Understanding GPU Memory Corruption at Extreme Scale: The Summit Case Study

Vladyslav Oles
Oak Ridge National Laboratory
Oak Ridge, TN, USA
olesv@ornl.gov

Anna Schmedding
William & Mary
Williamsburg, VA, USA
akschmedding@wm.edu

George Ostrouchov
Oak Ridge National Laboratory
Oak Ridge, TN, USA
ostrouchovg@ornl.gov

Woong Shin
Oak Ridge National Laboratory
Oak Ridge, TN, USA
shinw@ornl.gov

Evgenia Smirni
William & Mary
Williamsburg, VA, USA
esmirni@cs.wm.edu

Christian Engelmann
Oak Ridge National Laboratory
Oak Ridge, TN, USA
engelmannc@ornl.gov

ABSTRACT

GPU memory corruption and in particular double-bit errors (DBEs) remain one of the least understood aspects of HPC system reliability. Albeit rare, their occurrences always lead to job termination and can potentially cost thousands of node-hours, either from wasted computations or as the overhead from regular checkpointing needed to minimize the losses. As supercomputers and their components simultaneously grow in scale, density, failure rates, and environmental footprint, the efficiency of HPC operations becomes both an imperative and a challenge.

We examine DBEs using system telemetry data and logs collected from the Summit supercomputer, equipped with 27,648 Tesla V100 GPUs with 2nd-generation high-bandwidth memory (HBM2). Using exploratory data analysis and statistical learning, we extract several insights about memory reliability in such GPUs. We find that GPUs with prior DBE occurrences are prone to experience them again due to otherwise harmless factors, correlate this phenomenon with GPU placement, and suggest manufacturing variability as a factor. On the general population of GPUs, we link DBEs to short- and long-term high power consumption modes while finding no significant correlation with higher temperatures. We also show that the workload type can be a factor in memory's propensity to corruption.

CCS CONCEPTS

• **Hardware** → **Transient errors and upsets**; • **Computer systems organization** → *Single instruction, multiple data*; *Reliability*.

KEYWORDS

HPC, GPU memory failures, data analysis

ACM Reference Format:
Vladyslav Oles, Anna Schmedding, George Ostrouchov, Woong Shin, Evgenia Smirni, and Christian Engelmann. 2024. Understanding GPU Memory Corruption at Extreme Scale: The Summit Case Study . In *Proceedings of the 38th ACM International Conference on Supercomputing (ICS '24), June 04–07, 2024, Kyoto, Japan.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3650200.3656615

ICS '24, June 04–07, 2024, Kyoto, Japan

ACM ISBN 979-8-4007-0610-3/24/06
https://doi.org/10.1145/3650200.3656615

1 INTRODUCTION

At the verge of the exascale era, we are facing an unprecedented complexity of modern HPC systems, operating which in a reliable manner is a non-trivial task. In particular, computational clusters have fully embraced GPUs as a key component, characterized by higher parallel performance and an increased failure rate as compared to CPUs. Random bit flips, where the memory state unintentionally changes from 0 to 1 or vice versa, is one of the most pervasive and disrupting GPU errors. While GPUs are typically equipped with error-correcting code scheme that is able to detect and correct single bit flips, an occurrence of two bit flips in the same memory word (double-bit error, DBE) is a critical event that can potentially cost thousands of wasted node-hours.

Reliability of GPUs and in particular their memory has been the subject of multiple studies [23, 24, 28, 44, 45]. GPU memory bit flips have been successfully correlated with workload and thermal patterns on some systems [23, 24] but full understanding of the reasons behind this phenomenon is lacking. Except for the rare cause of cosmic radiation [41, 43], a more likely explanation is charge leakage to a neighboring memory cell, especially under extreme temperatures [17, 29] or intensive memory access [16]. Moreover, because the memory chip density is known to affect the incidence of memory errors [19], the potential impact of the 3D-stacking architecture used in high-bandwidth memory reliability is yet to be studied.

This paper aims to understand the relationship between GPU activity and bit flips in GPU memory by studying the Summit supercomputer, a pre-exascale system at the Oak Ridge Leadership Computing Facility. Commissioned in 2019 and still operational, Summit is equipped with 27,756 GPUs featuring HBM2 memory which provides an opportunity to study bit flips in GPU memory at scale. Our analysis of Summit operational data spanning over 2.5 years identifies strong relationships between certain operation patterns and DBEs. To the best of our knowledge, this is the first study addressing DBEs in HBM2 units at scale.

Comprehensive reliability study on large-scale data from a production HPC system. Our study is conducted on a multi-modal dataset featuring high-resolution (1Hz) power and thermal telemetry from all 27,756 GPUs in the system, job scheduler records,

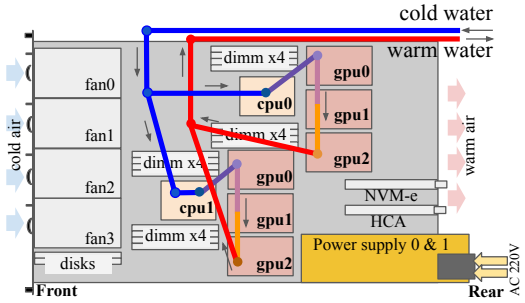


Figure 1: Physical layout of a Summit node [36]

and GPU memory error logs from a 2.5-year interval. By correlating memory errors with other operational data streams, we gain better understanding of factors associated with the errors and identify those factors for which no such association can be established.

Systematic identification of variables relevant to DBE occurrence. Given the scarcity of DBE events and the vast number of potential factors behind their occurrence, it is critical to narrow down the range of relevant factors. We apply various methods such as t-tests, survivability analysis, and interpretable machine learning models to systematically identify features that have high impact on DBEs. We identified variation in short-term power intake, application behavior, lifetime GPU utilization, and activity levels of a GPU in the idle state as some of the key variables with a strong relationship towards memory corruption.

Reliability characteristics of HBM2 units at scale. Our study reveals new insights about DBEs in HBM2 GPUs, further enriching knowledge gained by prior GPU reliability studies in HPC settings [23, 24, 28, 44, 45]. Among many findings, we report that

- geometrically central GPU placements in the node layout exhibit higher resilience to memory corruption;
- DBEs are strongly correlated with power consumption dynamics, and particularly the high power fluctuations that reach towards the thermal design power limits;
- the majority of identified DBE-prone HPC applications have mixed-precision capacity;
- higher intensity of the long-term GPU utilization increases its susceptibility to future memory corruption;
- the thermal state seen in production settings has minimal impact despite occasional high-temperature values near or beyond safe limits (80°C – 90°C).

2 SYSTEM AND DATA

2.1 Summit architecture

Summit, which entered No. 1 on the Jun. 2018 edition of the Top500 list [12] of supercomputers, is a 122.3 petaflops pre-exascale system located at the Oak Ridge Leadership Computing Facility (OLCF). It has a total of 4,626 IBM AC922 nodes organized into 257 cabinets (vertical stacks of 18 nodes), which are arranged on the floor as an incomplete 8 x 37 grid. Each Summit node is powered by two Power9 CPUs and six Nvidia Tesla V100 GPUs, whose physical layout is shown on Figure 1. The total of 27,756 V100 units are equipped with 16GB modules of stacked HBM2 memory.

Errors in memory can be caused by natural radiation and thermal neutrons (bit flips), by defects (stuck bits) and faulty memory logic [27, 39, 42]. The Nvidia Tesla V100 architecture offers several GPU memory error handling methods. It supports single error correction double error detection (SECDED) error correcting code (ECC). A single-bit error (SBE) is corrected upon reading from corrupted memory, while a double-bit error (DBE) can be only detected. The Nvidia driver additionally supports the retirement of memory pages that contain bad memory cells [26]. This dynamic page retirement excludes a memory page from subsequent allocations, which is referred to as offlining, once a page experienced a DBE or two SBEs on the same address. A memory page is then mapped out of usage by the driver upon next reattachment of the GPU. Offlining memory pages requires stopping all GPU clients and reinitializing the GPU or rebooting the system. Up to 64 memory pages can be retired, at which point the page retirement table is full. The number of DBE and SBE locations that can be temporarily stored for offlining upon the next reinitialization or reboot is at least 192 and can be up to 600 depending on the GPU model.

The following errors are logged to the system error log [1]:

- XID 48, a DBE detected,
- XID 63, a page retirement event (PRE) occurred, and
- XID 64, a page retirement failure (PRF) occurred.

In general, a logged PRE without a related preceding logged DBE points to two SBEs on the same address as being the reason for the page retirement. Otherwise, the reason is a DBE. A PRE followed by a related PRF means that the page has been listed for retirement but has not been retired yet, perhaps due to the absence of reboot. A PRF without a related preceding PRE means that the 64-page retirement limit is reached.

