THE UNIVERSITY OF ALABAMA®
Lee J. Styslinger Jr. College of Engineering
Computer Science

Do AI Workloads Fail Differently?

A Cross-Cluster Characterization of Failure Types in AI and HPC Systems

Trupesh Patel
Purushotham V. Bangalore
Department of Computer Science, The University of Alabama
trpatel2@crimson.ua.edu · pvbangalore@ua.edu

Why this study

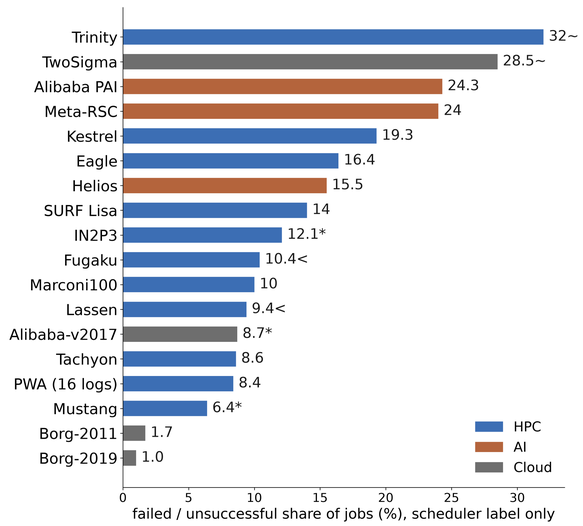
HPC studies say user error causes most failures. LLM training teams blame hardware. The evidence could not be compared, so we surveyed **162 datasets** and sorted each by what it can say:

- **20** explain **why** jobs failed
- **25** say only **that** jobs failed
- **44** give rates but no per-type breakdown
- **36** have no failure data
- **37** are duplicates, archives, or paper-only stats

Method

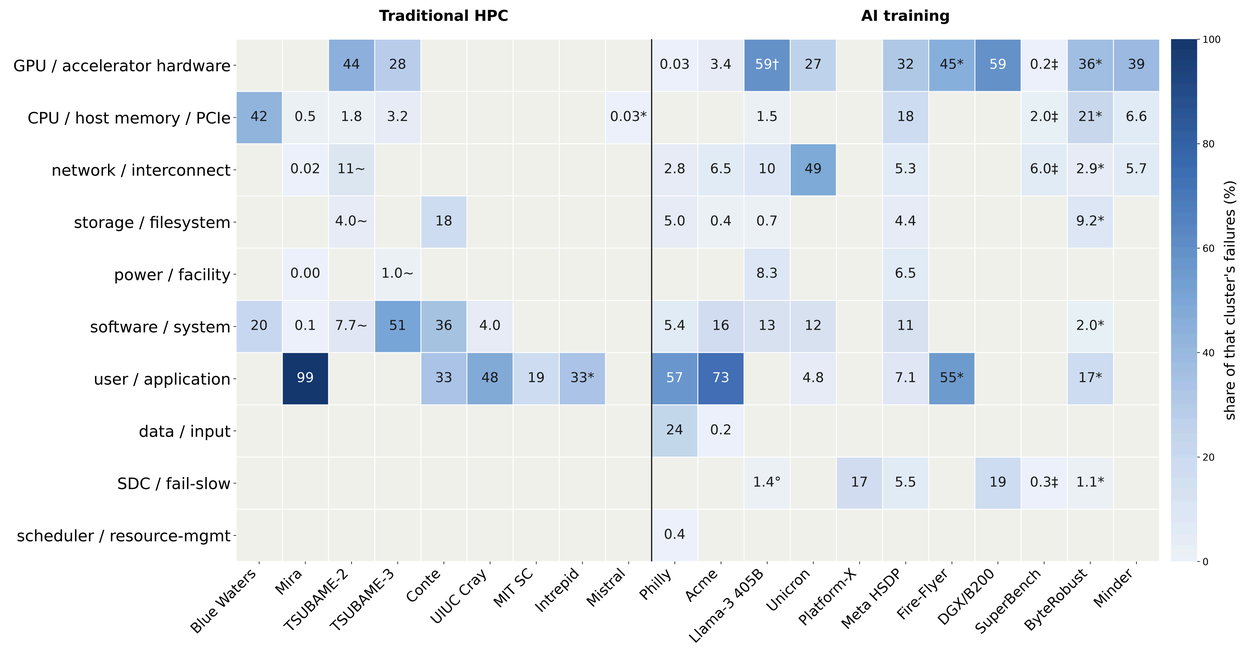
1. **Survey:** we gave each of **162** datasets one evidence verdict (cause, label, rate, or none).
2. **Taxonomy:** we merged **931** mentions from 37 studies into **88** types across **11** classes.
3. **Typing:** every type is tagged as a root cause, a symptom, or an outcome label.
4. **Denominators:** each system keeps its own, so comparisons stay within a column.
5. **Verification:** three passes (quote check, blind re-derivation, adjudication) leave every cell traceable to a source quote.

