GPU SMITH
COMPUTE ENGINEERING

GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

GPU Cluster Acceptance Testing: DCGM and NCCL Runbook

> 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 · 2026-09-19 · gpu cluster acceptance testing · dcgm diagnostics · nccl tests · gpu burn-in · cluster validation · all-reduce benchmark · multi-node gpu testing · acceptance criteria · ai infrastructure

Original article: https://gpusmith.com/articles/en/gpu-cluster-acceptance-testing-runbook

In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes
4. Define Scope and Freeze the Baseline
5. Preflight and Evidence Hygiene
6. Single-Node Diagnostic Ladder
7. Multi-Node Path and Collective Testing
8. Pass, Fail, Skip, and Quarantine
9. Implementation Considerations and Process Changes
10. Data Analysis and Evidence
11. Implications and Future Directions
12. Frequently Asked Questions (FAQs)
13. Conclusion

Executive Summary

GPU cluster acceptance testing is not one benchmark and not one green status. It is a controlled argument that the delivered configuration matches the purchase baseline, functions correctly, remains stable under stress, covers the intended communication paths, and meets thresholds named in the statement of work (SOW). A **Pass** from NVIDIA Data Center GPU Manager (DCGM) describes only the invoked test (docs.nvidia.com). It does not certify an entire node or cluster. Likewise, automated checks alone do not prove every run rule was satisfied (spec.org).

The defensible sequence is **baseline, preflight, single-node function, single-node stress, multi-node paths, performance qualification, disposition**. Identify GPUs by immutable UUID or PCI bus ID (docs.nvidia.com), freeze firmware and software, synchronize clocks, drain workloads, capture inlet conditions, and preserve commands plus raw output. Then run DCGM's quick checks, `context_create`, memory, PCIe, sustained diagnostic, targeted workloads, pulse, memory bandwidth, NVBandwidth, and optional NCCL Tests plugin as applicable. **DCGM's NCCL plugin remains single-node and does not require MPI** (docs.nvidia.com). Standalone **nccl-tests** is the tool for multi-process and multi-node collective coverage (github.com).

As of **September 18, 2026**, current DCGM documentation places MNNVBandwidth in **DCGM 4.7 and later** (docs.nvidia.com). Newer multi-node workloads have explicit release and device boundaries, so they cannot be generalized to every NVIDIA cluster. Unordered all-pairs coverage is **N(N-1)/2**, the edge count of a complete graph (www.csd.uwo.ca). A 32-GPU campaign therefore has **496 unordered pairs**, while a directional matrix has **992 directions**, arithmetic rather than measured performance.

No researched authority supplies a universal bandwidth floor, permissible error count, burn duration, or retest rule. `mnnvbandwidth` expressly applies **no minimum bandwidth threshold** (docs.nvidia.com). Acceptance must therefore source each threshold from the purchased system's validated configuration, vendor

documentation, or SOW. Results below that floor, any correctness error, any untested required path, or an unexplained Skip should quarantine the smallest affected unit. Do not average away a bad node or direction.

Introduction and Background

A new or repaired AI cluster arrives with several different questions hidden inside the word “works.” Is every purchased component present? Can software initialize every accelerator? Do memory and links move correct data? Does the system remain stable at sustained power and temperature? Are all required node-to-node paths exercised? Finally, does measured performance meet the number the buyer actually contracted for? **GPU cluster acceptance testing** must answer each question separately.

This report is a reproducible evidence runbook for data-center acceptance teams, operators, procurement engineers, and technical buyers. Its governing principle is that test scope and acceptance scope are not interchangeable. Diagnostic specifications should standardize test inputs, outputs, metrics, and formats (www.opencompute.org). A known test name can still be unavailable, disabled, unsupported on the installed SKU, or blocked by configuration. A Skip is therefore not a Pass.

