



NVIDIA DCGM vs AMD SMI: GPU Telemetry Field Mapping

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 · 2026-10-15 · [nvidia dcgm vs amd smi](#) · [dcm metrics](#) · [amd smi](#) · [device metrics exporter](#) · [gpu telemetry](#) · [gpu monitoring](#) · [prometheus](#) · [cross vendor observability](#)

Original article: <https://gpusmith.com/articles/en/nvidia-dcgm-vs-amd-smi-telemetry>



In this article

1. Executive Summary
 2. Introduction and Background
 3. NVIDIA DCGM and DCGM Exporter
 4. AMD SMI and Device Metrics Exporter
 5. Feature Comparison
 6. Performance and Benchmarks
 7. Data Analysis and Evidence
 8. Dashboard Migration Procedure
 9. Implications and Future Directions
 10. Frequently Asked Questions (FAQs)
 11. Conclusion
-

Executive Summary

For graphics processing unit (GPU) fleets, **NVIDIA Data Center GPU Manager (DCGM) Exporter** and **AMD Device Metrics Exporter** expose related signals, but they do not publish a common telemetry contract. This comparison selects NVIDIA DCGM Exporter **4.8.3** paired with DCGM **4.6.0** and AMD Device Metrics Exporter **1.5.2** as its release scope; the selection is not a claim that these are the latest releases. NVIDIA's active collector rows and AMD's enabled field list determine what appears at runtime; hardware, driver, permissions, and partitioning further narrow the output. The safest migration begins with the effective `/metrics` payload and a versioned dictionary, rather than a global rename of dashboards. (docs.nvidia.com) (instinct.docs.amd.com)

The clearest transferable families are device identity, occupied frame buffer or video random access memory (VRAM), temperature, and power, provided units and sensor scope are normalized. They are **approximate** mappings until a live sample confirms scope and cadence. DCGM's default frame buffer used value is described as **mebibytes (MiB)**; AMD's `GPU_USED_VRAM` is documented in **megabytes (MB)**, and the AMD value is absent on a Single Root Input/Output Virtualization (SR-IOV) hypervisor host using GIM. DCGM energy is a **millijoule counter since driver reload**; AMD exports accumulated **microjoules** from an accumulator tied to restart or GPU reset. Convert both to joules before comparing rates, and keep their reset boundaries distinct. (github.com) (instinct.docs.amd.com) (docs.nvidia.com) (rocm.docs.amd.com) (physics.nist.gov)

Activity, interconnect, and health need narrower decisions. `DCGM_FI_DEV_GPU_UTIL` is a device-sampled percentage; AMD's `GPU_GFX_ACTIVITY` is graphics-engine activity and its `GPU_UMC_ACTIVITY` is memory-engine activity. NVIDIA profiling fields are interval averages and are not identical to either AMD field. NVIDIA Peripheral Component Interconnect Express (PCIe) transmit and receive gauges are directional bytes per second; AMD's documented exporter offers an aggregate `PCIE_BANDWIDTH` in Mb/s. NVLink and extended GPU memory interconnect (XGMI) expose topology-specific signals, not one shared fabric counter. `XID` and AMD Field Identifier (AFID) values are vendor event identifiers, not interchangeable incident totals. (github.com) (instinct.docs.amd.com) (docs.nvidia.com) (github.com) (github.com)

The operational rule is to grade each mapping **exact**, **approximate**, **composable**, **vendor-specific**, or **unavailable**; carry source API, unit, kind, aggregation, sampling, reset, support, and privilege in the dictionary; and re-baseline alert thresholds on observed workload distributions. Preserve missing-series alerts separately from zero-value alerts because a disabled, unsupported, or unobserved series can disappear without proving that a GPU is idle or healthy. Prometheus distinguishes counter resets from gauges and marks stopped series stale. No hardware was available for this report, so the tables below are documented specifications and a validation plan, not measured parity. ([prometheus.io](#)) ([prometheus.io](#)) ([prometheus.io](#))

