GPU SMITH
COMPUTE ENGINEERING

GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

GPU ECC Error Troubleshooting: Xid, Reset, Drain or RMA

> Plan and build private AI compute with GPU Smith. We connect workload requirements with GPU hardware, networking, deployment and operating decisions so your infrastructure fits the work it must perform.

Published by GPU Smith · 2026-10-13 · gpu ecc error troubleshooting · nvidia gpu ecc errors · uncorrectable ecc · nvidia xid · gpu row remapping · gpu reset · dcgm diagnostics · gpu node drain · gpu rma

Original article: https://gpusmith.com/articles/en/gpu-ecc-xid-triage-decision-tree

In this article

1. Executive Summary
2. Introduction and Background
3. ECC, Page Retirement, Row Remapping and Xid
4. Preserve and Correlate the Incident Evidence
5. A Drain, Reset and Escalation Decision Tree
6. Reset Eligibility and Post-Action Validation
7. Recurrence Tracking and the OEM Evidence Package
8. Data Analysis and Evidence
9. Implications and Future Directions
10. Frequently Asked Questions (FAQs)
11. Conclusion

Executive Summary

GPU ECC error troubleshooting, for a graphics processing unit (GPU), should begin with evidence preservation and an explicit recovery action. Error-correcting code (ECC) telemetry, a driver Xid message, a scheduler health condition and a replacement decision describe different parts of an incident. A nonzero lifetime counter can be present when an instance starts without establishing an active fault; operators need the event time and a fresh baseline before drawing conclusions. (repost.aws) Google likewise distinguishes recovery after an uncorrectable error from declaring the host faulty. (docs.cloud.google.com)

The practical decision is **continue under observation, drain and reset, reboot, or retain isolation and escalate**. For the current Ampere-and-newer Xid catalog, a solitary **Xid 48** calls for GPU reset and field-diagnostic investigation; accompanying **Xid 63 or 64** changes the recovery workflow to drain and reset. Application-oriented Xids belong on a different investigation branch. (docs.nvidia.com) Managed services apply their own availability policies: Google Kubernetes Engine (GKE) can prevent further GPU scheduling and recreate an unhealthy node, while Amazon Elastic Container Service (ECS) GPU auto repair can replace an impaired instance. Those actions restore service capacity; they do not independently establish an original equipment manufacturer (OEM) replacement entitlement. (docs.cloud.google.com) (docs.aws.amazon.com)

Before recovery, preserve the physical GPU identity, complete kernel event sequence, ECC windows, retirement and remapping state, workload identity, software versions and relevant interconnect evidence. Azure's reporting interface illustrates why physical-host mapping and GPU identity matter, while national laboratory support guidance asks for job scripts, job identifiers, modules and host names. (learn.microsoft.com) (docs.olcf.ornl.gov) Drain completion also needs verification at the device: Kubernetes deliberately leaves ignored DaemonSet Pods running, and Slurm DRAIN allows existing jobs to finish. (kubernetes.io) (slurm.schedmd.com)

RMA, or return merchandise authorization, requires policy and diagnostic evidence. NVIDIA's published dynamic random-access memory (DRAM) policy requires a row-remapping failure flag validated by its Field Diagnostic tool. (docs.nvidia.com) A provider threshold, such as Azure's preview category for **more than 10 retired double-bit pages in seven days**, applies to that provider workflow rather than every GPU.

(learn.microsoft.com) After any action, check the current health state, actual diagnostic results and application behavior before returning capacity. Oracle Cloud Infrastructure (OCI) explicitly warns that completion of a repair workflow does not confirm correction of the underlying GPU fault. (docs.oracle.com) This report, checked against sources accessed on the publication date, supplies an evidence collector, operational decision tree and incident-bundle schema.

Introduction and Background

An incident starts when a job fails, an ECC counter changes, a driver records an Xid, or a platform removes a device from available capacity. These signals often arrive through separate systems. Kubernetes device plugins, for example, reduce allocatable resources when a device becomes unhealthy while leaving the node's capacity figure unchanged. The already assigned Pods remain a separate recovery concern. (kubernetes.io) (kubernetes.io)

The operator's immediate task is to protect the running workload, preserve evidence and choose the least disruptive supported recovery. The later engineering task is to determine whether the pattern follows application behavior, the installed software environment, the physical GPU or the surrounding platform. Google's GPU troubleshooting guidance uses patterns of Xids when deciding on repairs, which reinforces the value of a timeline over an isolated alert. (docs.cloud.google.com)

This report addresses fleet operators and support engineers after an event has been discovered. It treats hardware qualification, scheduler isolation and application recovery as connected decisions with separate evidence requirements. The proposed workflow is an operational synthesis of the cited documents, rather than a manufacturer's replacement policy.