2.2 Data preparation

Our study is based on several source datasets covering different aspects of Summit operations in the period between 1 Jan 2020 and 17 May 2022, which is an extension of the datasets in [36]. Table 1 describes the contents of the datasets. Using these source datasets, we prepared derived datasets (see Table 2), described below.

Nvidia error records store times and locations of DBEs and other GPU hardware and software errors (Table 2-(a)). Error locations, given as a combination of the Summit node and its PCI Express bus hosting the offending GPU, do not uniquely identify a physical unit as GPUs are occasionally moved between the Summit nodes for reliability purposes. To link GPU errors to physical units, we rely on boot logs of individual Summit nodes that record a mapping between the 6 PCI Express buses and serial numbers of the corresponding GPUs every time a node is restarted.

To better understand the circumstances associated with the GPU error events, we use their job allocation-contextual information to create GPU snapshots (Table 2-(b)), a concept similar to the black box flight recorder, that represent 1Hz time-series telemetry before and after the GPU error events. Further, to establish a baseline for comparative analysis, we create analogous GPU snapshots derived from non-error periods of the GPU lifetime (Table 2-(c)). A detailed description of snapshot construction is given in Section 4.

Table 1: Source Datasets (1 Jan 2020 to 17 May 2022)

id	Source	Sample Interval	Rows	Footprint	Description
(a)	NVidia GPU XID error log	At occurrence	3M	600MB	GPU error hardware and software errors
(b)	Summit node reboot log	At reboots	60K	7MB	Reboot time scan of GPU PCI Express bus and serial numbers
(c)	Job scheduler allocation history	End of every job	938K	285MB	Project, user, node count, allocation param., submit, start & end time
(d)	Per-node job scheduler allocation history	End of every job	88M	14GB	Per-node job allocation history, end of job statistics
(e)	Per-GPU power and thermal telemetry	1 sec	268B	16TB (compressed)	Per-node, per-component power and temperature

Table 2: Data Preparation Overview

id	Name	Source	Footprint	Description
(a)	Location augmented GPU error logs	Table 1-(a,b)	8MB	GPU error hardware and software errors augmented with their physical locations
(b)	DBE GPU Snapshots	Table 1-(a,c,d,e)	330KB	Per GPU snapshots of power & thermal time series in proximity with the DBEs
(c)	Sampled normal GPU Snapshots	Table 1-(a,c,d,e)	56MB	Per GPU snapshots of sampled normal snapshots of power & thermal time series

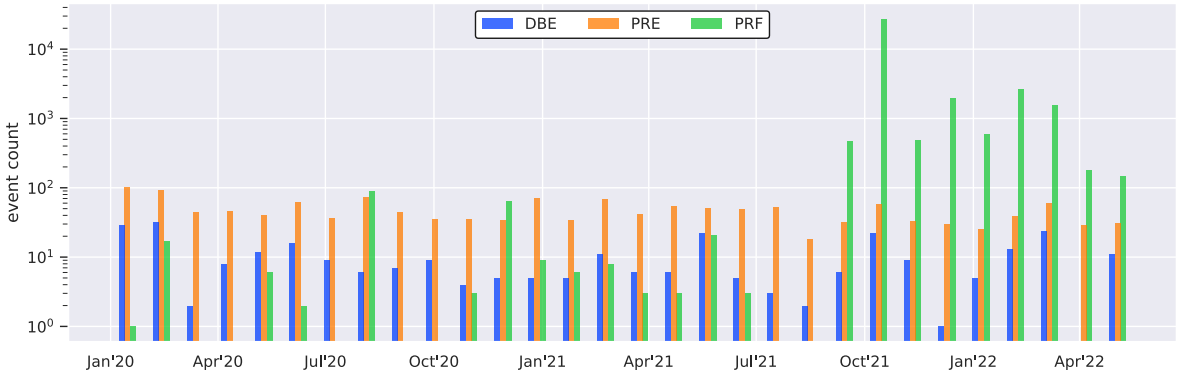


Figure 2: Temporal trends of memory corruption events on Summit.

2.3 Dataset availability

GPU error logs, node boot logs, and per-node job scheduler history (Table 1-(a,b,d)), as well as the snapshot dataset (Table 2-(b,c)), are publicly available at https://doi.org/10.13139/OLCF/1970187 [37].

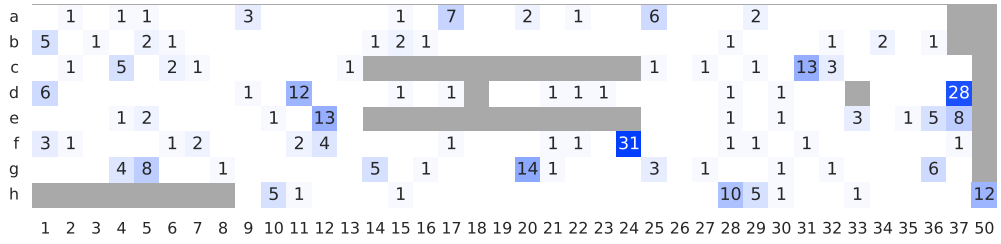
3 EXPLORATORY ANALYSIS

To understand general trends and patterns of GPU memory errors on Summit, we analyze Nvidia error logs to obtain summary statistics of DBE, PRE, and PRF occurrences. In the period of January 2020 – May 2022 Summit has seen the total of 295 DBEs, 1,430 PREs, and 35,791 PRFs, which translates to a system-wide mean time between events of 70.7 hours, 14.6 hours, and 0.6 hours, respectively. Surprisingly, the number of PRFs has a drastic 170-fold increase from the analogous count for the period of 2020 alone [36]. At the same time, only a fraction of nearly 28K Summit GPUs are affected by memory errors — DBEs, PREs and PRFs are encountered by 112, 1,011, and 138 GPUs, respectively. Furthermore, the top-1 GPUs by error count in each category have accounted accordingly for 10.5% of DBEs, 2.6% of PREs, and 73.3% of PRFs. Interestingly, the latter 73.3% have all occurred within the same HPC job, implying that PRFs can be induced by a user application, despite being considered hardware-caused in the NVidia list of GPU errors [1]. Because these 26,223 PRFs are preceded by a PRE at the very beginning of the job, we conclude that they were caused by the application repeatedly accessing the memory page whose retirement has been attempted and failed.

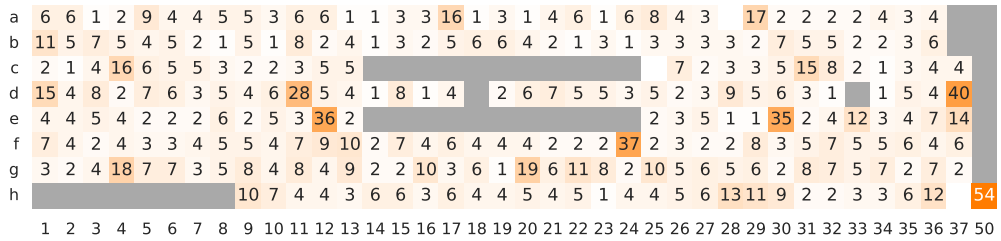
Finding 1. Large numbers of PRFs can be traced to application behavior.

To see if event occurrences have changed over time (e.g. due to GPU memory degradation), we plot the monthly error and failure event counts of DBEs, PREs, and PRFs (Figure 2). While the DBE and PRE counts seem consistent over the entire observation period, the PRFs show a significant uptick starting from September of 2021, shortly after the operating system of Summit was updated to Red Hat Enterprise Linux 8 on Aug 18. The 35,554 PRFs from 1 Sep 2021 – 17 May 2022 period took place within only 113 HPC jobs, out of the 15M jobs performed on Summit in this period. 97% of these PRFs occurred on GPUs with no DBEs in the observed 2.5-year interval, supporting the idea of application behavior, rather than hardware faults, being the cause.

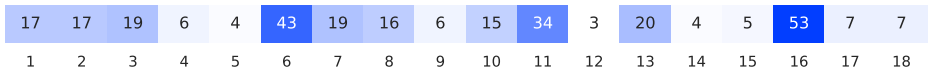
In the absence of reliable data about the HPC application run within a job, we use the project that a job belongs to and the submitting user as a proxy for identifying an application. We found that 97.2% of the 35,554 PRFs are only 10 project-user combinations spanning 27 jobs, which indicates that workload patterns of the associated applications are responsible for the anomalous PRF frequency. Prior to 1 Sep 2021, jobs corresponding to these project-user combinations encounter 46 PREs but no PRFs. It is possible that these applications never reached its page retirement limit until then. At the same time, the contiguity of the elevated



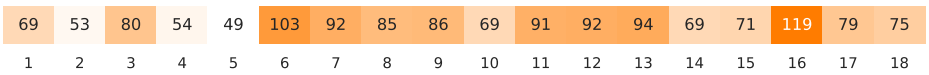
(a) DBE frequency per Summit cabinet in top-down view



(b) PRE frequency per Summit cabinet in top-down view.