Introduction and Background

A mixed **NVIDIA and AMD** private cluster usually inherits dashboards, recording rules, and alerts built around DCGM names. The practical decision is which rules survive a vendor addition and which need new definitions. A metric's spelling, even when both vendors call it “utilization,” is only a hint. An observability contract must name the measured object, the time window, the sensor or API, its unit, its type, the label identity, and when the number resets. Prometheus explicitly advises one quantity and one unit per metric; OpenTelemetry's developing GPU convention uses a fraction of time for utilization and a separate fraction for memory occupancy. Neither convention automatically makes vendor source fields semantically equal. ([prometheus.io](#)) ([opentelemetry.io](#)) ([opentelemetry.io](#))

GPU Smith's public scope includes private infrastructure integration and operational telemetry, so its role here is an **adjacent engineering perspective**, not a row in the vendor comparison. The report evaluates documented products and provides a method for operators to verify the mapping on their own hardware. It does not claim a GPU measurement or a client deployment. ([gpusmith.com](#)) ([gpusmith.com](#))

The stack boundary matters. [DCGM Exporter selects DCGM fields](#), watches GPU or other entities, labels them, and serves Prometheus output from `/metrics`. Without a remote host engine, it embeds a DCGM host engine. AMD SMI is the successor to ROCm SMI and provides API and command-line access; AMD's Device Metrics Exporter turns supported signals into Prometheus series. A dashboard often queries an exporter, not the underlying vendor API, so it must use the current exporter schema. ([docs.nvidia.com](#)) ([rocm.docs.amd.com](#)) ([instinct.docs.amd.com](#))

This comparison is a **semantic dictionary and migration procedure**. It excludes acceptance burn-in and economic capacity modeling. It treats a documented but absent signal as a support question to test, not as evidence that a device has no activity. NVIDIA warns that configuring a field does not guarantee emission; AMD's virtualization and partition documentation makes the same caution concrete for several fields. ([docs.nvidia.com](#)) ([rocm.docs.amd.com](#))

NVIDIA DCGM and DCGM Exporter

Capabilities

The current NVIDIA exporter documentation distinguishes **active default collector rows** from commented optional examples. Its shipped default covers clocks, temperatures, power, total energy, frame buffer memory, [GPU utilization](#), PCIe and NVLink throughput, and the last XID value. A configured Prometheus type is used as

written, without semantic validation against the DCGM field definition. Operators should therefore treat the collector CSV, the DCGM field identifiers, and the live output as three separate evidence layers.

(docs.nvidia.com) (github.com) (github.com)

A default DCGM watch cadence of **30,000 milliseconds** is configurable by field group. An HTTP scrape reads the latest cached host-engine sample, so a **15-second** Prometheus scrape does not create a **15-second** new device reading by itself. Product-dependent sampling also affects DCGM_FI_DEV_GPU_UTIL. Profiling fields, including DCGM_FI_PROF_SM_ACTIVE, report interval averages, and paused profiling watches return blank values. (docs.nvidia.com) (github.com)

Adoption and collection scope

NVIDIA's release notes pair Exporter **4.8.3** with DCGM **4.6.0**; the documentation cautions against mismatched versions. Kubernetes implementations show why exporter labels matter: AWS combines DCGM data with Amazon Elastic Kubernetes Service (EKS) metadata for pod-level views, and Google Kubernetes Engine (GKE) collection adds GPU and pod identity labels. These are examples of attribution, not evidence that an activity number changes meaning at the pod boundary. (docs.nvidia.com) (aws.amazon.com)

(docs.cloud.google.com)

Strengths and limitations

The DCGM field catalog is broad, but effective coverage depends on GPU generation, entity, driver, and privileges. Profiling is documented for supported data center products beginning with Volta, while the container workflow requires SYS_ADMIN for profiling fields. Multi-Instance GPU (MIG) samples add GPU_I_PROFILE and GPU_I_ID; a rule that aggregates without those labels may merge unlike entities. The default last-XID gauge records an identifier, whereas optional exporter XID counts are sampled records within a window and exporter totals reset on restart or collector rebuild. Neither can be renamed into a generic hardware fault count without changing the alert definition. (docs.nvidia.com) (github.com)