In practical terms, it is a **DCGM GPU cluster diagnostics** guide, **NCCL performance testing** reference, **GPU cluster burn-in test** plan, **GPU cluster validation checklist**, catalog of **DCGM diagnostic test commands**, **NCCL all-reduce benchmark** interpretation, **AI cluster acceptance criteria** template, and **multi-node GPU testing runbook**.

The report also separates **functional diagnostics** from **contract performance**. Test procedures should name expected results and trace them to requirements (swehb.nasa.gov). Standalone nccl-tests checks correctness and performance (github.com), but its output becomes an acceptance decision only after the SOW supplies topology, collective, message sizes, run count, aggregation rule, and threshold.

GPU Smith is an **adjacent independent engineering advisor**, not a chip or cloud provider. Its first-party method says engagements are delivered against written acceptance criteria and an as-built documentation set (gpusmith.com). That perspective belongs in the process design, not as a vendor row in a diagnostic-tool comparison.

Key Changes

Acceptance is now invocation-specific

The current DCGM documentation makes the result boundary unusually explicit: Pass, Fail, or Skip describes the selected invocation. Acceptance automation should retain the result, error category, targeted entities, parameters, elapsed time, and raw log. The Open Compute Project likewise calls for recording failing test names and per-chip test time (www.opencompute.org). A green summary without those fields is not enough to reconstruct coverage.

Multi-node DCGM has narrow product boundaries

DCGM's multi-node catalog is distinct from `dcgmi diag` plugins and includes **mnubergemm, nvloom, and mnnvbandwidth**. Their eligibility depends on release, device, and system configuration. Standalone NVLoom documents **N(N-1)** unidirectional pair measurements and **N(N-1)/2** bidirectional measurements that exploit symmetry (github.com) (github.com). These tools are valuable where supported, not universal substitutes for standalone tests.

Reproducibility is part of the deliverable

NIST distinguishes repeatability under the same conditions from reproducibility under changed conditions, and a valid reproducibility statement identifies what changed (www.nist.gov). The acceptance package should treat raw logs, manifests, topology, parameters, and environmental readings as deliverables. SPEC similarly describes an experimental record as a contemporaneously generated artifact (spec.org). Reconstructing a “clean” report later weakens provenance.

Define Scope and Freeze the Baseline

Acceptance begins before burn-in. Translate the bill of materials, validated design, and SOW into a machine-checkable baseline. Redfish can represent part numbers, serial numbers, producers, vendors, and production dates (redfish.dmtf.org). It can also represent BIOS, baseboard-management-controller firmware, device firmware, and drivers (redfish.dmtf.org). Where Redfish fields are unavailable, record that absence and use the vendor's approved inventory method.

The baseline should include:

- **Physical identity:** rack, chassis, node, GPU UUID, NIC, switch port, serial number, part number, and cable endpoint.
- **Software identity:** operating system, kernel, GPU driver, CUDA Toolkit, DCGM, NCCL, nccl-tests commit, MPI, Unified Communication X (UCX), NIC driver, and fabric manager.
- **Firmware identity:** system BIOS, BMC, GPU, NIC, switch operating system, retimer, NVSwitch, and power shelf where exposed.
- **Topology:** CPU and NUMA placement, PCI Express (PCIe) tree, GPU-to-GPU matrix, NIC locality, NVLink state, and node-to-switch mapping.
- **Environment:** inlet and exhaust temperature, GPU temperature, facility power state, board power, fan state, and collection interval.
- **Contract:** required tests, target entities, thresholds, allowable error counts, burn duration, retries, retest scope, evidence format, owner, and final disposition authority.

Use UUID or PCI bus ID because numeric GPU enumeration can change. Preserve a machine-readable PCI inventory rather than parsing display text; `lspci` provides a machine-readable form (man7.org). For InfiniBand, `ibstat` reports port state and active width (man7.org). For Ethernet, retain identification and diagnostic information from ethtool (www.kernel.org).

Freeze the baseline before testing. NIST defines configuration control as controlling modifications to hardware, firmware, software, and documentation (csrc.nist.gov). Any subsequent change creates a new evidence branch: identify the change, owner, reason, affected tests, and whether prior results remain applicable.

