Achieving AI Cluster Reliability at Scale:
Lessons in Reducing Workload Interruptions

AI CLUSTERS



Ramachandra Reddy Kallam *Production Systems Engineer - Meta*
Bob Wang *Hardware Systems Engineer, NPI AI - Meta*

OCP PLATINUM Member

Why Interruptions Matter at Scale

What is an Interruption?

An interruption is any event that stalls an AI training job and costs training progress.

01

Synchronous & Coupled

Training advances in lockstep, with coordination that never pauses.

Thousands of tightly coupled GPUs

02

Single Point of Failure

A single component failure in any node can instantly interrupt the entire distributed job, halting collective progress.

1 node failure halts entire job

03

Massive Blast Radius

Large training jobs can span 100k+ GPUs, so the blast radius of a single hardware failure is enormous.

Spans 100k+ GPUs per job

What Each Interruption Costs

THE RECOVERY CHAIN

Halt → Checkpoint Rollback → Re-mount → Reschedule → Prechecks → Warm-up

01

Fault Tolerance

Part of the chain is absorbed automatically. Many training jobs now resume without a full restart.

Auto-recovery absorbs part of chain

02

The Hidden Cost

The cost of an interruption is never zero. Even with automatic recovery, performance impact is paid by every rank.

Cost is never zero per rank

03

System Impact

Interruption rate drives throughput, time-to-model, GPU utilization, and cost.

Drives core cluster metrics

04

The Discipline

Each interruption gets counted, attributed, and fixed.

Count, attribute and fix every stall

The Interruption Pipeline

Detection to Resolution

01

Interruption

Job may auto-recover and training continues.

Detection & Recovery

02

Logging

Logs captured: host, symptoms, and candidate cause.

Telemetry Collection

03

Triage & Attribution

Every interruption is triaged and attributed to a root cause.

Root-Cause Analysis & Attribution

04

Resolution

Named failure mode → aggregate → prioritize → debug + fix.

Remediation & Fix

What Attribution Requires

01

Recovery ≠ Attribution

The job continues on a healthy host. The one that failed is still broken, and still has to be diagnosed.

The fault survives the recovery

02

Name the Failure Mode

Without a named failure mode, you cannot prioritize issues or scale debugging.

Unlabeled Errors = Unactionable Noise

03

AI-Assisted Triage & Attribution

Volume outpaces manual triage, so AI agents are now core to how we operate at scale.

AI-assisted triage on every interruption

AI-Assisted Triage & Attribution

Core Methodology & Rules

01
Investigation First

Every interruption gets a real investigation before it gets a label.

02
Source vs. Co-Victim

A host can fail because of something upstream.

One switch losing power makes every host look guilty while reporting nothing.

03
Evidence Package

The output is an evidence package, not a verdict.

It includes a timeline, exclusions, and confidence level.

04
Curated Catalog

Every result resolves against a curated catalog. If a result doesn't resolve, it isn't written.

05
Evidence-Based Routing

Evidence decides where to look next.

Gates keep the investigation out of paths the evidence does not support.

What AI-Assisted Triage & Attribution Gives Us

Key Benefits & Outcomes

01

The List Builds Itself

Attribution runs continuously. The issue list is always current, and no one assembles it by hand.

02

Answers Arrive in Minutes

Interruptions are investigated as they happen.
Causes are known while the job is still running.

03

Repeat Offenders Surface

With causes attributed fleet-wide, defect patterns become visible.

Bad hardware leaves the fleet quickly.

04

Effort Follows Impact

Work goes to what actually dominates.

Attribution covers every interruption, so the ranking reflects the whole population.

Interruption Sources by Platform

Failure Class	Platform-A	Platform-B
Scale-out network (switches, NICs, optics)	**23.6%**	11.7%
Scale-up fabric (switches, links, manager)	**24.2%**	9.2%
Memory (HBM, LPDDR, SRAM)	7.4%	**22.5%**
GPU device & firmware	2.4%	1.4%
Software, scheduling & tooling	12.5%	13.4%
Host / CPU platform & OS	10.8%	11.8%
Workload pre-run checks	8.6%	15.5%
Power & cooling	3.8%	7.5%
Application & illegal memory access	5.0%	5.5%
Other / misc	1.7%	1.4%

Key Insights

- **Generation Shift: network and scale-up fabric are 48% of Platform-A interruptions; Platform-B is memory-limited at 23%.**
- **Memory: HBM uncorrectables run 4.8x higher on Platform-B and system memory ECC 4.2x; SRAM and control-processor signatures are near parity.**
- **Non-Hardware: 26% of Platform-A and 34% of Platform-B interruptions are software, tooling or pre-run checks.**

Fabric & Network Memory & Silicon Host, Power & Cooling Software, Workload & Ops

Turning Top Issues into Fixes

Attribution produces a ranked list. What happens next depends on **who owns the issue**.