AMD SMI and Device Metrics Exporter

Capabilities

AMD SMI provides its own API and command-line surface; the **Device Metrics Exporter 1.5.2** publishes a Prometheus schema. Its default empty GPUConfig.Fields list exports all eligible fields, and configuration can prefix metric names. AMD release notes say the server-side collection filter follows enabled fields and labels, while error-correcting code (ECC) and PCIe status remain collected for health validation. An inventory of enabled configuration is therefore part of a schema migration. (rocm.docs.amd.com) (instinct.docs.amd.com)

AMD names separate graphics and memory engine activity percentages, GPU_GFX_ACTIVITY and GPU_UMC_ACTIVITY. GPU_USED_VRAM measures occupied capacity in MB. GPU_POWER_USAGE is watts, but in partitioned modes only the primary partition reports the device power sensor and other partitions can report zero. The accumulated GPU_ENERGY_CONSUMED value is in microjoules. A zero from a secondary partition must be interpreted as a visibility rule, not zero socket power. (instinct.docs.amd.com) (rocm.docs.amd.com)

Adoption and collection scope

The exporter release notes list supported accelerator and ROCm combinations, including **MI2xx and MI3xx with Radeon Open Compute (ROCm) 6.2 or later** and **MI350P with ROCm 7.13 or later**. Those are documented minimums, not a guarantee that each field appears on every product. AMD SMI distinguishes bare metal from virtual machine guest support; its graphics activity API is documented as unavailable in guests. A GIM/SR-IOV hypervisor host does not expose per-virtual-function GPU_USED_VRAM in the exporter list. (instinct.docs.amd.com) (rocm.docs.amd.com) (rocm.docs.amd.com)

Strengths and limitations

AMD's partition model is central to attribution. In Compute Partitioning mode (CPX) on an MI300X, one socket can yield **eight processor handles**, and device-wide GPU metrics live only on the primary partition. Exporter identity labels include partition and deployment information; guest mode can leave `card_model` and `serial_number` empty. An operator must choose physical-socket and logical-partition views explicitly. For SR-IOV hypervisor deployment, the documented exporter container uses a GIM management device and has a separate installation path. Optional profiler metrics are disabled by default and require performance-counter access. (rocm.docs.amd.com) (rocm.docs.amd.com) (instinct.docs.amd.com)

Feature Comparison

Mapping grades and dictionary rules

Exact means the observable, scope, unit after a mechanical conversion, time window, and reset contract match for the pinned versions. **Approximate** means the signals can support a similar operational question after a new baseline but retain a semantic difference. **Composable** means a documented formula plus labels can form a useful derived view. **Vendor-specific** means retaining the native signal is safer. **Unavailable** means the documented exporter has no defensible counterpart for the requested view. These are decision grades, not vendor performance ratings. A conversion can make units comparable without making sampling or sensor identity equal. (prometheus.io) (grafana.com)

Table 1 is a CSV-friendly field dictionary. Each row carries the native key, its closest counterpart, source API or exporter family, unit and kind, grade, and the main reset, support, or scope constraint. **Scope: the selected releases; documented specifications, not hardware measurements.** “Gauge” describes the exported series type; where a source does not specify a reset rule, the row says to verify it. Use release-specific exporter references for the dictionary; treat remaining unversioned API references as general context that needs separate version validation.