Preflight and Evidence Hygiene

The preflight proves the environment is capable of producing interpretable results. Run it on every node before any load test.

- **Synchronize time.** Capture `timedatectl status` and `chronyc tracking`; Ubuntu documents the former as a synchronization check and the latter for detailed accuracy (ubuntu.com) (ubuntu.com).
- **Drain interference.** Reserve selected GPUs, CPUs, memory, and network paths. Open MPI recommends being the only user of the systems and network when maximum performance is measured (docs.open-mpi.org).
- **Capture topology.** Save `nvidia-smi topo -m`, the PCIe tree, NIC locality, link state, routing, and switch-port map. Distinct internal network pathways should be tested separately (datatracker.ietf.org).
- **Capture versions.** Save `dcgmi --version`, `nvidia-smi`, `ompi_info --parsable`, `ucx_info -c`, and the exact Git commit. `ompi_info` supports machine-parsable output (docs.open-mpi.org); `git rev-parse HEAD` records the current commit object (git-scm.com).
- **Check device visibility.** Compare discovered GPUs, NICs, ports, and links with the frozen baseline. `ucx_info -d` shows devices UCX can use (raw.githubusercontent.com).
- **Capture environment.** Record sensor acquisition time, interval, inlet temperature, GPU temperature, board power, and chassis power. Redfish distinguishes reading acquisition time from report time (redfish.dmtf.org). ASHRAE uses equipment inlet temperature as a common reference point (handbook.ashrae.org).
- **Snapshot counters.** Save before values for ECC, PCIe, NVLink, RDMA, NIC, switch, and system logs. Redfish defines RDMA protocol-error and byte counters (redfish.dmtf.org).

Create a run identifier that joins every artifact. Keep console output, structured output, command line, environment, start and stop timestamps, process exit status, DCGM result, error codes, and monitoring streams. Bound operating-system logs to the test window; `journalctl` can filter entries from a specified date and emit newline-delimited JSON (man7.org). Hash the final bundle. `sha256sum` computes and checks SHA-256 digests (man7.org).

Suggested evidence layout:

```
acceptance/<run-id>/
  manifest/{hardware,firmware,software,topology,environment}.json
  preflight/{clock,links,counters,services}/
  dcgm/<node>/<test>/{command.txt,result.json,stdout.log,stats}/
  nccl/<campaign>/<collective>/<message-range>/<raw-output>/
  multinode/<diagnostic>/<raw-output>/
  disposition/{matrix.csv,exceptions.md,approvals.json}
  SHA256SUMS
```

Single-Node Diagnostic Ladder

Run tests from least invasive to most invasive. Stop on setup failures because later results may be uninterpretable. Templates below require the operator to substitute the pinned targets and parameters.

1. **Deployment checks:** `dcgmi diag -r 1 -j`. Archive library, permission, device-access, conflict, and hardware-state messages. Indirect tools invoked by a test suite should also be identified (www.opencompute.org).
2. **CUDA context:** `dcgmi diag -r context_create -j`. A Pass means basic CUDA initialization and context creation succeeded (docs.nvidia.com). It does not prove compute, memory, or communication.
3. **Memory and PCIe:** `dcgmi diag -r 2 -j`. Level 2 adds memory and PCIe tests. The memory test allocates **75% by default** and verifies patterns read back (docs.nvidia.com). Insufficient free memory is a Skip without a memory-integrity conclusion.
4. **Sustained diagnostic:** run the documented level or named plugin on drained GPUs. Its default duration is **three minutes** (docs.nvidia.com). Treat comparative cross-GPU checks as comparative, not an absolute performance warranty.
5. **Targeted stress and power:** supply SOW-approved parameters and trace expected results to the purchased requirement (swehb.nasa.gov). An unattainable power target caused by an enforced limit is a configuration issue, not proof of defective hardware.
6. **Pulse and long memory:** run `pulse_test` and `memtest` only where supported and contracted. Product-category standards, rather than a universal number, should supply detailed burn-in duration (www.opencompute.org).
7. **Memory bandwidth:** use the SKU configuration or SOW threshold. The plugin compares its best timed result with `minimum_bandwidth`, not every iteration (docs.nvidia.com). It does not test memory integrity.
8. **NVBandwidth:** exercise each documented copy path separately, consistent with the principle that distinct pathways require separate tests (datatracker.ietf.org). Compare measurements only with an approved platform floor.
9. **Optional DCGM NCCL plugin:** verify installed NCCL and test binaries, ownership, permissions, configured path, supported GPUs, and idle state. When DCGM runs as root, the binary must be root-owned and not group- or world-writable (docs.nvidia.com). This plugin covers single-node correctness, not multi-node qualification.

