Do AI Workloads Fail Differently? A Cross-Cluster Characterization of Failure Types in AI and HPC Systems

Trupesh Patel
Department of Computer Science
The University of Alabama
Tuscaloosa, AL, USA
trpatel2@crimson.ua.edu

Purushotham V. Bangalore
Department of Computer Science
The University of Alabama
Tuscaloosa, AL, USA
pvbangalore@ua.edu

I. Motivation

Two communities study why large computations fail, and they rarely read each other's evidence. High-performance computing has characterized system failures for two decades, concluding that user and application errors dominate failure counts on production supercomputers [1], [2]. Machine learning systems teams, training models synchronously on tens of thousands of GPUs, now report interruptions every few hours and attribute most of them to hardware [3], [4]. Whether these findings conflict is unclear, because the evidence is published in incompatible forms: some studies attribute failures to causes, many expose only scheduler labels such as FAILED or TIMEOUT, and others publish only rates from which no failure share can be derived. This poster normalizes that evidence across communities and answers a question the exascale era left open: hardware failure rates did not explode as predicted, but job-level exposure did.

II. Approach

We surveyed 162 publicly described datasets and operational reports on failures in HPC and AI workloads, accounting for every one [5]: 20 provide cause-attributed failure shares, 25 provide outcome labels only, 44 provide rates only, 36 contain no failure information, and the remaining 37 are duplicates, umbrella repositories, or paper-only statistics. From 931 enumerated failure-type mentions across 37 studies we built an eleven-class taxonomy in which every canonical failure type is tagged as a root cause, an observable symptom, or an outcome label. The distinction is central: a third of published failure mentions are outcome labels with no causal content, and folding them into cause categories inflates the user-error class. All values passed three independent verification passes (source-quote checking, blind re-derivation, adjudication), and every matrix cell traces to a quoted source statement.

III. The Failure Landscape

Our first result is a pair of denominator-disciplined failure maps. Across the twenty systems publishing cause-attributed shares, traditional HPC failure counts are dominated by user and application errors (99% on Mira [2]; 33–48% on university clusters [6]), while AI-training interruption costs are dominated by GPU hardware (59% of Llama-3 405B interruptions [3]; 78% hardware overall in both Meta reports [3], [4]). Per-job failure rates, however, are similar across communities where a controlled comparison exists (17% for ML jobs versus 14% for generic jobs on the same cluster [7]). The divergence is exposure: mean time to failure collapses from 47.7 days for 8-GPU jobs to 7.9 hours at 1,024 GPUs [4]. A second asymmetry separates counts from cost: infrastructure failures are rare by count but dominate lost compute (82% of failed-job GPU-time at Acme [8]), while walltime timeout alone consumes 33–44% of delivered compute on five HPC systems [2], [6], [9]. Operational maturity is a hidden variable: on comparable workloads, system-caused failure share differs by 13$\times$ between a university community cluster and a professionally operated one [6].

IV. GPU Errors Across Five Generations

We then characterized GPU error classes across twelve systems spanning K20X to B200. Three findings order fault-tolerance priorities. First, interconnect errors, not compute or memory, recur as the top attributable hardware class across sources: NVLink and off-the-bus/PCIe errors account for 43% of raw Xid events on a 10,000-GPU A100 fleet [10], 29% of failures on a recent B200 cluster [11], and the largest single interruption category of the Llama-3 run [3]. Second, uncorrectable memory errors are rare but uniformly fatal and worsening: at the same site with identical telemetry, H100 uncorrectable-memory mean time between errors is 3.2$\times$ worse than A100, and row-remapping fails in 41% of uncorrectable cases [12]. Third, the largest raw error category misleads: MMU/illegal-access events are 60–95% of critical Xids at NCSA Delta [12] but are predominantly application bugs wearing a hardware label, contaminating any failure-prediction model trained naively on error logs. GSP/driver errors collapsed from 26% to 0.16% across the A100-to-H100 transition [12], and correctable-error telemetry (row-

remap counts) predicts subsequent degradation [13], making it a prediction feature rather than a failure class.

V. Contributions and Artifact

The poster presents the survey statistics, the taxonomy with per-type rationale, both failure maps, the GPU error-class matrix ordered by generation, and a remediation-responsibility mapping grounded in vendor and operator documentation [14], [15]: application developers own illegal-access classes, operators own service restoration, and the vendor owns durable fixes under published replacement thresholds. None of the vendor or site documentation we examined assigns recovery to the middleware layer for any failure class, and no surveyed dataset couples job failures to verified root causes; together these define the community's most consequential gaps. All matrices, per-cell quotations, and the survey table are released as an artifact (Appendix A).

Acknowledgment

Text drafting, literature extraction, and figure preparation were assisted by Claude (Anthropic) [16], used as a general-purpose writing and analysis assistance tool under the authors' direction. The authors verified all quantitative claims against the cited published sources and take complete accountability for the contents of this work.

References

[1] B. Schroeder and G. A. Gibson, "A large-scale study of failures in high-performance computing systems," in *Proc. Int'l Conf. on Dependable Systems and Networks (DSN)*, 2006, pp. 249–258.

[2] S. Di, H. Guo, E. Pershey, M. Snir, and F. Cappello, "Characterizing and understanding HPC job failures over the 2K-day life of IBM BlueGene/Q system," in *Proc. Int'l Conf. on Dependable Systems and Networks (DSN)*, 2019, pp. 473–484.