(c) DBE frequency per node location (height) in the cabinet.



(d) PRE frequency per node location (height) in the cabinet.

Figure 3: Spatial distribution of DBEs and PREs on Summit.

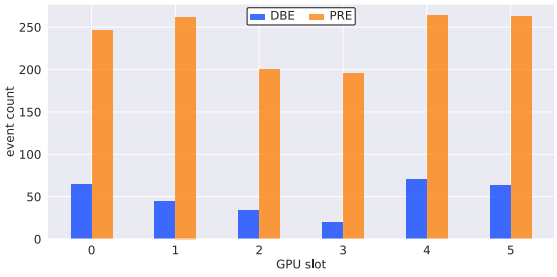


Figure 4: DBE and PRE frequency per GPU placement within Summit node.

PRF counts starting from this period makes it likely that the update of the Summit operating system enabled this.

To see if memory errors exhibit any spatial trends based on the location of a Summit node, we plot DBE and PRE counts per Summit cabinet and, separately, per node height in a cabinet (Figure 3). The correlation between locations of high DBE and PRE counts is likely a result of the similarity between causes of the two error types. With PREs being triggered both by DBEs and multiple SBEs on the same memory page, they are naturally more frequent than DBEs alone. At the same time, the plots do not indicate that heightened frequency of either of the error types is correlated with any of the 3 spatial dimensions of Summit node location. Most locations with abnormally high error counts are due to the presence of top offending GPUs, which is in line with the phenomenon observed in [44].

Figure 4 breaks down the DBE and PRE counts by GPU placement within a Summit node. It shows that GPU slots 2 (GPU 2 on CPU 0) and 3 (GPU 0 on CPU 1) — the two geometrically central placements (see Figure 1) — have lower counts for either error type. We discuss potential reasons for this trend in Section 4.2, where it is analyzed in the context of GPU utilization.

To understand the patterns of DBE occurrence on individual GPUs, we plot the timeline of DBEs for each GPU with multiple offenses over the studied period. Figure 5 shows the instances of DBEs, PRFs, node reboots, and decommissions on a shared time scale for the 33 such GPUs, of which 11 ended up decommissioned. The operating system update on 18 Aug 2021 is shown as a Summit-wide reboot. The plot illustrates that GPUs often experience DBEs in streaks, in which every subsequent DBE is occurring within days if not hours from its predecessor on the same unit. While the mean time between subsequent DBEs on the 33 multiple-offending GPUs is almost 20 days, the median is only 20 hours, which further supports the previous observation of error temporal locality with

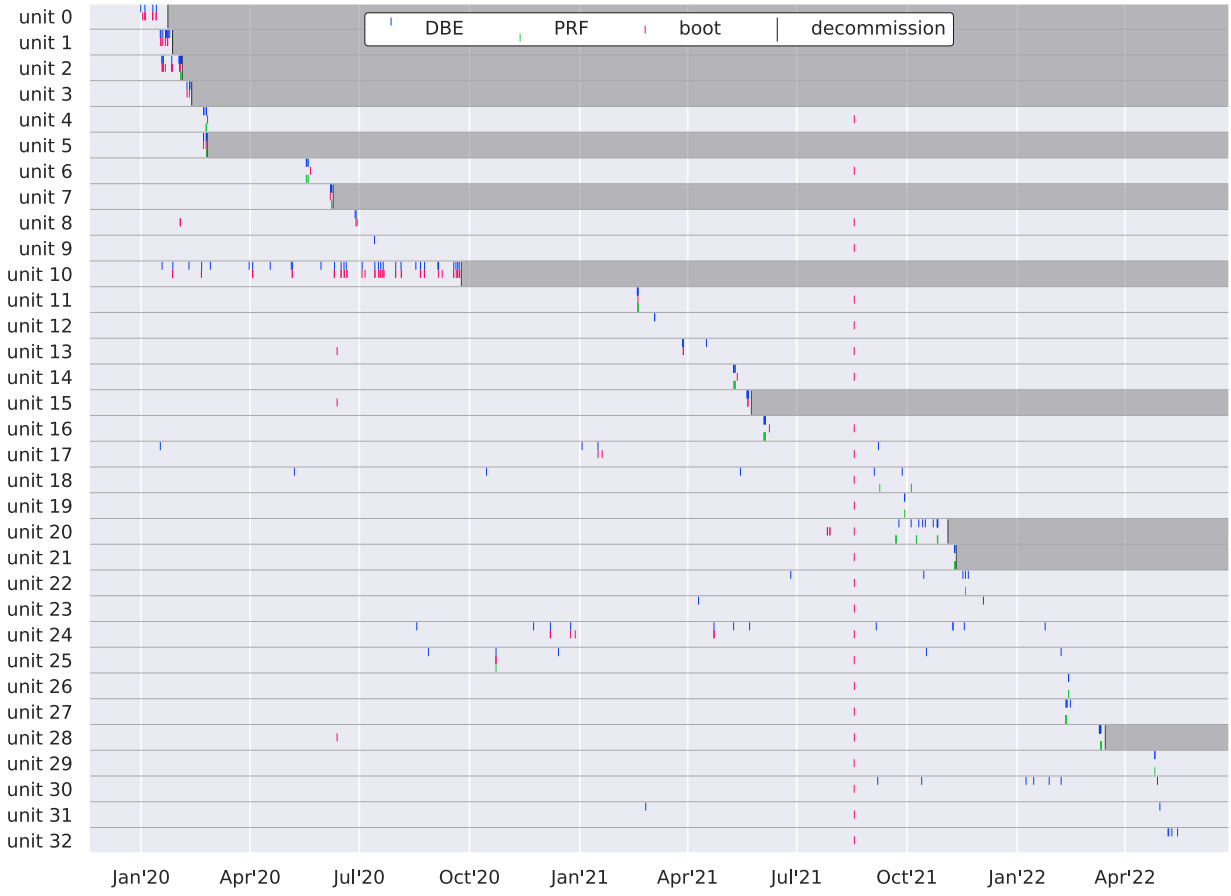


Figure 5: Timeline of memory error and failure events and reboots on GPUs with multiple DBEs.

the much older Nvidia Kepler GPUs and GDDR memory from [45]. Such small time between errors on individual units may be an outcome of reaching the 64-page retirement limit and the resulting inability to retire the offending memory page. Under such a scenario, GPUs are expected to encounter at least one PRF at the beginning of a DBE streak. While this pattern is often visible on the plot, it is not exhibited by all DBE streaks. This suggests predisposition (e.g. due to manufacturing variability i.e. the unintended differences between the manufacturing processes of individual chips) as another reason behind the contiguous susceptibility to DBEs.

Finding 2. DBEs often occur on the same GPU within days, hours, or even minutes from one another, resulting in DBE streaks that can last for weeks.

4 GPU SNAPSHOTS

A DBE occurrence on a chip is the result of both a *predisposition* of the chip to a DBE as well as the *stresses* that were put on the chip just before or accumulated over a period of time before its occurrence.

Predisposition to a DBE is a concept related to production yield, where manufactured chips are put under various stresses and classified into quality categories. Each quality category still contains a range of predispositions that differ in the amount of stress they can handle before a DBE. The predisposition may be fixed by the manufacturing process for a given memory chip but it may also change due to the accumulation of various stresses of its use.

Stresses that can lead to a DBE are various attributes and actions of the Summit architecture that affect a given GPU memory. This includes local power fluctuations, position in the local cooling architecture, job scheduling policies, and even HPC application logic. To study this, we need data on the stresses over various windows of GPU use, as we don't know the time scales over which a predisposition to a DBE can be affected.

4.1 Snapshot construction

To explore potential trends behind GPU operations associated with DBEs, and to better understand operational patterns on Summit, we constructed a dataset of "GPU snapshots" by combining the telemetry and job scheduler data with the error logs (Table 2-(b,c)). For every DBE, we collect aggregates of power consumption and temperature measurements of the offending GPU before the DBE, as well as the parameters of the job running on the respective node at the time of DBE. Building off of the time scales chosen in [24], we aggregate the telemetry over periods of 1 minute, 5 minutes, 15 minutes, 1 hour, and 6 hours preceding the DBE, as well as for the

period between 1 Jan 2020 and the error to represent lifetime GPU usage. The aggregates themselves are comprised of the minimum, maximum, range (the difference between the two extrema), average, and "fluctuation", defined as the average difference between consecutive measurements and aimed to quantify the volatility of GPU usage intensity.

Because the telemetry data lacks 30% of the expected observations and contains 15% missing values (2% and 21% of the power and thermal measurements, respectively), and because "lifetime" aggregates are taken over periods of different lengths, using the average and not the total in the fluctuation aggregates ensures that they are directly comparable among the snapshots. For the same reasons, the average aggregates are chosen as a proxy for the cumulative intensity of GPU utilization. Because GPU power and especially thermal sensors can sometimes produce faulty zero measurements, we replace these with NaNs for the purposes of aggregation.