After every stage, snapshot the same counters and sensors collected before it. The delta is more useful than an unbounded lifetime count. If a test changes configuration, restart the evidence branch and rerun every dependent stage.

Multi-Node Path and Collective Testing

Begin below the collective layer. Confirm physical link state and rate, run the fabric vendor's approved point-to-point test, and verify MPI launch independently. NVIDIA recommends low-level checks to confirm ports are up in the expected configuration (docs.nvidia.com). `ib_write_bw` can isolate bandwidth between two nodes, but it still requires an SOW floor.

Build nccl-tests with the exact pinned repository revision, CUDA, NCCL, compiler, and MPI. Multi-node operation requires `MPI=1` and `MPI_HOME` (github.com). Record the rank equation: total ranks equal processes multiplied by threads multiplied by GPUs per thread (github.com).

A template, not a universal prescription:

```
mpirun -np <ranks> -N <ranks-per-node> -hostfile <hosts> \
  -x NCCL_DEBUG=VERSION -x NCCL_DEBUG_FILE='<run-dir>/nccl.%h.%p.log' \
  ./build/all_reduce_perf -b <min> -e <max> -f 2 \
  -g <gpus-per-thread> -t <threads> -w <warmups> -n <iterations> \
  -c <check-iterations> -T <seconds> -J <result.json>
```

Verify every flag against the pinned revision. In the current repository documentation, warmups default to **1**, timed iterations to **20**, and run cycles set to zero repeat indefinitely (github.com) (github.com) (github.com). Infinite mode is safe only with an external duration, monitoring, log-capacity, and stop policy.

Test collectives and message ranges that represent the purchased workload. AllReduce alone is not complete application coverage. One published vendor validation example uses AllReduce, Broadcast, and AllGather (infohub.delltechnologies.com). Test meaningful placements: within node, adjacent nodes, across leaf switches, across spines or racks, and isolated one-GPU-per-node groups. NCCL_TESTS_SPLIT can create eight one-GPU-per-node groups on eight-GPU nodes (github.com).

Use DCGM multi-node diagnostics only when the installed release and exact SKU qualify. MNNVBandwidth normal mode requests all directed pairs at up to **16 GPUs**, then samples N pairs above 16; all-pairs mode explicitly covers every unordered pair. Its `time_to_run` is one batch deadline, not a per-pair allowance (docs.nvidia.com). A too-short budget can therefore leave required pairs untested.

Pass, Fail, Skip, and Quarantine

Use a disposition taxonomy that prevents setup problems from masquerading as healthy hardware:

- **Pass:** the stated invocation completed and met every threshold it actually enforced.
- **Performance pass:** a valid measurement also met the buyer-supplied floor under specified conditions.
- **Fail, correctness:** wrong data, reported workload error, nonzero required error delta, or failed pair.
- **Fail, resilience:** reset, crash, timeout under a valid budget, thermal breach, power breach, or monitored error beyond the SOW allowance.
- **Fail, performance:** correct completion below the contractual floor, evaluated with the named statistic.
- **Skip, unavailable:** unsupported SKU, disabled plugin, missing binary, inadequate permissions, or unmet prerequisite.
- **Invalid:** interference, baseline drift, clock problem, logging loss, wrong target, insufficient duration, or incomplete coverage.
- **Not required:** explicitly excluded by the approved matrix, never silently omitted.