GPU Smith is an adjacent engineering advisor in this context. Its published scope includes telemetry, failure-mode analysis and operating procedures for datacenter operations; this report adopts that procedural perspective without presenting the consultancy as a chip manufacturer or repair authority. (gpusmith.com)

A useful incident record should answer the following questions before anyone clears state:

- **Identity:** Which physical device and host were involved? Azure explicitly needs physical-host mapping for its reporting workflow. (learn.microsoft.com)
- **Workload:** Which job, script and software modules were active? The National Energy Research Scientific Computing Center (NERSC) requests those details for troubleshooting. (docs.nersc.gov)
- **Sequence:** What preceded the device event, including Peripheral Component Interconnect Express (PCIe) transport errors? Linux documents those separately from device-specific errors. (docs.kernel.org)
- **Recovery:** Which action completed, and what do its diagnostic results show? DCGM requires inspection of warnings and per-entity results. (docs.nvidia.com)
- **Qualification:** Which support organization owns the next diagnostic decision? Dell identifies the driver bug report as material for support escalation. (www.dell.com)

ECC, Page Retirement, Row Remapping and Xid

Volatile ECC and **aggregate ECC** are different observation windows. NVIDIA defines volatile counts relative to the last driver load and aggregate counts as lifetime counters. On Linux, driver unloading can make the volatile window shorter than a host uptime interval. Clearing aggregate counts is unsupported on Ampere and later. Preserve both values together with the driver-load context; a smaller volatile value after recovery cannot demonstrate that lifetime history disappeared. (docs.nvidia.com) Amazon Web Services (AWS) explicitly cautions against interpreting a nonzero counter at instance launch as an active problem. (repost.aws)

Page retirement removes affected memory pages from use. **Row remapping** replaces affected rows in hardware on supported newer GPUs. NVIDIA describes row remapping as the replacement for the earlier retirement scheme starting with Ampere, and says a reset is required for remapping to take effect. (docs.nvidia.com) Keep the GPU generation in the record so the repair mechanism is interpreted correctly. For legacy page retirement, pending pages await blacklisting at reboot and can remain allocatable beforehand. (docs.nvidia.com)

Table 1, checked as of October 13, 2026, distinguishes the incident signals from the conclusions they can support. Provider and scheduler fields are included because they affect the incident's operational handling.

Signal	What it establishes	What the operator still needs
Volatile and aggregate ECC	A recent driver-load window and a lifetime-history window, respectively. (repost.aws)	Capture time, previous baseline and action history.
Pending row remapping	A repair is scheduled for a later reset. NVIDIA distinguishes this from the reset-required recovery action. (docs.nvidia.com)	Current recovery action, affected work and a supported maintenance window.
Recorded remapped-row total	Entries recorded in InfoROM, the GPU's persistent information storage; this is not a direct count of rows already remapped in hardware. (docs.nvidia.com)	Pending state and post-reset comparison.
Kubernetes unhealthy-device report	Allocatable resource reduction; capacity can remain unchanged. (kubernetes.io)	Existing Pod status and host-level recovery.
Managed-service repair condition	A provider has selected a repair workflow. Amazon Elastic Kubernetes Service (EKS) and ECS document different actions for GPU errors. (docs.aws.amazon.com) (docs.aws.amazon.com)	Provider-specific eligibility, completed action and current health.

The table explains why no individual dashboard cell is an RMA verdict. A queued repair, historical error count and blocked scheduler resource each have a different meaning. Retain their raw values rather than flattening them into a single “bad GPU” label.

Xid is the NVIDIA driver's error identifier. Classify the complete sequence using the catalog applicable to the GPU generation. The current catalog covers Ampere and newer; older GPUs are directed to an archive. (docs.nvidia.com) The relevant actions are discussed in the decision table below. A workload investigation also needs to exclude allocation mistakes: NERSC documents that a Perlmutter job step must explicitly request GPUs, even when its parent allocation contains a GPU node. (docs.nersc.gov)

Finally, ECC telemetry is different from software memory-access checking. Compute Sanitizer can identify out-of-bounds accesses, misalignment and other execution problems. Those results inform the application branch; they should remain separately labeled from memory-repair telemetry. (docs.nersc.gov)