We collect the user, project, and allocation flags of the job allocation (if any) on the hosting node at the time of snapshot. In order to decide whether the GPU has been utilized by the job, we also measure peak GPU power consumption between the job start and the snapshot.

In addition to the DBE snapshots, we also construct 50,000 "normal" snapshots by extracting the same features for randomly chosen GPUs and moments in time within the studied period. The normal snapshots serve as an analogous data representation of baseline GPU operations on Summit. To ensure no overlap between the DBE and normal GPU operations, we remove normal snapshots within 24 hours from the nearest DBE on the respective GPU.

Independence of observations (sample points) is an assumption that permeates most statistical and machine learning methods. While this assumption is rarely perfectly satisfied in practice, stronger departures can introduce bias and invalidate inference. Because subsequent DBEs on the same GPU within a short period of time may be induced by the same underlying event, we exclude them from our analysis in an effort to obtain independently occurring DBEs. To choose what constitutes a sufficiently long period after which a subsequent DBE can be deemed independent, we visualize the distribution of elapsed time between consequent DBEs on individual GPUs in Figure 6. The vast majority of subsequent DBEs occur within 2 to 5 days from their predecessor. Based on the distribution shape, we set a conservative cut-off value for independent DBEs to be 5 days. After excluding subsequent snapshots within less than 120 hours after a DBE on the same GPU, our dataset consists of 166 DBEs and 49,963 normal snapshots.

To further facilitate our analysis, we narrow down the scope to those DBEs which occur on GPUs utilized by the current job. Since the job scheduler data does not directly contain information about whether an individual GPU (or any of them) has been used during a job, let alone before a particular moment (i.e. when the snapshot was taken), we infer this information from GPU power consumption collected between the job start and the snapshot, by comparing its peak value against a data-informed threshold.

Figure 7 shows the distribution of GPU power consumption minima and maxima within Summit jobs executed between 1 Jan 2020 and 17 May 2022. The distribution is plotted based on a random sample of 1% (33K) of the jobs corresponding to 40M of GPU power

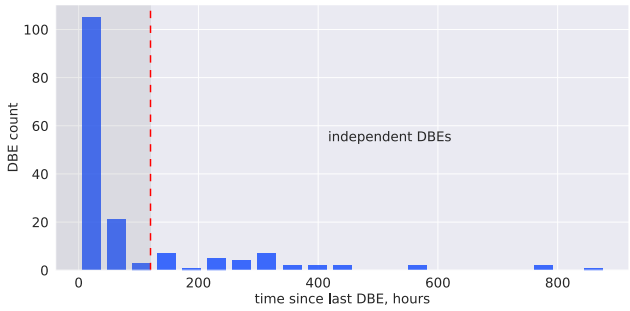


Figure 6: Distribution of elapsed time between same-GPU DBEs, with values over 900 hours truncated. Dashed line at 5 days (120 hours) shows the cut-off value to deem DBEs independent of the previous occurrence.

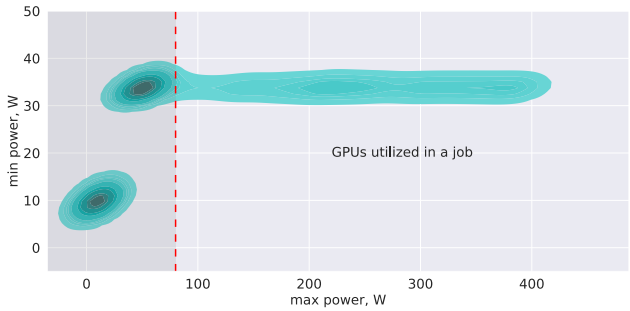


Figure 7: Distribution of GPU power consumption extrema in individual jobs. Dashed line at 80W shows the cut-off value to deem GPUs utilized by the job.

extremum pairs. Because the minima represent levels of GPU activity in the idle state, we posit that the bottom cluster represents power profiles of jobs with no GPU allocation, while the top one is comprised of jobs that requested GPUs, putting them in the state of heightened readiness. GPU peak power consumption divides the top cluster into two subclusters of idle (centered at ≈50W) and utilized GPUs. The latter cluster starts from a little over 100W, which is in line with the existing knowledge of the V100's power consumption during a typical memory-bound workload — the least power-demanding workload type [4]. Figure 7 indicates that the majority of jobs requesting GPUs end up not using them, which can be explained by the runs that never reach the GPU utilization stage e.g. due to testing purposes or code failure. For our analysis, we set the peak power consumption threshold to 80W: GPUs exceeding it during the pre-snapshot part of a job are considered "under a workload" and therefore relevant to our analysis. Using this threshold, only 22K snapshots (including 94 DBEs) see prior GPU utilization in the current job. The remaining 28K snapshots without GPU utilization by the job are only used when analyzing general patterns of GPU usage, see Section 4.2.

Finding 3. Almost a half of (independent) DBEs occurring within a job do not result from GPU utilization.

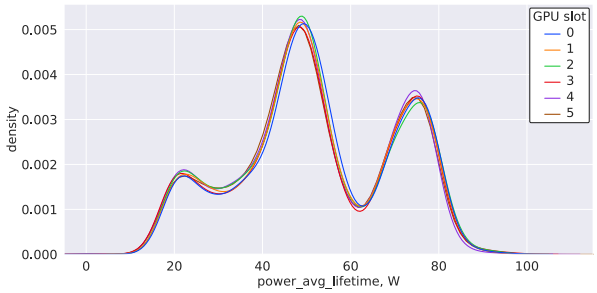


Figure 8: Distribution of lifetime GPU power consumption average by GPU placement on a node.

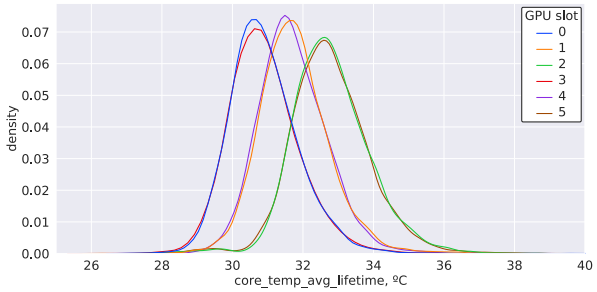


Figure 9: Distribution of lifetime GPU memory temperature average by GPU placement, with values over 40°C truncated.

4.2 Patterns of lifetime GPU utilization

To understand GPU utilization patterns on Summit independent of workload specifics, we analyze the lifetime telemetry aggregates in our snapshots. We use lifetime power consumption average for a GPU as a proxy for frequency and intensity of its overall usage. Figure 8 shows the distribution of lifetime power consumption average, grouped by GPU placement within the hosting node. Given that the aggregate is a proxy for frequency and intensity of cumulative GPU utilization, the plot suggests three distinct modes of GPU usage on Summit, corresponding to the average power consumption of around 21W, 49W, and 76W. Because this trend is near-identical for all six GPU slots, we conclude that it is an indication of consistently uneven GPU workloads assigned at the node level, likely resulting from the job scheduler policies on Summit. Analogously, Figure 9 shows the distribution of lifetime memory temperature average per GPU placement in the 50K snapshots. It demonstrates a clear dependency of GPU temperature on the order in which the coolant reaches GPU locations in two separate loops — one per CPU.

Finding 4. Summit nodes comprise of 3 groups that are consistently given uneven GPU workloads, likely due to the node-level job scheduling logic.

To put the lower DBE counts for the geometrically central GPU slots 2 and 3 from Figure 4 in the context of GPU utilization, we first perform the chi-square test of independence between the snapshot type and the GPU slot. When run on the original set of 295 DBEs and 50,000 normal snapshots (Table 3(a)), it yielded the p-value of $\approx 10^{-7}$, confirming the existence of a significant difference in

Table 3: Contingency tables counting DBE and normal snapshots across the six GPU slots.

(a) All snapshots.

	0	1	2	3	4	5
DBE	64	44	34	20	70	63
Normal	8,303	8,313	8,410	8,231	8,392	8,315

(b) After discarding snapshots within 120 hours after a DBE.

	0	1	2	3	4	5
DBE	34	28	26	17	28	33
Normal	8,303	8,313	8,409	8,231	8,392	8,315

(c) After additionally discarding snapshots without GPU utilization.

	0	1	2	3	4	5
DBE	20	16	18	9	14	17
Normal	3,716	3,699	3,676	3,730	3,734	3,623

the DBE-to-normal snapshot ratio across the GPU slots. However, merely removing snapshots within less than 120 hours after a same-GPU DBE (Table 3(b)) and re-running the test gave the p-value of 0.254, indicating no significant dependency of the DBE-to-normal snapshot ratio on the GPU slot. Further removing the 28K snapshots without GPU utilization by the current job (Table 3(c)) produced the p-value of 0.437.