Quarantine the smallest traceable unit that explains the finding: GPU, node, cable, switch port, leaf domain, direction, or software image. Preserve its peers as controls. Retest first with the identical baseline to establish repeatability, then change exactly one variable. NIST notes that repeatability can be expressed through dispersion of repeated results (www.nist.gov).

Never average away a failed node or direction. Dell's management documentation illustrates the conservative group rule by assigning the highest-severity member status to the group (www.dell.com). The acceptance matrix should do the same unless the SOW explicitly authorizes redundancy-based acceptance.

Implementation Considerations and Process Changes

Table 1 converts the test ladder into a contractual acceptance matrix. “Buyer must supply” is intentional where no universal criterion exists.

Layer	Scope and prerequisite	Command or evidence	Acceptance source	Disposition
Inventory	Every purchased component; baseline frozen	GPU UUIDs, serials, BOM, firmware and software manifest	Purchase order and validated configuration	Mismatch: hold handoff
Software/context	Every GPU visible; services ready	`dcgmi diag -r 1 -j`; `context_create`	Tool correctness plus SOW exclusions	Setup issue: correct and rerun
Memory/PCIe	Idle target, adequate free memory	Level 2 JSON, statistics, counter deltas	DCGM result and vendor-configured thresholds	Any correctness failure: quarantine
Stress/power	Drained GPUs, steady environment	Diagnostic, targeted stress/power, pulse, monitoring	Vendor configuration; buyer supplies duration and error allowances	Breach: quarantine and isolate
Local paths	Supported exact SKU	NVBandwidth, topology and link deltas	Buyer supplies path-specific floor	Missing required path: incomplete
Single-node collectives	External binary installed and trusted	DCGM `nccl_tests` output	Correctness plus buyer floor where required	Skip is not Pass
Cross-node collectives	MPI, fabric, placement and versions pinned	Standalone nccl-tests JSON and NCCL logs	Buyer supplies collective, sizes, statistic and floor	Bad node/direction remains failed
Multi-node NVLink	Eligible release and SKU only	`dcgmi mndiag` raw and aggregate logs	DCGM correctness; buyer supplies bandwidth floor	Unsupported: use alternative plan

The table prevents a common category error: tool completion and contractual performance are separate columns. It also makes ownership explicit. Procurement controls the contract source, platform engineering controls the baseline, operations controls isolation, and the vendor owns documented remediation under the SOW.

Table 2 compares coverage, not products. It should be read as a selection guide rather than a claim that one tool supersedes another.

Instrument	Native scope	What a Pass establishes	What it does not establish	Primary artifact
DCGM single-node plugins	One host, selected supported GPUs	Invoked functional or stress checks met enforced criteria	Uninvoked plugins, multi-node paths, universal performance	JSON, logs, plugin statistics
DCGM NCCL Tests plugin	Single node only	External NCCL workload completed and passed plugin checks	MPI or cross-node collectives	DCGM result plus external output
Standalone nccl-tests	Processes, threads, GPUs and MPI nodes selected by operator	Correctness checks and measured collective performance for that topology	Contract acceptance without a supplied floor	Console or JSON plus NCCL logs
DCGM mnnvbandwidth	Eligible multi-node NVLink systems	Requested transfers completed without reported error	General bandwidth qualification	Text, aggregate logs, JSON limitations
Low-level fabric tools	Selected endpoint pair and protocol	That path met the tool's explicit condition	Collective behavior or all paths	Per-port counters and raw output

The practical consequence is layered evidence. A cluster can pass every single-node plugin and still have a cross-node placement problem. It can also complete collectives correctly yet fail a contractual throughput floor. Both outcomes are internally consistent.

Table 3 maps common outcomes to the next evidence and accountable owner.