The half that cannot answer "why"



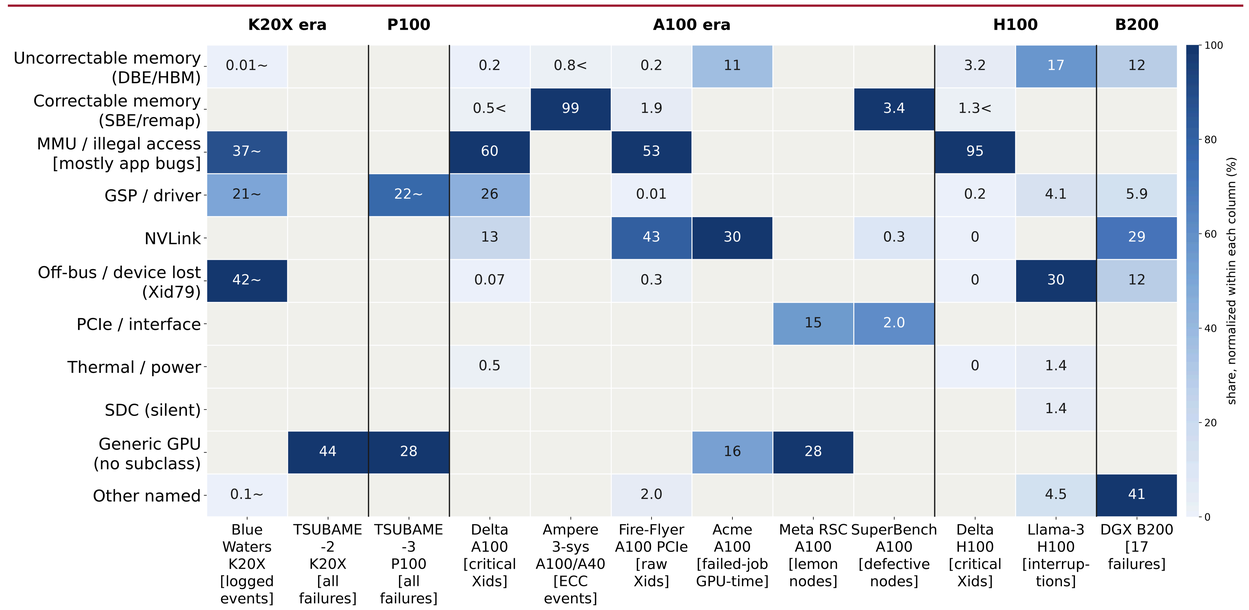
- Eighteen systems publish only scheduler outcomes: the share of jobs marked failed runs from 1% to 32%.
- None can attribute a single failure to a cause; more than half of the survey's usable datasets (25 of 45) fall in this group.
- Borg's low bars are an accounting artifact: most terminations there are dependency-cascade kills, not failures.
- * counted from the raw public trace ~ read from a figure < upper bound.

Where failures come from



- Traditional HPC (left): user and application mistakes dominate failure counts. Mira 99%, UIUC Cray 48%, Intrepid 33%.
- AI training (right): GPU hardware dominates interruptions. Llama-3 59%, DGX/B200 59%, ByteRobust 36%.
- TSUBAME-2 (2012, GPU-dense) already looks like an AI column: the pattern follows GPU density, not workload type.
- Gray = not quantified by that study. Each column keeps its own denominator, so compare within a column only.

GPU error classes, K20X to B200



- A100 era: interconnect (NVLink up to 43%) and GSP/driver errors (26%) dominate the profile.
- H100: memory takes over. Uncorrectable errors reach 17% of Llama-3 interruptions; at Delta, H100 MTBE is 3.2× worse than A100.
- B200 puts NVLink back on top (29% of failures): the reliability profile is generation-dependent, not a stable property of "the GPU".
- The illegal-access row is mostly application bugs, not hardware. Color is normalized within each column.

What the data shows

1. Failure rates are similar (ML 17% vs generic 14%); exposure is not: at 1,024 GPUs, one failure every 8 hours.
2. User errors dominate counts, but infrastructure dominates lost compute (82% at Acme); timeouts take a third to 44%.
3. Interconnect is the top attributable GPU hardware class across every basis measured.
4. Uncorrectable memory always kills the job; at Delta, H100 is 3.2× worse than A100, and repair fails 41% of the time.
5. Illegal-access errors (60–95% of critical Xids at Delta) are mostly application bugs wearing a hardware label.
6. Operations matter: system causes 53% of failures at one cluster and 4% at a comparable one.

High-priority areas to explore

1. Interconnect recovery: documented remediation stops at operator reset, then vendor; no software-layer recovery is documented. **NVIDIA Xid guide**
2. Memory-error containment: checkpoints must be sized to H100-era rates, and vendor thresholds govern replacement. **NVIDIA RMA policy**
3. Correlated-failure screening: continuous fleet telemetry is already operator practice at Delta; correlated defects have no documented owner. **Delta SC'25**
4. Application-fault containment: the vendor routes the illegal-access classes (60–95% of Delta's critical Xids) to the user's debugger. **NVIDIA Xid guide**
5. Correctable-error telemetry as early warning: single events are "no action" by design; prediction lives at fleet level. **Xid 63: IGNORE**

Who fixes what today: users own the illegal-access classes (the Xid catalog routes 13/31/43 to the application debugger); operators restore service (drain, reset, reboot); the vendor owns the durable fix **(NVIDIA RMA policy)**. GSP/driver shows the vendor layer working: 26% of critical Xids on A100, 0.16% on H100. **Delta SC'25**

Lesson learned: none of the vendor or site documentation we examined assigns recovery to the middleware layer (MPI, NCCL, checkpoint libraries) for any failure class. That unowned layer is where fault-tolerance research operates.

Artifact

We release the full survey table of all **162** datasets, the taxonomy with rationale, and every matrix with per-cell source quotes. What the community still lacks is a public dataset that couples job failures to verified causes.

Scan for all 162 datasets
github.com/TrupeshKumarPatel/hpc-ai-failure-survey

Full paper: "Characterizing Failure Types Across HPC and AI Clusters," FTXS @ SC26, to appear.

Acknowledgment

Text drafting, literature extraction, and figure preparation were assisted by Claude (Anthropic), 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.