To attribute the significance of the chi-squared test on the Table 3(a) data to individual GPU slots 0–5, we considered standardized residuals [3] of their DBE snapshot counts. The residuals follow the standard normal distribution and were computed as 2.33, -0.79, -2.43, -4.48, 3.17, 2.17, respectively. Because their critical value at the significance level of $\alpha = 0.05$ is ± 1.96, we conclude that GPU slots 2 and 3 are less prone to DBE streaks (but not significantly so to independent DBEs) than other GPU placements. At the same time, the two GPU slots are assigned consistently similar workloads (Figure 8) and their thermal modalities exhibit much higher resemblance to slots 5 and 0, respectively, than they do to each other (Figure 9). We conclude that the phenomenon is not linked to patterns in GPU utilization or telemetry and therefore captures resilience properties specific to the GPU physical locations.

Finding 5. Geometrically central GPU placements on a node are more resilient to DBE streaks.

To study how various lifetime aggregates relate to one another, we consider their pairwise correlations. Figure 10 visualizes the Spearman's correlation coefficients [40] between lifetime minima, averages, and fluctuation aggregates. Some of the correlations are easily explainable: the near-perfectly correlated power consumption average and fluctuation are a consequence of oscillating power measurements of an engaged GPU. The substantial correlation between the core and memory temperature aggregates follows from the shared cooling and proximity of the two GPU components. The reasons behind the significant negative correlation between the fluctuation of power and (mostly memory) temperature are not as straightforward. A potential explanation relates to the state of thermal equilibrium brought to engaged GPUs by the consistently high

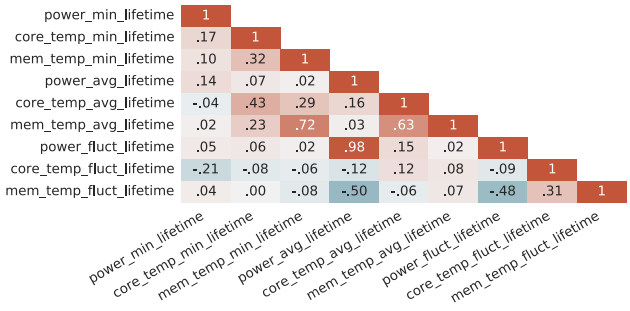


Figure 10: Correlation between lifetime telemetry aggregates.

energy intake and a steady supply of the coolant. This phenomenon has been documented in [36] (see Figures 12 and 17 therein).

5 STATISTICAL INFERENCE

Here, we only use snapshots with GPU utilization by the currently running job. Our dataset contains a total of 22K such snapshots, of which 94 are DBE snapshots.

5.1 Thermal and power usage differences

Testing the difference between population means is a staple for comparing numerical data obtained from two distinct groups. We employ this approach to see if some telemetry aggregates have significantly different means in DBE and normal snapshots, which would hint at GPU usage patterns associated with such memory errors. Specifically, we run the two-sample (unpaired) Student's t-test for each of the 90 telemetry aggregates (power/core t°/memory t° × min./max./range/avg./fluct. × 1min/5min/15min/1h/6h/lifetime). Table 4 shows all of the aggregates for which the test detects a statistically significant difference between DBE and normal snapshots at $\alpha = 0.05$. The variable in bold also has its means significantly different at the more stringent significance level of $\alpha = 0.00057$, established using the Šidák correction [38] to bound the chance of any false positives among the 90 test outcomes by 0.05.

At p-value 0.00056, the range of GPU power consumption over the 15 minutes prior to a snapshot is the only variable with a significant difference between the means for the corrected α. Its distribution in DBE and normal snapshots is shown on Figure 11(a) and demonstrates that the 15-minute power range is bigger in DBEs by on average 33W. To illustrate the associated thermal effects, we plot the distribution of 15-minute memory temperature range (Figure 11(b)), showing an average difference of less than 1.5°C between DBE and normal snapshots. The latter phenomenon is unlikely to be a cause of DBEs, given the consistent existence of a similar temperature difference between the GPUs 0 and 2 for each CPU (see Figure 9) that is not necessarily matched by an increase in their DBE occurrence. It is therefore more plausible that the high variation in short-term power intake is directly causing the heightened susceptibility of GPU memory to DBEs — or that the two phenomena share an underlying cause such as high-intensity workload patterns increasing the likelihood of cell-to-cell interference [16].

Interestingly, the significance of the difference in `power_range_15min` is considerably higher than in the variable's two components, the 15-minute power minimum (p-value 0.01722) and maximum (p-value 0.00180). The same trend is exhibited by the respective aggregates of core and memory temperatures. This suggests that DBEs correlate stronger with jumps in GPU usage intensity than with its peaks alone. Moreover, the fact that short-term telemetry averages in DBEs and normal snapshots are not significantly different even at $\alpha = 0.05$ (Table 4) means that the duration of peak power consumption is not a factor for memory corruption.

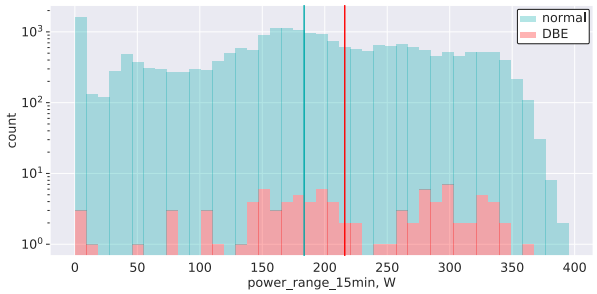
Table 4: Variables whose means in DBE and normal snapshots are significantly different at $\alpha = 0.05$. The means of the boldened variable are significantly different even at the corrected $\alpha = 0.00057$.

Variable	p-value	DBE ⩽ normal
power_min_lifetime	0.01618	>
power_avg_lifetime	0.01588	>
power_fluct_lifetime	0.01413	>
core_temp_min_lifetime	0.01033	>
core_temp_fluct_lifetime	0.01120	<
power_max_6h	0.01196	>
power_range_6h	0.01106	>
mem_temp_fluct_1h	0.04880	<
power_min_1h	0.00361	<
power_max_1h	0.00927	>
power_range_1h	0.00409	>
core_temp_max_1h	0.02344	>
core_temp_range_1h	0.00239	>
mem_temp_max_1h	0.02848	>
mem_temp_range_1h	0.00803	>
power_min_15min	0.01722	<
power_max_15min	0.00180	>
power_range_15min	**0.00056**	>
core_temp_max_15min	0.04717	>
core_temp_range_15min	0.00381	>
mem_temp_range_15min	0.02683	>
power_max_5min	0.00274	>
power_range_5min	0.00124	>
core_temp_range_5min	0.02453	>
power_max_1min	0.00271	>
power_range_1min	0.00239	>
core_temp_max_1min	0.04593	>

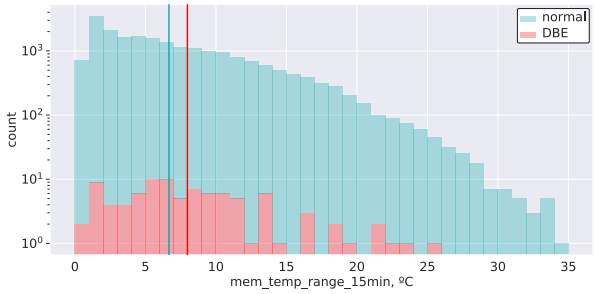
The pattern of higher power and thermal maxima in DBEs, representing more intense GPU utilization, is also visible on other short-term time scales (Table 4). While power consumption and temperature of an engaged GPU are closely correlated, the lower significance of the differences in thermal rather than power extremities is consistent with the conclusion previously made in [35]: the correlation between temperature and DRAM errors is indirect and vanishes when controlling for utilization. This in turn contrasts with the suggestion of a causal link between high temperature and GPU memory corruption made in [45].

> **Finding 6.** DBEs are associated with recent intensive GPU utilization characterized by substantial changes in power intake over short periods of time. The association with elevated temperatures is minor and likely a consequence of the above.

Furthermore, the fact that the difference in power maximum is more pronounced at the 15-minute scale than at 5- or at 1-minute scale indicates that the highest intensity of GPU utilization is typically observed further in the past and not immediately before the error. Such a delay is likely caused by the gap between the corruption of GPU memory and its detection caused by accessing it.



(a) Power consumption range over the last 15 minutes.



(b) Memory temperature range over the last 15 minutes.

Figure 11: Distribution of telemetry aggregates in DBE and normal snapshots. Vertical lines show the distribution means.

Finding 7. The time between GPU memory corruption and its detection in an HPC application can span minutes or even tens of minutes.