Observed class	Next evidence	Immediate action	Owner
Setup or unavailable	Plugin load, binary path, ownership, services, SKU support	Mark Skip or Invalid; correct prerequisite	Platform engineering
Correctness error	Raw mismatch, GPU UUIDs, ECC/PCIe/NVLink deltas, peer control	Quarantine endpoints; repeat unchanged	Hardware/vendor interface
Thermal or power issue	Inlet, GPU and chassis time series; limits; fan and PDU state	Hold stress stage; isolate facility versus node	Data-center operations
Asymmetric path	Direction matrix, port counters, route, cable and switch map	Preserve direction; test reverse and peer swap	Network/fabric team
Below-contract performance	Raw samples, placement, topology, interference, exact floor	Rerun controlled; do not replace with fleet mean	Acceptance authority

This matrix keeps remediation evidence-driven. NASA guidance for test procedures emphasizes expected results and traceability to requirements (swehb.nasa.gov). Cleanup should also return the test system to a known state before the next controlled run.

Data Analysis and Evidence

Pair coverage grows quadratically. For **N** endpoints, unordered all-pairs coverage is **N(N-1)/2**; a complete graph has that many edges (www.csd.uwo.ca). Directional coverage is **N(N-1)** because A-to-B and B-to-A are distinct. The arithmetic is:

- **8 endpoints:** 28 unordered pairs, 56 directions.
- **16 endpoints:** 120 unordered pairs, 240 directions.
- **32 endpoints:** 496 unordered pairs, 992 directions.
- **64 endpoints:** 2,016 unordered pairs, 4,032 directions.
- **128 endpoints:** 8,128 unordered pairs, 16,256 directions.

These are coverage counts, not performance results. Sequential all-pairs testing can exceed a fixed wall-clock budget as N rises, so the matrix must disclose whether it ran all pairs, a documented sample, or a topology-stratified subset. A completion status without the tested-pair list cannot prove full path coverage.

Interpret NCCL bandwidth fields correctly. Algorithm bandwidth, **algbw**, is message size divided by operation time (github.com). Bus bandwidth, **busbw**, applies a collective-specific conversion. For AllReduce across N ranks:

```
busbw = algbw × 2(N-1)/N
```

For ReduceScatter and AllGather:

```
busbw = algbw × (N-1)/N
```

The formula depends on the collective (github.com). Therefore, compare the same collective, rank count, message size, placement, data type, and software baseline. Do not compare AllReduce busbw directly with a different collective as if they were identical workloads.

Statistics also belong in the SOW. Open MPI warns that short runs collect fewer samples and are less accurate (docs.open-mpi.org). It recommends warmups for short communication events and says reports should identify whether the statistic is a minimum, average, or maximum (docs.open-mpi.org). SPEC's **2026** run rules offer an example, not a GPU-cluster prescription: three runs use the median, while two use the slower result (www.spec.org). The buyer must choose its own rule before observing results.

Quantitative evidence should include error deltas and environmental context, not bandwidth alone. Redfish exposes correctable and uncorrectable error counts since reset (redfish.dmtf.org). It also defines actual chassis watts (redfish.dmtf.org). State sensor intervals because periodic, change-triggered, and request-triggered reports are not equivalent (www.dmtf.org).

Implications and Future Directions

Acceptance programs should evolve from a single burn-in script into a versioned test specification. New DCGM releases can add capabilities, alter product eligibility, and change result semantics. Pin the release and keep the producing version with JSON because DCGM does not publish a stable versioned schema for `dcgmi` JSON output (docs.nvidia.com).

Three process changes follow:

- **Contract earlier.** Add thresholds, topology, statistics, duration, allowable errors, retry limits, and disposition before purchase. “Run NCCL” is not measurable acceptance language.
- **Automate evidence, not judgment.** Automation should collect, correlate, hash, and populate the matrix. The acceptance authority still evaluates exceptions against the SOW.
- **Design for differential retest.** Stable identifiers and dependency mapping allow a repaired GPU, cable, NIC, or software image to trigger the smallest defensible retest set.
- **Preserve raw and rendered output.** Human-readable summaries aid review, while raw artifacts support reanalysis as parsers and criteria change.
- **Treat environmental context as test input.** An allowable envelope is a functionality boundary, not a reliability statement (handbook.ashrae.org).