Preserve and Correlate the Incident Evidence

Capture evidence **before reset, reboot, driver reload or counter clearing**. Dell documents the broad device view supplied by `nvidia-smi -q`, retired-page queries and bug-report collection. (www.dell.com) (www.dell.com) The following collector is a proposed read-only Python pattern. It records each command's output, error stream and exit status, so an inaccessible device or permission failure remains visible in the bundle.

```
#!/usr/bin/env python3
import os
import subprocess
from pathlib import Path

incident_dir = Path(os.environ["INCIDENT_DIR"])
if not incident_dir.is_absolute():
    raise SystemExit("INCIDENT_DIR must be absolute")
try:
    command_timeout = int(os.environ.get("COMMAND_TIMEOUT_SECONDS", "60"))
except ValueError:
    raise SystemExit("COMMAND_TIMEOUT_SECONDS must be a positive integer")
if command_timeout <= 0:
    raise SystemExit("COMMAND_TIMEOUT_SECONDS must be a positive integer")
os.umask(0o077)
incident_dir.mkdir(parents=True, exist_ok=True)

def capture(name, command):
    out_path = incident_dir / (name + ".stdout")
    err_path = incident_dir / (name + ".stderr")
    with out_path.open("w") as out, err_path.open("w") as err:
        try:
            rc = subprocess.run(
                command, stdout=out, stderr=err, check=False,
                timeout=command_timeout
            ).returncode
        except subprocess.TimeoutExpired:
            err.write("\nTimed out after " + str(command_timeout) + " seconds\n")
            rc = "TIMEOUT"
        except OSError as error:
            err.write(str(error) + "\n")
            rc = 127
    with (incident_dir / "exit-status.txt").open("a") as status:
        status.write(name + " " + str(rc) + "\n")

capture("utc_before", ["date", "-u", "+%FT%TZ"])
capture("kernel", ["dmesg"])
```

```
capture("gpu_all", ["nvidia-smi", "-q"])
capture("utc_after", ["date", "-u", "+%FT%TZ"])
```

Use a fresh directory for each collection. Set `COMMAND_TIMEOUT_SECONDS` to a positive integer; this proposed collector defaults to 60 seconds per command. Output already written to the files survives a timeout, which is recorded as `TIMEOUT` rather than a numeric exit code; collection then proceeds to the remaining commands. Python's `run()` kills and waits for the child before raising `TimeoutExpired`. (Python documentation) Kernel logs are collected before the GPU query. The collector does not filter kernel output to one Xid, because nearby messages can change the interpretation.

The timeout is not a guaranteed wall-clock bound: Python still waits for child termination, and Linux kernel waits can be uninterruptible. (Linux kernel documentation) If a driver call remains stuck in an uninterruptible kernel wait, use a separate host-level collection session for logs and command-state evidence. This is an operational precaution, not a demonstrated GPU failure in this report. The revised collector was checked with simulated command success, failure, missing executables and timeout; no physical GPU collection or reset was performed. Run it in the actual service environment and record any permission restrictions.

Associate the resulting files with these additional records:

- **Physical GPU:** Universally unique identifier (UUID), serial number, model and PCIe bus-device-function (BDF) address from the device inventory. Azure accepts GPU manufacturer, serial, model and PCIe location in its reporting schema. (learn.microsoft.com)
- **Host:** Scheduler node name and cloud instance or physical-host identity. Azure's required resource identifier does not replace its physical-host mapping. (learn.microsoft.com)
- **Software:** Driver, GPU firmware, operating-system and diagnostic versions. Dell documents a video basic input/output system (VBIOS) query for device firmware identification. (www.dell.com)
- **Workload:** Job or Pod identity, script, command, modules, container reference and application error output. (docs.nersc.gov) (docs.olcf.ornl.gov)
- **Accounting:** Job and step states, exit codes and allocated resources. NERSC documents job-ID filtering with `sacct -j`. (docs.nersc.gov)
- **Repair state:** Unmodified ECC, retired-page and row-remapping output, including unavailable fields. Dell's retired-page query includes identity, address and cause. (www.dell.com)
- **Transport context:** Kernel PCIe Advanced Error Reporting (AER) messages and available counters. Linux exposes AER counters in sysfs as additional evidence. (docs.kernel.org)

Correlate **UUID, BDF, job and time** in the same record. Preserve the original timestamp format and identify any conversion to Coordinated Universal Time (UTC). The proposed schema uses both a collection timestamp and an event interval because collection can happen after the affected job exits.

PCIe evidence needs its own interpretation. Linux describes correctable AER events as errors corrected by hardware, while fatal errors make the link unreliable. These transport categories should not be relabeled as GPU DRAM ECC categories. (docs.kernel.org) (docs.kernel.org) Keep the surrounding host messages so support can examine that distinction.

The same principle applies to missing output. Preserve a failed command and its error message; do not convert an unsupported field or an inaccessible device into a zero. A support bundle is useful when it explains what could be observed and what could not. AWS requests the instance identity, detailed device output and driver bug report when Xid problems persist. (repost.aws)

A Drain, Reset and Escalation Decision Tree

The decision tree below is a proposed operator workflow. **Drain** means preventing new placement and completing or safely terminating existing work under the applicable scheduler policy. Slurm's DRAIN permits current jobs to complete, whereas DOWN stops running and suspended jobs. (slurm.schedmd.com) (slurm.schedmd.com) Kubernetes eviction respects PodDisruptionBudgets (PDBs), which express permitted voluntary disruption. (kubernetes.io)

Table 2 maps documented signal classes to incident branches. Its NVIDIA catalog entries are checked as of October 13, 2026, against the current Ampere-and-newer catalog; the older-generation archive and platform-specific reset restrictions still need separate checks. (docs.nvidia.com)