[3] A. Grattafiori *et al.*, "The Llama 3 herd of models," arXiv preprint arXiv:2407.21783, 2024.

[4] A. Kokolis, M. Kuchnik, J. Hoffman, A. Kumar, P. Malani, F. Ma, Z. DeVito, S. Sengupta, K. Saladi, and C.-J. Wu, "Revisiting reliability in large-scale machine learning research clusters," in *Proc. IEEE Int'l Symp. on High-Performance Computer Architecture (HPCA)*, 2025, pp. 1259–1274, arXiv:2410.21680.

[5] T. Patel and P. V. Bangalore, "HPC-AI failure survey: Evidence artifact," https://github.com/TrupeshKumarPatel/hpc-ai-failure-survey, 2026.

[6] R. Kumar, S. Jha, A. Mahgoub, R. Kalyanam, S. Harrell, X. C. Song, Z. Kalbarczyk, W. Kramer, R. Iyer, and S. Bagchi, "The mystery of the failing jobs: Insights from operational data from two university-wide computing systems," in *Proc. Int'l Conf. on Dependable Systems and Networks (DSN)*, 2020, pp. 158–171.

[7] X. Chu, D. Hofstätter, S. Ilager, S. Talluri, D. Kampert, D. Podareanu, D. Duplyakin, I. Brandic, and A. Iosup, "Generic and ML workloads in an HPC datacenter: Node energy, job failures, and node-job analysis," 2024, arXiv:2409.08949.

[8] Q. Hu, Z. Ye, Z. Wang, G. Wang, M. Zhang, Q. Chen, P. Sun, D. Lin, X. Wang, Y. Luo, Y. Wen, and T. Zhang, "Characterization of large language model development in the datacenter," in *Proc. USENIX Symp. on Networked Systems Design and Implementation (NSDI)*, 2024, pp. 709–729.

[9] K. Menear, A. Nag, J. Perr-Sauer, M. Lunacek, K. Potter, and D. Duplyakin, "Mastering HPC runtime prediction: From observing patterns to a methodological approach," in *Proc. Practice and Experience in Advanced Research Computing (PEARC)*, 2023.

[10] W. An, X. Bi, G. Chen, S. Chen, C. Deng, H. Ding, K. Dong, Q. Du, W. Gao, K. Guan *et al.*, "Fire-Flyer AI-HPC: A cost-effective software-hardware co-design for deep learning," in *Proc. Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC)*, 2024, arXiv:2408.14158.

[11] D. Kang, E. Hwang, H. Lee, H. Kim, H. Koo, J. Shin, J. Kang, J. Kang, J. Heo, J. Kim, J. Lee, J. Yang, K. Cho, and Y. Song, "From detection to recovery: Operational analysis on LLM pre-training with 504 GPUs," arXiv preprint arXiv:2605.09370, 2026.

[12] S. Cui, A. Patke, H. Nguyen, A. Ranjan, Z. Chen, P. Cao, G. Bauer, B. Bode, C. Di Martino, S. Jha, C. Narayanaswami, D. Sow, Z. T. Kalbarczyk, and R. K. Iyer, "Story of two GPUs: Characterizing the resilience of Hopper H100 and Ampere A100 GPUs," in *Proc. Int'l Conf. for High Performance Computing, Networking, Storage and Analysis (SC)*, 2025, arXiv:2503.11901.

[13] Y. Xiong, Y. Jiang, Z. Yang, L. Qu, G. Zhao, S. Liu, D. Zhong, B. Pinzur, J. Zhang, Y. Wang *et al.*, "SuperBench: Improving cloud AI infrastructure reliability with proactive validation," in *Proc. USENIX Annual Technical Conf. (ATC)*, 2024, pp. 835–850.

[14] NVIDIA Corporation, "Xid errors," NVIDIA GPU Deployment and Management Documentation, 2025, https://docs.nvidia.com/deploy/xid-errors/.

[15] ——, "RMA process," NVIDIA GPU Deployment and Management Documentation, 2025, https://docs.nvidia.com/deploy/rma-process/.

[16] Anthropic, "Claude (large language model)," https://claude.ai, 2026.

Appendix
Artifact Description

Summary. The artifact is a fully self-contained evidence bundle behind every number on the poster: (1) the survey table, a CSV/Markdown record assigning each of the 162 surveyed datasets one of seven evidence verdicts with a quoted justification per entry; (2) the failure-type taxonomy, 931 enumerated raw mentions canonicalized into 88 types in 11 classes, each type carrying its class-assignment rationale and its cause/symptom/outcome-label tag; (3) the cross-cluster matrices (count basis and time basis, 44 system columns) and the GPU error-class matrix (12 systems, 11 classes), with the reporting basis stated per column and a quoted source sentence recorded for every populated cell; (4) the three poster figures with the Python (matplotlib) script that renders them from the matrices; and (5) the three-pass verification report documenting the six value errors caught and corrected.

Data provenance. Every value derives from published, publicly accessible sources: peer-reviewed failure studies, arXiv operational reports, public dataset documentation, and vendor documentation (NVIDIA Xid catalog and RMA/Field Diagnostic policies). No proprietary, private, or personally identifiable data is included; chart-read values are marked as such with a stated uncertainty.

Availability. The bundle is publicly available at https://github.com/TrupeshKumarPatel/hpc-ai-failure-survey. No special hardware or software is needed to inspect the artifact: the survey table, taxonomy, and matrices are plain CSV/Markdown, and the figures regenerate with Python 3 and matplotlib.