For an adjacent advisor such as GPU Smith, the useful role is to help reconcile the validated design, vendor documentation, and SOW into auditable criteria. Its published method places integration, burn-in, and acceptance testing against written criteria (gpusmith.com). That is process framing, not a substitute for NVIDIA or system-vendor product specifications.

Frequently Asked Questions (FAQs)

What is a GPU cluster acceptance test?

It is a controlled set of inventory, correctness, stress, path-coverage, and performance checks mapped to written acceptance criteria. A benchmark number alone is not an acceptance test because it lacks the baseline, required scope, threshold source, and disposition rule.

Which DCGM command should run first?

Begin with the quick deployment suite, typically `dcgmi diag -r 1 -j`, on the exact target set. Resolve software, permissions, visibility, and conflicting-workload issues before memory or stress tests. Preserve the command and full output.

Does a DCGM Pass mean the cluster is accepted?

No. It means that invocation passed what it tested. Passing automated checks alone is not sufficient proof that all run rules were satisfied (spec.org). Parse the diagnostic result, not only the shell status.

Is the DCGM NCCL plugin a multi-node test?

No. The documented plugin is single-node. Use standalone nccl-tests with a pinned MPI-enabled build for multi-node collectives, then apply the buyer's topology and threshold rules.

What is a valid NCCL test pass criterion?

At minimum: correct output, successful completion, no disallowed monitored errors, all required placements exercised, and performance meeting the SOW floor using the specified statistic. There is no universal bandwidth number applicable to every GPU, fabric, topology, collective, message size, and software release.

How long should GPU cluster burn-in run?

Use the duration in the vendor documentation, validated configuration, or acceptance SOW. Do not invent one. OCP material likewise refers detailed burn-in duration to the applicable product-category standard rather than supplying a universal duration (www.opencompute.org).

What belongs in the final evidence package?

Include the approved matrix, hardware and software manifests, topology, environment, commands, raw outputs, structured results, error and sensor deltas, exceptions, retest lineage, approvals, and hashes. NIST recommends preserving original audit content and time ordering (nvlpubs.nist.gov).

Conclusion

A defensible GPU cluster acceptance program is a chain of bounded claims. The frozen manifest establishes what was delivered. Preflight establishes that the environment and evidence system are ready. Single-node diagnostics establish specific aspects of initialization, memory, PCIe, compute, power, and local communication. Multi-node tests establish only the paths, placements, collectives, and message ranges actually exercised. Contract thresholds then decide whether valid measurements are acceptable.

The operational rule is simple: **Pass means only what the invocation tested; Skip means required coverage is still open; completion is not a bandwidth qualification**. Current DCGM multi-node features are useful but version- and SKU-bounded. Standalone nccl-tests supplies flexible collective coverage, but it cannot invent the buyer's threshold. MNNVBandwidth can prove that requested transfers completed without reported errors while still making no minimum-bandwidth determination.

Acceptance should be granted only when every required matrix row has an accountable result, every threshold has an authoritative source, and every exception has an approved disposition. Quarantine failures at the smallest traceable unit, retain directionality, and rerun under a controlled baseline. A signed summary without raw evidence is weak; a hashed, versioned bundle that reconstructs the decision is durable. That standard supports an informed choice to accept, quarantine, repair, retest, or return the system under the agreed SOW.