Observed context	Documented interpretation or action	Operator branch
Xid 13 or 31; Xid 43	The current catalog specifies application restart for the first pair and IGNORE for the latter, whose application fault can leave the GPU healthy. (docs.nvidia.com)	Preserve evidence, investigate the application and check for additional device-health signals.
Xid 48 alone, or with 63/64	Alone: reset and field-diagnostic investigation. With the companion events: drain and reset. Companion events have their own catalog actions. (docs.nvidia.com)	Follow the complete sequence and recovery action; do not classify from the last log line alone.
Xid 94 or 95	Restart the affected application after the contained event; the uncontained event requires GPU reset before applications restart. (docs.nvidia.com)	Determine the affected physical device, instances and platform-specific reset scope.
Xid 79 with PCIe context	Driver access through PCI Express is lost; the current catalog specifies bare-metal restart and support contact. (docs.nvidia.com)	Investigate platform and transport evidence; do not substitute an ECC-counter threshold.
Managed GPU unhealthy condition	GKE can remove GPU scheduling eligibility; EKS documents reboot or replacement according to node-management mode. (docs.cloud.google.com) (docs.aws.amazon.com)	Follow the provider workflow and verify service state afterward.
Persistent repair failure or failed validation	DCGM health history and active test results supply different evidence. (docs.nvidia.com)	Keep the affected capacity isolated and prepare the vendor/OEM evidence package.

For this proposed workflow, a corrected-only event with no required recovery action or failed-repair indication enters observation, subject to the platform's policy. Preserve and trend the evidence. Google's use of event patterns supports keeping the timeline rather than turning each occurrence into a replacement verdict. (docs.cloud.google.com)

The table is intentionally sequence-aware. A recoverable application event can coexist with a separate hardware-health condition. An ECC event can also cause a provider to replace capacity before the physical component has been qualified for replacement. Azure's preview reporting categories and ECS's repair actions illustrate those provider decisions. (learn.microsoft.com) (docs.aws.amazon.com)

One-Page Operator Flow

Read the **GPU Recovery Action** field separately from the underlying fault. NVIDIA documents five values with the following responses. (NVIDIA nvidia-smi documentation)

GPU Recovery Action	Response
`None`	No recovery action is needed.
`Reset`	Terminate GPU processes and perform an eligible GPU reset.
`Reboot`	Reboot the operating system before restarting applications.
`Drain P2P`	Quiesce peer-to-peer traffic and requery the action; follow the separate branch below.
`Drain and Reset`	Stop new scheduling; let unaffected work finish or checkpoint, then reset.

```
ECC / Xid / GPU-unhealthy event
  |
  +--> Preserve raw device state, logs, identities and workload context
  |
  +--> Identify generation, complete Xid sequence and recovery action
         Xid 154 reports the recovery action required for companion Xids
          |
          +--> Corrected-only event, no required recovery or failed repair
          |      Observe under the applicable local/provider policy
          |      Preserve the baseline and track recurrence
          |
          +--> Application branch, no additional device-health signal
          |      Investigate / restart affected application
          |      Validate workload and observe device
          |
          +--> Drain P2P recovery action
          |      Stop new scheduling and quiesce all peer-to-peer traffic
          |      Terminate GPU processes conducting peer-to-peer traffic
          |      Disable UVM persistence mode
          |      Requery NVML_FI_DEV_GET_GPU_RECOVERY_ACTION
          |      Follow the returned action; if still DRAIN_P2P, reset
          |      Apply the reset-eligibility gates below before any reset
          |
          +--> Drain/reset or reset/reboot recovery required
          |      Stop new placement at the necessary scope
          |      Finish/checkpoint or terminate affected work as appropriate
          |      Verify device users and platform reset eligibility
          |      Perform supported reset or reboot
          |      Collect new state and validate
```

```
           |
           +--> Failed remapping, failed validation, or unresolved recurrence
                  Retain isolation and assemble evidence
                  Vendor/OEM policy and diagnostic qualification required

Return capacity only after the supported recovery action,
current device health and workload validation have been assessed.
```

For `Drain P2P`, NVIDIA requires quiescing peer traffic, disabling UVM persistence and querying the recovery action again. Follow the returned action; if it remains `DRAIN_P2P`, NVIDIA specifies GPU reset. (NVIDIA recovery flags) (NVIDIA nvidia-smi documentation)

Apply the flow with these operational gates:

- **Placement:** Inspect both scheduling state and existing workloads. Device-plugin health changes do not migrate already assigned Pods. (kubernetes.io)
- **Slurm state:** Inspect the node before and after DRAIN; allocated state determines whether it is DRAINING or DRAINED. (slurm.schedmd.com)
- **Kubernetes drain:** Use the eviction workflow and inspect disruption-budget blockers. (kubernetes.io)
- **Residual clients:** Check skipped DaemonSets and other device users before reset. (kubernetes.io)
- **Checkpoint:** Establish whether the application has a valid native restart path. NERSC's documented transparent checkpoint facility excludes GPU applications. (docs.nersc.gov)
- **Retry:** Distinguish rerunning a job from restoring saved state; NERSC describes requeue as cancel and rerun. (docs.nersc.gov)
- **Escalation:** Keep the workload script, modules and exact error output attached to the support case. (docs.nersc.gov)