While no lifetime telemetry aggregates show a significant difference between DBE and normal snapshots at $\alpha = 0.00057$, the relationships between those with detectable differences below $\alpha = 0.05$ allow for interpretations. In particular, we established that GPU temperature fluctuation is inversely proportional to the workload. It implies that `core_temp_fluct_lifetime` and `power_avg_lifetime` both reflect the long-term frequency and intensity of GPU utilization, with respective p-values of 0.01120 and 0.01588. Because the two variables are not strongly correlated (Figure 10), the probability of their detected differences both being a false positive is not much higher than $0.01120 \cdot 0.01588 = 0.00018$, meaning that at $\alpha = 0.00057$ we can conclude they jointly are different between the DBE and normal snapshots. This suggests that, analogously to a car's odometer reading being correlated to its likelihood of a malfunction, frequent utilization of a GPU over its lifetime makes it more susceptible to future memory corruption events. Of note is the absence of lifetime temperature averages in Table 4, further supporting the idea that high temperatures are not a significant factor behind the DBEs. This is consistent with a recent finding showing low dependence of DBEs on temperature on the Titan supercomputer [20].

Another potentially related pair of lifetime aggregates from Table 4 is the lifetime minima of GPU power consumption (p-value 0.01618) and core temperature (p-value 0.01033), both reflecting the level of GPU activity in the idle state. Using the same reasoning as above, we deduce that at least one of them is significantly elevated in the population of GPUs experiencing a DBE. It implies that a GPU's susceptibility to memory corruption is associated with its permanently higher baseline activity levels. Because the elevated lifetime power and temperature minima cannot result from stresses of using the GPU (as they do not affect the initial GPU activity after being installed), we hypothesize that the phenomenon is likely caused by manufacturing variability. The latter might be responsible for the higher DBE rate either directly or through incurring more wear on the units (i.e. pushing their "odometer" through permanently elevated power consumption or temperature).

Finding 8. GPU susceptibility to future DBEs increases with the frequency and intensity of its lifetime activity. Manufacturing variability might be a factor behind the mildly raised activity levels in some GPUs.

5.2 Snapshot classification

Unlike the statistical tests that analyze one variable at a time, machine learning models can provide more insights into the relevancy of combinations of features. We apply interpretable classification methods for differentiating between DBE and normal snapshots based on telemetry features. To prepare the data, we drop rows with missing telemetry data and subsample the non-DBE snapshots to help with the class imbalance, resulting in 92 DBE and 184 normal snapshots. To assess classification performance, we present modeling results in terms of area under the receiver operating characteristic curve (ROC AUC). A ROC AUC value closer to 1 is better. We evaluate the results from applying naïve Bayes, logistic regression, linear support vector machine, and random forest. All models are trained using a 70/30 split for training and testing respectively.

When performing classification on the subsampled snapshot data set, all methods initially perform poorly. The second column of Table 5 shows the classification performance of various predictors using the entire data set. Random forest, which is known to be a good method for failure classification in other studies [5, 30], has a ROC AUC of 0.53. The third column shows the classification performance when limiting the dataset to the snapshots of GPUs with prior DBE encounters (39 DBE snapshots vs 45 normal), which indicates predisposition. The noticeable improvement in the model accuracy for these snapshots is consistent with other machine learning predictors of GPU memory corruption, trained on SBEs [24].

Finding 9. Predicting GPU utilization-linked DBEs from telemetry data using interpretable models is much easier for previously offending GPUs. This suggests that the stress factors in already predisposed units are distinct from those in the general GPU population.

Random forests are known to perform well even on small sample sizes (e.g. 30 observations) [18, 21, 32], even when the variables significantly outnumber the observations [7]. Given that the random forest offers superior prediction performance with previously

Table 5: DBE prediction on all GPUs and GPUs which have had a DBE in the past.

Method	ROC AUC (all GPUs)	ROC AUC (previously offending GPUs)
Bernoulli Naïve Bayes	0.56	0.74
Gaussian Naïve Bayes	0.59	0.54
Complement Naïve Bayes	0.63	0.69
Multinomial Naïve Bayes	0.63	0.69
Logistic Regression	0.56	0.76
Linear Support Vector Machine	0.57	0.76
Random Forest	0.53	**0.84**

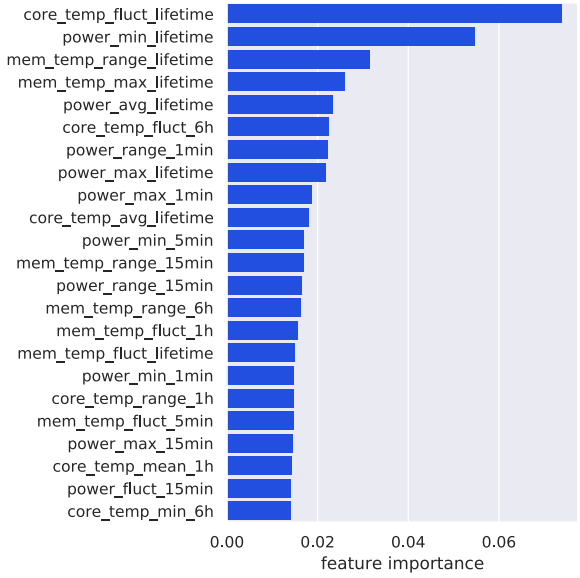


Figure 12: Feature importance (Mean Decrease in Impurity) in the random forest model.

offending GPUs, we turn to this model for insights. To this end, we apply feature elimination strategies to remove unimportant features which may detract from the accuracy of the model. We start by ranking the features by their importance based on the associated mean decrease in impurity (MDI) in the trained model, see Fig. 12. MDI quantifies the decrease in variance associated with splitting on a feature (in particular, it is the R^2 in a linear regression of the response on the output of one-level decision trees that split on this feature [2]) and therefore serves as a measure of feature importance. Using MDI-based ranking, we iteratively build a feature set by adding the most important features one at a time. We start with a model using only `core_temp_fluct_lifetime`, followed by a model using both `core_temp_fluct_lifetime` and `power_min_lifetime`, etc. Interestingly, using only the first three features yields a model with 0.9166 ROC AUC. We note that these three features are lifetime aggregates, indicating that the predisposition of a GPU is more informative for DBE prediction than its short-term stressors.

Finding 10. Lifetime features are the most critical to the predication of DBEs on susceptible GPUs, further supporting the idea of importance of predisposition among DBE factors.

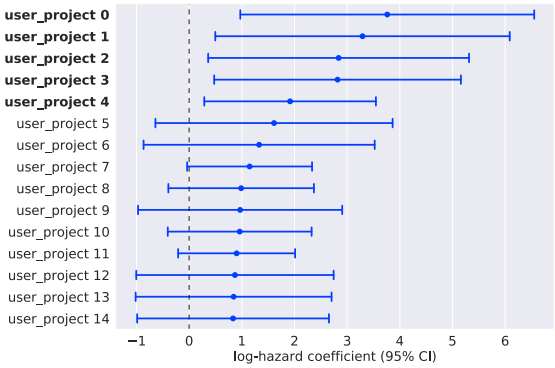


Figure 13: Top-15 user-project combinations by estimated DBE hazard coefficient. DBE-susceptible combinations (log-transformed coefficient above 0 at $\alpha = 0.05$) are in bold.

5.3 Effect of workload patterns

To study the effects of HPC workloads on DBE susceptibility in GPUs, we use an analogy from medical statistics in which the GPU memory chips are *patients* and the jobs running on them are their *treatments*. The patients arrive (have the respective job started) and are observed until the moment of a snapshot, which represents either an outcome (DBE) or an absence of such (normal snapshot). Such a setup enables fitting a *Cox regression* model, a common technique in survival analysis [9], to the snapshot data to quantify the DBE susceptibility associated with various workload types. Cox regression estimates the hazard coefficients that apply to the baseline DBE likelihood to represent the risk associated with running each HPC application (among those captured by our snapshots). The applications where the left endpoint of the 95% confidence interval for the respective hazard coefficient exceeds 1 (or 0 after the log-transformation) are considered DBE-susceptible. Cox regression uses the time elapsed in a job before the snapshot, which carries information about the absence of DBEs during this period, to infer how the baseline DBE likelihood changes over time. The type of HPC application was chosen as the only predictor in the model because, unlike most of our variables, it stays fixed throughout the observation period, which is less demanding to the number of observations required to capture its effects. To reduce the chance for the observed effects to be caused by predisposition to DBEs of a particular GPU, we remove from the analysis the snapshots of GPUs that have previously experienced a DBE.

As in Section 3, we circumvent the lack of information about HPC application in the job scheduler records by using the project-user combination as its proxy. Each project-user combination corresponds to a group of snapshots taken within the jobs run by this user and within this project. To reduce the likelihood of obtaining spurious results, we remove the user-project combinations with 10 or fewer snapshots.