Source links

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

1. https://docs.nvidia.com/datacenter/dcgm/latest/reference/diagnostics/index.html
2. https://spec.org/osg/policy/
3. https://docs.nvidia.com/deploy/nvidia-smi/
4. https://docs.nvidia.com/datacenter/dcgm/latest/reference/diagnostics/plugins/nccl-tests.html

5. https://github.com/NVIDIA/nccl-tests
6. https://docs.nvidia.com/datacenter/dcgm/latest/reference/multi-node-diagnostics/mnnvbandwidth.html
7. https://www.csd.uwo.ca/%7Eabrandt5/teaching/DiscreteStructures/Chapter7/graphs.html
8. https://gpusmith.com/articles/en/infiniband-vs-spectrum-x-vs-roce-vs-ethernet-ai-clusters
9. https://www.opencompute.org/documents/external-ver-0-3open-compute-specification-server-component-resilience-workstream-pdf
10. https://swehb.nasa.gov/spaces/SWEHBVB/pages/32604437/Test%2B-%2BSoftware%2BTest%2BProcedures
11. https://gpusmith.com/
12. https://github.com/NVIDIA/nvloom/blob/main/README.md
13. https://www.nist.gov/pml/nist-technical-note-1297/nist-tn-1297-appendix-d1-terminology
14. https://redfish.dmtf.org/redfish/schema_index
15. https://gpusmith.com/articles/en/nvlink-5-vs-nvlink-6
16. https://gpusmith.com/articles/en/gpu-data-center-cooling-design
17. https://man7.org/linux/man-pages/man8/lspci.8.html
18. https://man7.org/linux/man-pages/man8/ibstat.8.html
19. https://www.kernel.org/pub/software/network/ethtool/
20. https://csrc.nist.gov/glossary/term/configuration_control
21. https://ubuntu.com/server/docs/how-to/networking/chrony-client/
22. https://docs.open-mpi.org/en/v5.0.1/tuning-apps/benchmarking.html
23. https://datatracker.ietf.org/doc/rfc2544/
24. https://docs.open-mpi.org/en/main/man-openmpi/man1/ompi_info.1.html
25. https://git-scm.com/docs/git-rev-parse
26. https://raw.githubusercontent.com/openucx/ucx/master/docs/source/faq.md
27. https://gpusmith.com/articles/en/gpu-rack-power-requirements-planning-guide
28. https://redfish.dmtf.org/schemas/DSP2053_2026.1.html
29. https://handbook.ashrae.org/Handbooks/A15/IP/A15_CH19/a15_ch19_ip.aspx
30. https://man7.org/linux/man-pages/man1/journalctl.1.html
31. https://man7.org/linux/man-pages/man1/sha256sum.1.html
32. https://docs.nvidia.com/datacenter/dcgm/latest/reference/diagnostics/plugins/context-create.html
33. https://docs.nvidia.com/datacenter/dcgm/latest/reference/diagnostics/plugins/memory.html
34. https://docs.nvidia.com/datacenter/dcgm/latest/reference/diagnostics/plugins/diagnostic.html
35. https://www.opencompute.org/documents/open-rack-specification-v21
36. https://docs.nvidia.com/datacenter/dcgm/latest/reference/diagnostics/plugins/memory-bandwidth.html
37. https://docs.nvidia.com/deeplearning/nccl/user-guide/docs/troubleshooting/networking_troubleshooting.html
38. https://infohub.delltechnologies.com/en-us/l/dell-ai-factory-with-nvdia-and-dell-networking/nccl-validation-6/
39. https://www.dell.com/support/manuals/en-us/dell-openmanage-enterprise/ome_4_1_online_help_and_user_guide/view-the-baseline-compliance-report?guid=guid-d0eaf47f-74f3-42f2-93b0-1476b2d0f920

40. https://www.csd.uwo.ca/~abrandt5/teaching/DiscreteStructures/Chapter7/graphs.html
41. https://github.com/NVIDIA/nccl-tests/blob/master/doc/PERFORMANCE.md?plain=1
42. https://www.spec.org/cpu2026/docs/runrules.html
43. https://www.dmtf.org/sites/default/files/standards/documents/DSP2051_1.0.1.pdf
44. https://docs.nvidia.com/datacenter/dcgm/latest/reference/command-line-reference/dcgmi/index.html
45. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/800-171r3/NIST.SP.800-171r3.html

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.