For the application branch, a smaller reproducer can help, but preserve the original launch configuration. NERSC warns that reducing Message Passing Interface (MPI) task count can change the algorithm and prevent reproduction. (docs.nersc.gov) Its Compute Sanitizer guidance also documents source-line information without changing optimization level. (docs.nersc.gov) Repeatability is evidence to investigate, not proof of a particular hardware replacement outcome.

Reset Eligibility and Post-Action Validation

A drained scheduler node is not necessarily an idle physical GPU. Kubernetes leaves ignored DaemonSet Pods running; Slurm holding a running job does not suspend or cancel it. (kubernetes.io) (slurm.schedmd.com) NVIDIA requires root privileges and no applications using the affected devices, including monitoring clients. Omitting the device selector expands reset to all GPUs. (docs.nvidia.com)

Reset blast radius depends on hardware topology and platform support. NVIDIA's reset matrix distinguishes direct NVLink from NVSwitch configurations, architecture-specific Fabric Manager dependencies and virtual-machine permissions. An individual reset may be unavailable even when the event names an individual GPU. With Multi-Instance GPU (MIG) partitions, check other instances on that physical device. The A100-specific

triage guidance calls for draining other instances before reset with MIG enabled, and immediate node reboot for an uncontained event with MIG disabled. Apply that platform guidance when selecting the action. (docs.nvidia.com) Establish the supported reset set before selecting a command.

For an eligible, fully idle device, the documented targeted pattern is `sudo nvidia-smi -i GPU_UUID -r`; treat the UUID as an inventory-resolved physical device selector. (docs.nvidia.com) This command belongs after evidence collection and eligibility checks. If reset is unsupported, follow the documented host, virtual-machine or provider reboot procedure. GKE's guest-reboot workflow explicitly includes cordoning and eviction before reboot. (docs.cloud.google.com)

Before action, verify:

- **Isolation scope:** The intended nodes or resources are unavailable for new work, including scheduling exceptions. (kubernetes.io)
- **Existing work:** Current jobs have completed or reached the agreed termination point. Slurm DRAIN permits completion. (slurm.schedmd.com)
- **Device clients:** Scheduler drain has been followed by an inspection of residual Pods and processes. (kubernetes.io)
- **Restart material:** Saved application state represents a correct run state. (docs.nersc.gov)
- **Provider sequencing:** Managed replacements can apply their own drain and task-stop sequence. (docs.aws.amazon.com)
- **Evidence:** Job IDs, host name and relevant scripts remain associated with the incident. (docs.olcf.ornl.gov)

After action, collect a fresh bundle and compare the raw repair flags with the pre-action state. **DCGM**, NVIDIA's Data Center GPU Manager, separates passive health monitoring from active diagnostics; a historical warning and a later diagnostic result can disagree without being interchangeable measures. Its diagnostics guidance also requires inspecting warnings, skipped tests and per-entity errors, beyond the process exit status. (docs.nvidia.com)

Use the installed release's diagnostic command reference. The current reference provides quick through extended suites, explicit entity selection and named tests; older node-triage guidance describes a different suite range and duration estimate. A representative current quick-suite pattern is `dcgmi diag --run 1 --entity-id gpu:0`, after verifying that the entity identifier matches the intended physical GPU. Longer or targeted testing must follow that release's drain and access prerequisites. (docs.nvidia.com) Do not promise a universal completion time.

Then validate the affected workload against its own correctness and performance expectations. OCI explicitly calls for checking current health signals and application health after repair; workflow completion is insufficient by itself. (docs.oracle.com) Record the outcome before returning scheduling eligibility. In Slurm, UNDRAIN clears draining but does not change the base state, and valid node registration remains necessary for new scheduling. (slurm.schedmd.com)

Recurrence Tracking and the OEM Evidence Package

Track recurrence by **physical GPU UUID and serial**, with host placement as a separate dimension. Retain the job script and software context so later investigations can distinguish repeated application behavior from events associated with a device across placements. The Oak Ridge Leadership Computing Facility (OLCF) support requirements provide a useful model for keeping host, job and software identities together. (docs.olcf.ornl.gov) NERSC similarly requests error output and the command's execution time. (docs.nersc.gov)

Table 3 is a proposed incident-bundle schema. It is a portable record design, not a claim that every tool emits identical field names.

