “Highly accurate protein structure prediction with alphafold,” Nature, vol. 596, no. 7873, pp. 583–589, 2021.
[9] J. Jones, A. Svyatkovskiy, and W. Tang, “Predicting disruptive instabilities in controlled fusion plasmas through deep learning,” Nature, vol. 568, no. 7753, pp. 526–531, 2019.
[10] Z. Fan and E. Ma, “Predicting orientation-dependent plastic susceptibility from static structure in amorphous solids via deep learning,” Nature Communications, vol. 12, no. 1, pp. 1–13, 2021.
[11] L. Casalino, A. C. Dommer et al., “AI-driven multiscale simulations illuminate mechanisms of SARS-CoV-2 spike dynamics,” The International Journal of High Performance Computing Applications, p. 10943420211006452, 2021.
[12] top500.org, “TOP500 List,” 2021, https://www.top500.org/lists/top500/2021/11/.
[13] J. Fowers, K. Ovtcharov et al., “A Configurable Cloud-scale DNN Processor for Real-time AI,” in Proceedings of the 45th Annual International Symposium on Computer Architecture, ser. ISCA. Piscataway, NJ, USA: IEEE Press, 2018, pp. 1–14, [Online]. Available: https://doi.org/10.1109/ISCA.2018.00012
[14] Tesla, “Ahead of ‘Dojo,’ Tesla Reveals Its Massive Precursor Supercomputer,” 2021. [Online]: https://www.hpcwire.com/2021/06/22/ahead-of-dojo-tesla-reveals-its-massive-precursor-supercomputer/.
SUMMARY OF THE EXPERIMENTS REPORTED

We ran different benchmarks across machine learning, molecular dynamics and graph analytics. Using vendor-specific profiling tools (specified below), we measured power (W), temperature (degree C), frequency (MHz) and performance (kernel runtime/iteration duration/total runtime depending on application). A summary of our experiments:

- **SGEMM on NVIDIA V100s**: an application we wrote that utilizes NVIDIA’s cuBLAS library to perform matrix multiplication on two matrices containing single-precision floats (cuda 10.1, nvcc 10.1, nvprof 10.1, gcc 4.8.5)
- **SGEMM on AMD MI60s**: we also wrote sgemm with AMD’s rocBLAS library (rocm 4.0.1)
- **ResNet-50 on NVIDIA V100s**: a stable CNN application used widely in the HPC community as a benchmark (cuda 10.1, nvprof 10.1)
- **BERT on NVIDIA V100s**: a transformer model that is widely used for Natural Language Processing (NLP), part of the MLPerf suite.
- **LAMMPS on NVIDIA V100s**: a molecular dynamics application widely used for scalable science experiments (cuda 10.1, nvcc 10.1, nvprof 10.1)
- **PageRank on NVIDIA V100s**: a graph analytics benchmark that uses sparse matrix vector multiplication (cuda 10.1, nvcc 10.1, nvprof 10.1)

AUTHOR-CREATED OR MODIFIED ARTIFACTS:

**Artifact 1**
Persistent ID: [https://doi.org/10.5281/zenodo.7010207, https://github.com/hal-uw/gpu_variability_sc22_artifact.git]

Artifact name: Experiments Ran in ‘Characterizing Variability in Large-Scale, Accelerator-Rich Systems’ Paper

*Reproduction of the artifact with container:* In our GitHub repository, there are top-level shell scripts for each application used in our paper. These scripts pull container images from the Docker registry using Singularity, and run the application within the container. We provide detailed instructions for how to use these scripts for easy reproducibility in the README files for each application. All experiments assume access to a compute node with a GPU. Prerequisites for all experiments are specified in the respective README files.