The resulting dataset contains 49 DBE and 20K normal snapshots, corresponding to 270 distinct user-project combinations. After representing these combinations with binary dummy variables, we fit the model with the l_2-penalization of 0.01 using its implementation from the Python library *lifelines* [11]. Our analysis identifies 5 DBE-susceptible user-project combinations, whose jobs have a

higher chance of encountering a DBE per unit of time than a randomly chosen job at the significance level of $\alpha = 0.05$. Their 95% confidence intervals for the log-transformed hazard coefficients, along with the confidence intervals for the next 10 user-project combinations by estimated DBE hazard, are shown on Figure 13.

The 5 DBE-susceptible combinatons altogether correspond to 134 snapshots (0.7% of the dataset), of which 6 were DBE snapshots (12.2% of the DBE subset). Based on the names of respective jobs, their projects, and information provided by some of their users, we identify the types of HPC applications corresponding to 4 out of 5 DBE-susceptible combinations.

Finding 11. Out of 5 identified HPC applications on Summit with statistically significant DBE susceptibility, at least 4 use mixed-precision arithmetic.

6 RELATED WORK

A number of previous field studies have employed statistical approaches to understand memory corruption, including in GPUs. [43] and [41] identified the effects of vendor and cosmic ray exposure on the bit flip rate in DDR3 DRAM, and [19] demonstrated that memory architecture and workload patterns are also factors of significance. Recent works have studied the memory error in DDR4 memory, in particular the error rate variance by vendor, the distribution of single-bit vs multiple-bit errors [6], and the effects of temperature and unit position within the HPC system [14].

Log sequence analysis has been used to predict systems failures including GPU errors [10]. The physical location for survival analysis of GPUs in large scale HPC facilities is also identified as important [28]. Large system characterization studies have focused on the exploration of events that could lead to single-bit errors [44, 45] and have identified features such as temperature, power, workload, and location to drive machine learning models for prediction [22].

However, none of these works correlates bit flips in GPU memory with GPU utilization patterns at multi-scale time resolution. While GPU temperature within 1 hour before a DBE is considered in [22], its analysis is performed manually and lacks certainty in establishing the connection between high temperature and memory corruption. This is complemented by our work that quantifies the confidence in such a connection and contrasts it with a significantly stronger association between power consumption and DBEs. The approaches of error classification and survival analysis taken in our paper are similar to SBE data in [23, 24] and [28], but use significantly different features, in particular telemetry aggregates at fine-grained time resolution enabled by DBE data. Unlike [23, 24], we focus on using interpretable classification models similar to those used to understand for deep characterization of the reasons that lead to DBEs. In contrast to [28], our survival analysis identifies HPC applications associated with memory corruption. In addition, the above large-scale studies focus on GDDR5 units, whereas our findings pertain to more contemporary GPUs with high-bandwith memory.

Beyond system-focused studies that characterize the conditions that trigger GPU errors, another line of research focuses on how to characterize GPU application resilience in the presence of single-bit and multi-bit faults. Fault injection tools facilitate this characterization [13, 15, 25, 34, 46, 47] and they operate by injecting faults at the software level, microarchitecture level, or into low-level SASS instructions. Application resilience to DBEs is the subject of many works [33, 49] and has focused on the sensitivity of application resilience to input parameters [31, 50] and software hardening techniques to improve application resilience[8, 48].

7 DISCUSSION AND CONCLUSIONS

Our analyses identified a number of operation patterns in Nvidia HBM2 GPUs associated with the increased risk of DBEs, such as physical placement on a node, stresses from HPC operations, and individual predisposition. The affected GPUs seemed to be more susceptible to DBEs even if all their compromised memory pages were successfully replaced with spare ones. Due to the seemingly low-level physical nature of the patterns reported in Findings 3, 5, 6, 8, 9, 10, and 11, we believe that they are likely to manifest in other HBM2 GPUs.

The rarity of DBEs together with the unavailability of temporal SBE data for Summit has been a significant impediment to this study. At the same time, the knowledge of precise timing of DBE detection together with the scale of analyzed data have enabled fine-grained analyses resulting in novel findings about GPU reliability.

Another limitation is the lack of control for the intensity of memory operations in a job, caused by unavailability of this information. All other things being equal, a job using more GPU memory will have a higher likelihood of catching a DBE due to invoking a larger number of ECC checks. If known, the amount of GPU memory used by a job should be used to scale down (up) the weight of individual DBE (normal) observations to amplify the effects of other, less-understood factors behind the DBE occurrence.

Given the limited amount of information contained in our data and correlations between some of the discovered patterns, the question about the root cause(s) of DBEs remains open. Establishing causality in GPU memory corruption would require more data (both in the number of observations and the amount of information about them) or a controlled testing environment.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their constructive comments and valuable feedback. This work was supported by, and used the resources of, the Oak Ridge Leadership Computing Facility, located in the National Center for Computational Sciences at ORNL, managed by UT Battelle, LLC for the U.S. DOE (under the contract No. DE-AC05-00OR22725). The publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid up, irrevocable, world-wide license to publish or reproduce the published form of the manuscript, or allow others to do so, for U.S. Government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan). Smirni and Schmedding were partially supported by National Science Foundation IIS-2130681.

REFERENCES

[1] [n. d.]. NVidia XID errors. https://docs.nvidia.com/deploy/xid-errors/index.html.

[2] Abhineet Agarwal, Ana M Kenney, Yan Shuo Tan, Tiffany M Tang, and Bin Yu. 2023. MDI+: A Flexible Random Forest-Based Feature Importance Framework. *arXiv preprint arXiv:2307.01932* (2023).

[3] Alan Agresti. 2012. *Categorical data analysis*. Vol. 792. John Wiley & Sons.

[4] Ghazanfar Ali, Sridutt Bhalachandra, Nicholas Wright, Alan Sill, and Yong Chen. 2020. Evaluation of power controls and counters on general-purpose Graphics Processing Units (GPUs). (2020).

[5] Jacob Alter, Ji Xue, Alma Dimnaku, and Evgenia Smirni. 2019. SSD failures in the field: symptoms, causes, and prediction models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2019, Denver, Colorado, USA, November 17-19, 2019*. 75:1–75:14. https://doi.org/10.1145/3295500.3356172

[6] Majed Valad Beigi, Yi Cao, Sudhanva Gurumurthi, Charles Recchia, Andrew Walton, and Vilas Sridharan. 2023. A Systematic Study of DDR4 DRAM Faults in the Field. In *2023 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. IEEE, 991–1002.

[7] Anne-Laure Boulesteix, Silke Janitza, Jochen Kruppa, and Inke R König. 2012. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 2, 6 (2012), 493–507.

[8] Zitao Chen, Guanpeng Li, and Karthik Pattabiraman. 2021. A low-cost fault corrector for deep neural networks through range restriction. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 1–13.

[9] David R Cox. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 34, 2 (1972), 187–202.

[10] Anwesha Das, Frank Mueller, and Barry Rountree. 2020. Aarohi: Making Real-Time Node Failure Prediction Feasible. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. IEEE, 1092–1101.

[11] Cameron Davidson-Pilon. 2019. lifelines: survival analysis in Python. *Journal of Open Source Software* 4, 40 (2019), 1317. https://doi.org/10.21105/joss.01317

[12] Jack J. Dongarra, Hans W. Meuer, and Erich Strohmaier. [n. d.]. *Top500*. Retrieved May 7, 2019 from https://www.top500.org/

[13] Bo Fang, Karthik Pattabiraman, Matei Ripeanu, and Sudhanva Gurumurthi. 2014. Gpu-qin: A methodology for evaluating the error resilience of gpgpu applications. In *2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. IEEE, 221–230.

[14] Kurt B Ferreira, Scott Levy, Joshua Hemmert, and Kevin Pedretti. 2022. Understanding memory failures on a petascale Arm system. In *Proceedings of the 31st International Symposium on High-Performance Parallel and Distributed Computing*. 84–96.

[15] Siva Kumar Sastry Hari, Timothy Tsai, Mark Stephenson, Stephen W Keckler, and Joel Emer. 2015. SASSIFI: Evaluating resilience of GPU applications. In *Proceedings of the Workshop on Silicon Errors in Logic-System Effects*.

[16] Yoongu Kim, Ross Daly, Jeremie Kim, Chris Fallin, Ji Hye Lee, Donghyuk Lee, Chris Wilkerson, Konrad Lai, and Onur Mutlu. 2014. Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors. *ACM SIGARCH Computer Architecture News* 42, 3 (2014), 361–372.

[17] Jamie Liu, Ben Jaiyen, Yoongu Kim, Chris Wilkerson, and Onur Mutlu. 2013. An experimental study of data retention behavior in modern DRAM devices: Implications for retention time profiling mechanisms. *ACM SIGARCH Computer Architecture News* 41, 3 (2013), 60–71.