Record field	Definition and evidence source	Why it matters
Event and collection times	Original event interval, timezone, collection start and finish; include the command time requested by support. (docs.nersc.gov)	Separates incident occurrence from later sampling.
Physical and service identity	GPU UUID/serial/BDF, host name, instance ID and scheduler node; Azure additionally requires physical-host mapping. (learn.microsoft.com)	Maps telemetry to support and placement records.
Software and firmware	Driver, diagnostic, runtime and container versions; include available VBIOS information. (www.dell.com)	Preserves the environment for reproduction.
Xid and kernel context	Complete event sequence, preceding and following messages, and separately labeled PCIe AER evidence. (docs.kernel.org)	Avoids reducing transport and device events to one category.
ECC and repair state	Raw counter views, pending/failure flags and retired-page details where supported. (www.dell.com)	Records history and outstanding repair work.
Workload and accounting	Script, job/step IDs, launch options, modules, errors and exit states. (docs.nersc.gov) (docs.olcf.ornl.gov)	Connects device evidence with application behavior.
Action and validation	Requested action, scope, actual completion, current health and application outcome. (docs.oracle.com)	Documents whether service recovery was established.
Qualification policy	Applicable OEM/provider policy, diagnostic requested, case reference and disposition; threshold remains policy-dependent. (www.dell.com)	Keeps support qualification separate from a local alert rule.

The schema is most useful when it preserves raw attachments alongside the normalized summary. Unsupported fields should carry their availability status. Record the diagnostic command and release with its results so another engineer can interpret warnings and skipped tests.

For an escalation package:

- **Attach raw device output:** AWS specifically requests detailed `nvidia-smi -q` output for persistent Xid problems. (repost.aws)
- **Attach the driver bug report:** Dell identifies `nvidia-bug-report.sh` as support material. (www.dell.com)
- **Attach launch context:** Include the original job command or script and modules. (docs.nersc.gov)
- **Attach timeline and state:** OCI's support checklist includes identities, conditions, events, workload/PDB state and health evidence. (docs.oracle.com)

- **State the action outcome:** Include current application and device health after recovery. (docs.oracle.com)
- **Request the applicable qualification procedure:** Public diagnostic collection guidance does not itself establish an OEM replacement threshold.

For **when to RMA a GPU with ECC errors**, distinguish escalation readiness from replacement approval. DCGM's online tests operate within the installed environment; NVIDIA requires separate offline diagnostics for complete hardware validation and RMA. (docs.nvidia.com) The retrieved Dell guide establishes collection procedures but does not publish a universal ECC replacement count. Public Lenovo and HPE pages targeted during this research did not yield readable bodies after both permitted retrieval methods, so no replacement criterion is attributed to them.

Data Analysis and Evidence

The quantitative core of this incident workflow is **state, scope and observation window**, rather than an invented fleet-wide error-count cutoff. Vendor and provider numbers belong to different decision systems.

NVIDIA's published DRAM policy identifies failure paths involving an additional uncorrectable remap in a bank already containing **eight** remapped uncorrectable rows, or **512** total uncorrectable remappings. (docs.nvidia.com) A repeated attempt on an already-remapped row can also fail below the bank count, so adding up events alone does not reproduce the policy. (docs.nvidia.com) The policy's qualification condition remains a set failure flag validated by Field Diagnostic. (docs.nvidia.com) Its page does not assign every listed threshold to individual GPU models; confirm applicability with the manufacturer or OEM instead of turning those values into a universal local rule.

Azure's preview **DBEOverLimit** reporting category specifies **more than 10 retired double-bit ECC pages in seven days**. (learn.microsoft.com) That is a provider impact-reporting criterion. OCI's documented recovery workflow instead requires an eligible reboot signal to remain active for **10 minutes** and applies a **60-minute** repeat-reboot cooldown. (docs.oracle.com) Those are automation timing parameters, not memory replacement limits.

Console-message counts also require care. Linux rate-limits correctable and nonfatal AER messages; fatal and Downstream Port Containment (DPC) errors are not rate-limited. (docs.kernel.org) Available AER counters offer another evidence channel. (docs.kernel.org) A local incident summary should therefore distinguish logged messages, hardware counters and independent incident episodes.

Reading a Bundle (Hypothetical Example)

The invented normalized excerpt below illustrates field interpretation. It is not measured device output.

```
{
  "ecc_volatile_uncorrectable": 1,
  "ecc_aggregate_uncorrectable": 1,
  "row_remap_pending": true,
  "row_remap_failed": false,
  "recovery_action": "Drain and Reset",
```

```
  "job_state": "failed",
  "post_validation": "not_run"
}
```

The next action follows the recorded recovery requirement and affected workload. The absence of a failed-remap flag does not justify marking validation complete. After the supported action, store a second sample and the actual application result. OCI's distinction between completed workflow and corrected health supports this final verification step. (docs.oracle.com)

For recurrence analysis, compare samples only when UUID, field semantics and observation windows match. AWS's volatile-versus-aggregate explanation supplies the baseline distinction. (repost.aws) Proposed local metrics include episodes per device-active hour, time to isolation and time to validated return. These are operational measurements, not manufacturer thresholds; their denominators and sampling gaps should be published with the results.