01

Supplier-Owned

Joint debug with the silicon supplier. Fixes land in the next firmware, BIOS, or driver release.

02

Meta-Owned

Root cause lands with the owning team. Fixes ship as tooling updates, configuration changes, or hardware replacement.

03

Different Cadences

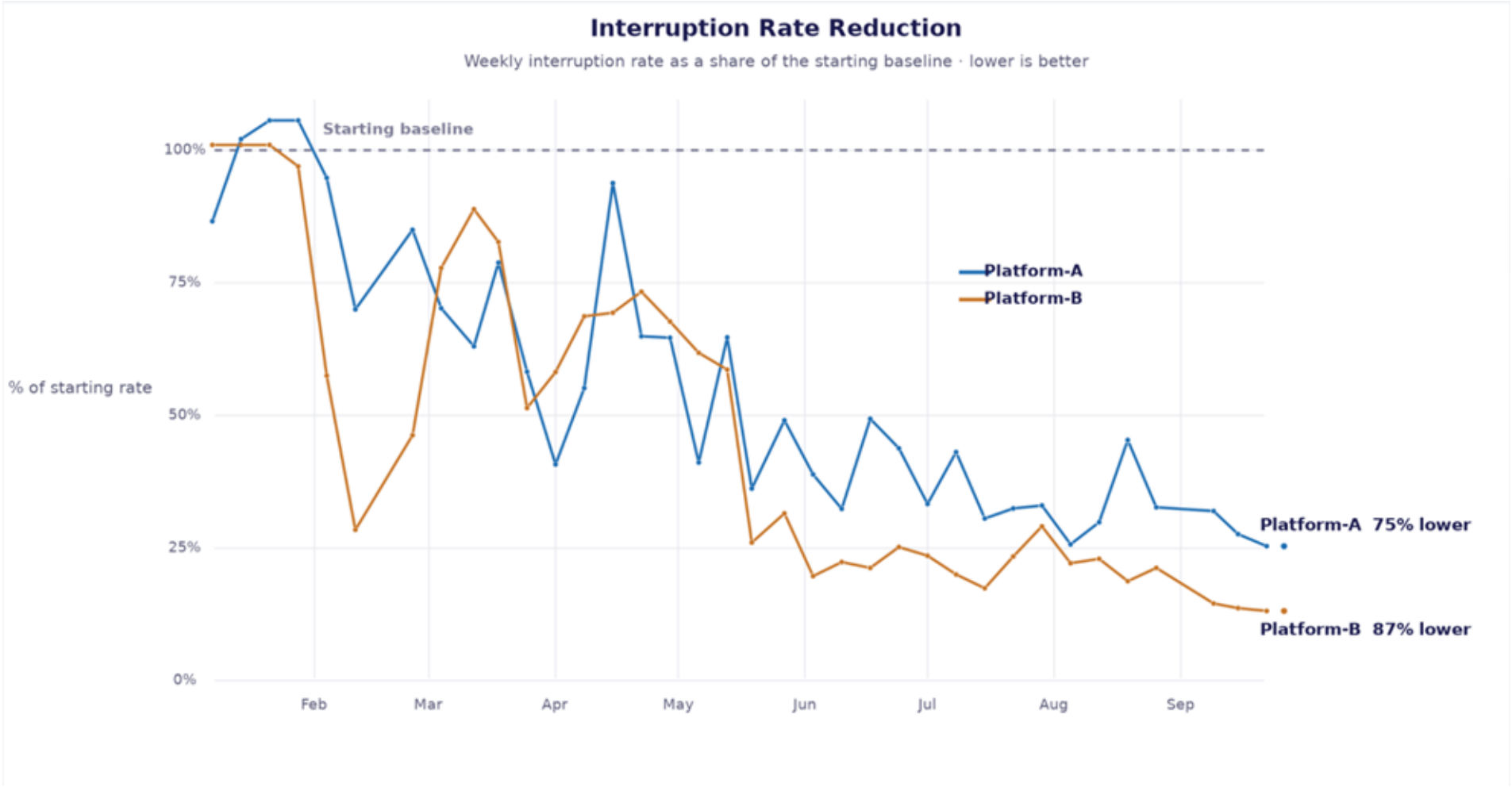
Anything that changes the host or rack rides a release train, whoever owns it. Tooling fixes ship any time.

04

A Common Language

One set of failure classes across both platforms and with the silicon supplier. The next platform inherits them.

Fewer Interruptions, Longer Runs



Covers 2026 only. Platform-A was already partway through an earlier improvement cycle at the start of this window; Platform-B was new.

Call to Action

- Problem to solve
  - Interruptions are a shared problem for anyone running training at this scale. There is no common way to describe or compare what each of us sees.
- Timeline for contribution availability
  - Open to contributing what we have if there is interest.

Open Discussion