[18] Jing Luan, Chongliang Zhang, Binduo Xu, Ying Xue, and Yiping Ren. 2020. The predictive performances of random forest models with limited sample size and different species traits. *Fisheries Research* 227 (2020), 105534.

[19] Justin Meza, Qiang Wu, Sanjeev Kumar, and Onur Mutlu. 2015. Revisiting memory errors in large-scale production data centers: Analysis and modeling of new trends from the field. In *2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. IEEE, 415–426.

[20] Jie Min, Yili Hong, William Q. Meeker, and George Ostrouchov. 2023. A Spatially Correlated Competing Risks Time-to-Event Model for Supercomputer GPU Failure Data. arXiv:2303.16369 [stat.AP]

[21] Daniel Moraes, Pedro Benevides, Hugo Costa, Francisco D Moreira, and Mário Caetano. 2021. Influence of Sample Size in Land Cover Classification Accuracy Using Random Forest and Sentinel-2 Data in Portugal. In *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*. IEEE, 4232–4235.

[22] Bin Nie, Devesh Tiwari, Saurabh Gupta, Evgenia Smirni, and James H Rogers. 2016. A large-scale study of soft-errors on GPUs in the field. In *IEEE International Symposium on High Performance Computer Architecture (HPCA'16)*. IEEE, 519–530.

[23] Bin Nie, Ji Xue, Saurabh Gupta, Christian Engelmann, Evgenia Smirni, and Devesh Tiwari. 2017. Characterizing temperature, power, and soft-error behaviors in data center systems: Insights, challenges, and opportunities. In *IEEE 25th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS'17)*. IEEE, 22–31.

[24] Bin Nie, Ji Xue, Saurabh Gupta, Tirthak Patel, Christian Engelmann, Evgenia Smirni, and Devesh Tiwari. 2018. Machine learning models for GPU error prediction in a large scale HPC system. In *48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'18)*. IEEE, 95–106.

[25] Bin Nie, Lishan Yang, Adwait Jog, and Evgenia Smirni. 2018. Fault Site Pruning for Practical Reliability Analysis of GPGPU Applications. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 749–761.

[26] NVIDIA Corporation. [n. d.]. *Dynamic Page Retirement, Technical Note*. Retrieved July 10, 2021 from https://docs.nvidia.com/deploy/dynamic-page-retirement/index.html

[27] Daniel Oliveira, Sean Blanchard, Nathan DeBardeleben, Fernando Fernandes dos Santos, Gabriel Piscoya Dávila, Philippe O. A. Navaux, Andrea Favalli, Opale Schappert, Stephen Wender, Carlo Cazzaniga, Christopher Frost, and Paolo Rech. 2021. Thermal neutrons: a possible threat for supercomputer reliability. *J. Supercomput.* 77, 2 (2021), 1612–1634. https://doi.org/10.1007/s11227-020-03324-9

[28] George Ostrouchov, Don Maxwell, Rizwan Ashraf, Christian Engelmann, Mallikarjun Shankar, and James Rogers. 2020. GPU lifetimes on Titan supercomputer: Survival analysis and reliability. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis (SC'20)*. 15–20. https://doi.org/10.1109/SC41405.2020.00045

[29] Minesh Patel, Jeremie S Kim, and Onur Mutlu. 2017. The reach profiler (reaper) enabling the mitigation of dram retention failures via profiling at aggressive conditions. *ACM SIGARCH Computer Architecture News* 45, 2 (2017), 255–268.

[30] Riccardo Pinciroli, Lishan Yang, Jacob Alter, and Evgenia Smirni. 2023. Lifespan and Failures of SSDs and HDDs: Similarities, Differences, and Prediction Models. *IEEE Trans. Dependable Secur. Comput.* 20, 1 (2023), 256–272. https://doi.org/10.1109/TDSC.2021.3131571

[31] Fritz G Previlon, Charu Kalra, David R Kaeli, and Paolo Rech. 2018. Evaluating the impact of execution parameters on program vulnerability in gpu applications. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 809–814.

[32] Yanjun Qi. 2012. Random forest for bioinformatics. In *Ensemble machine learning: Methods and applications*. Springer, 307–323.

[33] Behrooz Sangchoolie, Karthik Pattabiraman, and Johan Karlsson. 2017. One Bit is (Not) Enough: An Empirical Study of the Impact of Single and Multiple Bit-Flip Errors. In *Dependable Systems and Networks (DSN), 2017 47th Annual IEEE/IFIP International Conference on*. IEEE, 97–108.

[34] Dimitris Sartzetakis, George Papadimitriou, and Dimitris Gizopoulos. 2022. gpuFI-4: A Microarchitecture-Level Framework for Assessing the Cross-Layer Resilience of Nvidia GPUs. In *2022 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*. 35–45. https://doi.org/10.1109/ISPASS55109.2022.00004

[35] Bianca Schroeder, Eduardo Pinheiro, and Wolf-Dietrich Weber. 2011. DRAM errors in the wild: a large-scale field study. *Commun. ACM* 54, 2 (2011), 100–107.

[36] Woong Shin, Vladyslav Oles, Ahmad Maroof Karimi, J Austin Ellis, and Feiyi Wang. 2021. 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*. 1–14. https://doi.org/10.1145/3458817.3476188

[37] Woong Shin, Vladyslav Oles, Anna Schmedding, George Ostrouchov, Evgenia Smirni, Christian Engelmann, and Feiyi Wang. 2023. OLCF Summit Supercomputer GPU Snapshots During Double-Bit Errors and Normal Operations. (4 2023). https://doi.org/10.13139/OLCF/1970187

[38] Zbyněk Šidák. 1967. Rectangular confidence regions for the means of multivariate normal distributions. *J. Amer. Statist. Assoc.* 62, 318 (1967), 626–633.

[39] Charles Slayman. 2011. *JEDEC Standards on Measurement and Reporting of Alpha Particle and Terrestrial Cosmic Ray Induced Soft Errors*. Springer US, Boston, MA, 55–76. https://doi.org/10.1007/978-1-4419-6993-4_3

[40] Charles Spearman. 1961. The proof and measurement of association between two things. (1961).

[41] Vilas Sridharan, Nathan DeBardeleben, Sean Blanchard, Kurt B Ferreira, Jon Stearley, John Shalf, and Sudhanva Gurumurthi. 2015. Memory errors in modern systems: The good, the bad, and the ugly. *ACM SIGARCH Computer Architecture News* 43, 1 (2015), 297–310.

[42] Vilas Sridharan and Dean Liberty. 2012. A study of DRAM failures in the field. In *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 1–11. https://doi.org/10.1109/SC.2012.13

[43] Vilas Sridharan, Jon Stearley, Nathan DeBardeleben, Sean Blanchard, and Sudhanva Gurumurthi. 2013. Feng shui of supercomputer memory: Positional effects in DRAM and SRAM faults. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 1–11.

[44] Devesh Tiwari, Saurabh Gupta, George Gallarno, Jim Rogers, and Don Maxwell. 2015. Reliability lessons learned from GPU experience with the Titan supercomputer at Oak Ridge Leadership Computing Facility. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'15)*. IEEE, 1–12.

[45] Devesh Tiwari, Saurabh Gupta, James Rogers, Don Maxwell, Paolo Rech, Sudharshan Vazhkudai, Daniel Oliveira, Dave Londo, Nathan DeBardeleben, Philippe Navaux, et al. 2015. Understanding GPU errors on large-scale HPC systems and the implications for system design and operation. In *IEEE 21st International Symposium on High Performance Computer Architecture (HPCA'15)*. IEEE, 331–342.

[46] Timothy Tsai, Siva Kumar Sastry Hari, Michael Sullivan, Oreste Villa, and Stephen W Keckler. 2021. Nvbitfi: dynamic fault injection for gpus. In *2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE, 284–291.

[47] Alessandro Vallero, Dimitris Gizopoulos, and Stefano Di Carlo. 2017. SIFI: AMD southern islands GPU microarchitectural level fault injector. In *2017 IEEE 23rd International Symposium on On-Line Testing and Robust System Design (IOLTS)*. IEEE, 138–144.

[48] Lishan Yang, Bin Nie, Adwait Jog, and Evgenia Smirni. 2021. Enabling Software Resilience in GPGPU Applications via Partial Thread Protection. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 1248–1259.

[49] Lishan Yang, Bin Nie, Adwait Jog, and Evgenia Smirni. 2021. Practical Resilience Analysis of GPGPU Applications in the Presence of Single- and Multi-Bit Faults. *IEEE Trans. Comput.* 70, 1 (2021), 30–44.

[50] Lishan Yang, Bin Nie, Adwait Jog, and Evgenia Smirni. 2021. SUGAR: Speeding Up GPGPU Application Resilience Estimation with Input Sizing. *Proc. ACM Meas. Anal. Comput. Syst.* 5, 1 (2021), 01:1–01:29. https://doi.org/10.1145/3447375