Implications and Future Directions

A durable runbook should put the evidence collector before the recovery command and make scheduling eligibility an explicit final step. Device-plugin health reporting can change new-placement capacity while leaving existing workloads assigned, so alert closure and application recovery require separate checks. (kubernetes.io) Provider automation also needs to be recorded: ECS replacement and GKE repair are service-management actions with their own workflows. (docs.aws.amazon.com) (docs.cloud.google.com)

Generation and tooling metadata deserve the same retention as error values. NVIDIA explicitly says memory reliability, availability and serviceability (RAS) repair support is not guaranteed for every Blackwell GPU. (docs.nvidia.com) A runbook should therefore resolve the actual model and supported feature set before interpreting missing or changed fields.

Monitoring adapters can introduce naming differences. DCGM Exporter preserves legacy metric names while mapping them to current canonical fields; the exporter metric name and backend field should be recorded together when normalizing an incident bundle. (docs.nvidia.com) This is also why a cached command catalog should carry the installed release, not merely a copied dashboard label.

The application-debugging branch needs platform-specific tools. NERSC documents NVIDIA Compute Sanitizer, while the Lawrence Livermore National Laboratory (LLNL) El Capitan guide describes compiler instrumentation and runtime checking for its GPU AddressSanitizer workflow. Those procedures should remain scoped to their documented platforms. (docs.nersc.gov) (hpc.llnl.gov)

Operational improvements should include:

- **Reproducible collection:** Keep the original error output and job script available. (docs.nersc.gov)
- **Controlled drain:** Verify PDBs and residual workloads rather than assuming a cordon stops all clients. (kubernetes.io) (kubernetes.io)
- **Explicit return:** Record current node state and registration before clearing isolation. (slurm.schedmd.com)

- **Support continuity:** Preserve the host name, modules and job identifiers throughout the case. (docs.olcf.ornl.gov)
- **Written acceptance:** Define the application's return criteria and retain the recovery evidence. GPU Smith's published documentation scope includes operating procedures and acceptance reports, which is the relevant advisory perspective here. (gpusmith.com)

Refresh the catalog mappings, diagnostic selectors and platform reset matrix when drivers, GPU generations or node-management modes change. A procedure remains reviewable when its claims point to dated source evidence and its local decisions remain labeled as local.

Frequently Asked Questions (FAQs)

Does every NVIDIA GPU ECC error require replacement?

No. AWS cautions that a nonzero lifetime counter at instance launch does not establish an active fault, and Google distinguishes uncorrectable-error recovery from a faulty-host report. Preserve the incident sequence and follow the applicable support qualification process. (repost.aws) (docs.cloud.google.com)

Can a GPU reset clear the evidence needed for troubleshooting?

It can change the observation context, which is why the proposed workflow captures evidence first. AWS distinguishes lifetime ECC history from volatile counters since driver reload. Compare the post-action sample with the original baseline rather than treating a restarted window as a clean lifetime history. (repost.aws)

When should a GPU node be drained?

Drain when the selected recovery requires affected work to leave the reset scope, or when the platform's unhealthy-resource workflow blocks new placement. GKE's documented health handling and Slurm's DRAIN semantics provide concrete examples. Verify existing workloads separately. (docs.cloud.google.com) (slurm.schedmd.com)

Is a successful DCGM run sufficient for an RMA decision?

No. NVIDIA says separate offline diagnostics are required for complete hardware validation and RMA. Retain the online test's actual results, warnings and skipped tests alongside the support bundle. (docs.nvidia.com)

Should an affected job simply be requeued?

Requeue can restart execution, but it does not by itself establish restored application state. NERSC describes it as cancel and rerun. Confirm the application's restart behavior and usable checkpoint before choosing that action. (docs.nersc.gov) (docs.nersc.gov)

Conclusion

The supported recovery action, preserved evidence and verified return to service should drive GPU ECC troubleshooting. A historical counter, scheduler health condition and application failure each contribute context. Their interpretation depends on the device, workload, software environment and action history. AWS's distinction between lifetime and volatile counters explains why a fresh window alone is insufficient evidence of a repaired device. (repost.aws)

A practical runbook connects physical identity with the job timeline, retains raw repair state, isolates the necessary scheduling scope and checks residual device users before recovery. Kubernetes's drain behavior illustrates why that last check matters: ignored DaemonSet Pods continue to run. (kubernetes.io) Support evidence should remain attached to the same record, including scripts, modules and job identifiers. (docs.olcf.ornl.gov)

Return capacity after assessing current health and application results. OCI's explicit distinction between a completed repair workflow and correction of the underlying fault captures the central validation requirement. (docs.oracle.com) If the state remains unresolved, keep the affected capacity isolated and escalate with the evidence package.