DCGM key	AMD configuration key	Unit, kind and source API	Sampling and aggregation	Reset or rollover	Support and privilege	Grade	Source URLs
DCGM_FI_DEV_GPU_UTIL	GPU_GFX_ACTIVITY	Percent gauge; DCGM device / AMD SMI graphics engine	NVIDIA product-dependent sample; AMD engine activity; no portable aggregation	Gauge; no counter reset	Partition activity can be primary-only; verify access	Approximate	(github.com) (instinct.docs.amd.com)
DCGM_FI_PROF_SMI_ACTIVE	GPU_GFX_ACTIVITY	Ratio vs percent gauges; DCGM profiler / AMD SMI	NVIDIA interval average vs AMD engine activity	Gauge; no counter reset	NVIDIA supported profiling hardware and SYS_ADMIN; AMD profiler separate	Approximate	(github.com) (instinct.docs.amd.com)
DCGM_FI_DEV_FB_USED	GPU_USED_VRAM	MiB vs MB gauges; DCGM frame buffer / AMD SMI VRAM	Capacity snapshot; sum only across distinct physical devices	Gauge; no counter reset	AMD GIM host lacks per-VF use; verify memory scope	Comparable	(github.com) (instinct.docs.amd.com)
DCGM_FI_DEV_FB_FREE	GPU_FREE_VRAM	MiB vs MB gauges; capacity	Capacity snapshot; distinguish reserved memory	Gauge; no counter reset	Verify device and partition visibility	Approximate	(github.com) (instinct.docs.amd.com)
DCGM_FI_DEV_GPU_TEMP	GPU_EDGE_TEMPERATURE, GPU_JUNCTION_TEMPERATURE, GPU_MEMORY_TEMPERATURE	Celsius gauges; separate sensor APIs	Sensor snapshot; choose edge, junction, or memory before aggregation	Gauge; no counter reset	AMD edge MI2xx, junction MI3xx; validate permissions	Approximate	(github.com) (instinct.docs.amd.com)
DCGM_FI_DEV_POWER_USAGE	GPU_POWER_USAGE	Watts gauges; DCGM board / AMD socket	DCGM field 155 device power (not the raw instantaneous field 157) versus AMD generation-specific current or average	Gauge; no counter reset	AMD secondary partition can report zero; primary owns sensor	Approximate	(docs.nvidia.com) (docs.nvidia.com) (rocm.docs.amd.com)
DCGM_FI_DEV_TOTAL_ENERGY_CONSUMPTION	GPU_ENERGY_CONSUMED	mJ vs microjoules cumulative counters; device API	Convert to joules; apply rate before summing devices	Driver reload vs restart or GPU reset	AMD physical sensor scope; verify primary partition	Comparable	(docs.nvidia.com) (rocm.docs.amd.com)

DCGM key	AMD configuration key	Unit, kind and source API	Sampling and aggregation	Reset or rollover	Support and privilege	Grade	Source URLs
DCGM_FI_DEV_SM_CLOCK	GPU_CLOCK / gpu_clock	MHz gauges; selected clock domain	Clock snapshot; select AMD clock_type before aggregating	Gauge; no counter reset	Check enabled clock field and device support	Approximate	(github.com) (instinct.docs.amd.com)
DCGM_FI_PROF_PCIE_TX_BYTES	PCIE_BANDWIDTH	Bytes/s directional vs Mb/s aggregate gauges	NVIDIA interval throughput; AMD aggregate instantaneous	Gauge; no counter reset	NVIDIA profiler privilege; AMD MI3xx primary scope	Unavailable	(github.com) (rocm.docs.amd.com)
DCGM_FI_DEV_NVLINK_BANDWIDTH_TOTAL	GPU_XGMI_LINK_TX / GPU_XGMI_LINK_RX	MB/s gauge vs cumulative KB per link	Aggregate versus per-link; rate AMD values only after validation	AMD rollover unspecified; verify live counter	XGMI topology and architecture dependent	Compatible	(github.com) (instinct.docs.amd.com)
DCGM_FI_DEV_XID_ERRORS	GPU_AFID_ERRORS	Last-code status or event-like gauges	Last observed code, not an event rate	Code clearing/reset unspecified; retain native series	Vendor code sets and device support differ	Vendor-specific	(github.com) (instinct.docs.amd.com)
DCGM_EXP_GPU_HEALTH_STATUS	GPU_ECC_UNCORRECT_* and AMD health fields	Status gauge versus error counters	Watch state versus subsystem totals	Exporter lifecycle vs hardware counter reset needs testing	NVIDIA health opt-in; AMD ECC support varies	Unavailable	(docs.nvidia.com) (instinct.docs.amd.com)

The default exporter series DCGM_FI_DEV_POWER_USAGE maps to DCGM_FI_DEV_BOARD_POWER_WATTS (field 155), which NVIDIA describes as “Power usage for the device in Watts.” The separate DCGM_FI_DEV_BOARD_POWER_RAW_WATTS (field 157) is the one described as “Current instantaneous power usage of the device in Watts.” (docs.nvidia.com) (docs.nvidia.com)

The table supports query design, not a promise of identical thresholds. It also illustrates why a simple `label_replace()` cannot repair a semantic mismatch. Capacity can be normalized mathematically, while activity and temperature still need sensor and time-window validation. The exported `gpu_clock` field may represent several clock types, so the target clock type must be selected on AMD before recording it under a neutral name. (prometheus.io) (grafana.com)

Mappings to reject

Table 2 lists tempting translations that would erase information. The row identifies the operational question that should replace the rename.