Replacement qualification should follow the applicable manufacturer, OEM or provider procedure. The operational decision tree provides a repeatable way to reach that handoff; it does not supply a universal count-based warranty rule. A complete incident record makes each decision reviewable: what was observed, what was preserved, what action was supported, what completed, and what validation showed.

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. https://repost.aws/knowledge-center/ec2-linux-troubleshoot-xid-errors
2. https://docs.cloud.google.com/compute/docs/troubleshooting/troubleshooting-gpus
3. https://docs.nvidia.com/deploy/xid-errors/analyzing-xid-catalog.html
4. https://docs.cloud.google.com/kubernetes-engine/docs/troubleshooting/gpus?hl=en
5. https://docs.aws.amazon.com/AmazonECS/latest/developerguide/managed-instances-gpu-auto-repair.html
6. https://learn.microsoft.com/en-us/azure/azure-impact-reporting/guest-health-impact-report
7. https://docs.olcf.ornl.gov/support/index.html
8. https://kubernetes.io/docs/tasks/administer-cluster/safely-drain-node/
9. https://slurm.schedmd.com/scontrol.html
10. https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/latest/rma-policy-thresholds-for-row-remapping.html
11. https://learn.microsoft.com/en-us/azure/azure-impact-reporting/guest-health-impact-categories
12. https://docs.oracle.com/en-us/iaas/Content/ContEng/Tasks/contengautorepairgpuworkernodes.htm
13. https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/device-plugins/
14. https://gpusmith.com
15. https://docs.nersc.gov/jobs/troubleshooting/

16. https://docs.kernel.org/PCI/pcieaer-howto.html
17. https://docs.nvidia.com/datacenter/dcgm/latest/learn/modules/dcgm-diagnostics.html
18. https://www.dell.com/support/kbdoc/en-in/000190243/na
19. https://docs.nvidia.com/deploy/nvidia-smi/index.html
20. https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/latest/row-remapping.html
21. https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/error-recovery-and-response-flags.html
22. https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/latest/user-visible-statistics.html
23. https://docs.aws.amazon.com/eks/latest/userguide/node-health-nma.html
24. https://docs.nersc.gov/jobs/
25. https://docs.nersc.gov/tools/debug/compute-sanitizer/
26. https://docs.python.org/3/library/subprocess.html
27. https://docs.kernel.org/scheduler/completion.html
28. https://docs.nersc.gov/jobs/monitoring/
29. https://slurm.schedmd.com/quickstart_admin.html
30. https://docs.nersc.gov/development/checkpoint-restart/
31. https://docs.nersc.gov/tools/debug/
32. https://docs.nvidia.com/deploy/gpu-debug-guidelines/gpu-node-triage.html
33. https://gpusmith.com/articles/en/nvlink-vs-infinity-fabric-gpu-interconnects
34. https://docs.nvidia.com/datacenter/dcgm/latest/reference/command-line-reference/dcgmi/dcgmi-diag.html
35. https://docs.nvidia.com/datacenter/dcgm/latest/user-guide/feature-overview.html
36. https://docs.nvidia.com/deploy/a100-gpu-mem-error-mgmt/supported-gpus.html
37. https://gpusmith.com/articles/en/dcgm-exporter-gpu-monitoring-metrics-alerts-triage
38. https://docs.nvidia.com/datacenter/dcgm/latest/reference/dcgm-exporter-metrics.html
39. https://hpc.llnl.gov/documentation/user-guides/using-el-capitan-systems/using-el-capitan-systems-debugging-tools

THE PUBLISHER

About GPU Smith

GPU Smith provides independent engineering and research for private AI infrastructure. We help technical buyers and operators reason about GPU workload sizing, hardware selection, procurement, deployment and inference operations, from the compute system to the networking, power and cooling requirements around it.

Start with the workload

Useful infrastructure decisions begin with the models, data, throughput, latency and operating constraints a team actually has. GPU Smith helps connect those requirements with system design and deployment choices. The goal is a privately controlled AI environment whose capacity and operational demands are understood before a purchasing decision.

Hardware and supplier research

Our public reference library covers GPUs, complete systems and networking components. The server-vendor directory and manufacturing research help teams investigate suppliers, compare options and follow sources behind technical claims. We distinguish manufacturer specifications from measured benchmarks and advertised capabilities from tested configurations. Reference pages are research resources, not a live inventory listing or a binding equipment quote.

Deployment and operations

GPU Smith's engineering scope includes infrastructure integration, networking, power, cooling and the practical operation of inference workloads. Our published articles explain the tradeoffs behind deployment, procurement and ongoing operations so teams can ask better questions and document decisions.

Work with GPU Smith

Explore the hardware library, GPU references, networking references and vendor research. Contact GPU Smith with your workloads and deployment constraints to discuss a project and confirm current scope, pricing and availability.

Inclusion of a manufacturer or product in our research does not imply a partnership, certification or endorsement.

Website: https://gpusmith.com

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.