Tempting translation	Why it fails	Replacement decision	Evidence
GPU_UTIL equals GFX_ACTIVITY	Device-sampled busy percentage and graphics-engine activity have different sample semantics.	Plot both natively and set an AMD baseline for the same workload.	(github.com) (instinct.docs.amd.com)
UMC_ACTIVITY equals used VRAM	Engine activity is time busy; VRAM use is occupied capacity.	Keep separate bandwidth-pressure and capacity panels.	(rocm.docs.amd.com) (instinct.docs.amd.com)
Last XID equals AFID count	Both are last event identifiers, with vendor-specific code spaces.	Record raw code and separately count observed event changes.	(github.com) (instinct.docs.amd.com)
NVLink bandwidth equals XGMI beats	One is aggregate throughput; the other can be per-neighbor data beats.	Preserve link topology, multiply beats by 32 bytes only where documented, then derive a rate.	(github.com) (instinct.docs.amd.com)
Pod GPU usage equals activity	A pod-level metric can mean allocated GPU count.	Check the producing API and label join before charting utilization.	(docs.aws.amazon.com) (kubernetes.io)

The do-not-translate cases protect alerts from false parity. For example, an allocated GPU count may be stable while compute activity falls, and a missing per-partition power sensor may legitimately report zero. Neither observation alone establishes an idle or healthy accelerator. ([docs.aws.amazon.com](#)) ([prometheus.io](#))

Performance and Benchmarks

This article does not benchmark hardware. A telemetry schema comparison cannot establish throughput or efficiency, and no controlled cross-vendor GPU workload was run. The measurable comparison is **observability fidelity**: does each selected time series cover the same entity, interval, quantity, and aggregation? NVIDIA profiling metrics are interval averages, not a kernel trace; AMD's optional profiler metrics are disabled by default and have a separately configured sampling window. Those facts make profiler and general-purpose utilization panels different instruments. ([docs.nvidia.com](#)) ([instinct.docs.amd.com](#))

A useful test workload alternates a known busy interval with an idle interval while both exporters are scraped. An AWS operational example demonstrates **20 seconds at full utilization followed by 20 seconds at zero**. That example is a reproducible shape for checking lag and smoothing, not a cross-vendor benchmark result. Compare raw samples first, then use equal recording windows. DCGM's **30-second** default watch cadence means a short burst can be undersampled even if Prometheus scrapes more frequently. AMD's health service has a separate **30-second** default polling interval, which should not be mistaken for every metric's sampling period. ([aws.amazon.com](#)) ([docs.nvidia.com](#)) ([instinct.docs.amd.com](#))

For capacity panels, use a controlled allocation and release, then inspect byte scale and reserved-memory behavior. For power, hold a stable workload long enough to compare gauge medians and the derivative of energy counters after conversion. For interconnect, test an explicitly directional host-device transfer and a peer transfer separately. The NVIDIA PCIe direction is defined from the GPU's perspective; AMD's exporter

aggregate PCIe bandwidth cannot provide that direction in the documented schema. AMD SMI's separate API can return bytes sent and received over one second, but that is an integration decision beyond a name swap. (docs.nvidia.com) (instinct.docs.amd.com) (rocm.docs.amd.com)

For Device Metrics Exporter 1.5.2 on MI250, AMD reports that reading PCIe bandwidth takes about one second per GPU because the kernel `pcie_bw` sampler blocks per read. Validate the effective scrape duration when enabling this field. (instinct.docs.amd.com)

Data Analysis and Evidence

The most useful numerical data here are **unit and collection-contract differences**, not market-size estimates. A dashboard migration can be wrong by a scale factor even when graphs look plausible. The calculation below follows each vendor's published unit and NIST's distinction between decimal and binary byte multiples. It is a schema conversion, not a measured hardware result. (github.com) (physics.nist.gov)

Table 3 gives arithmetic templates for a neutral recording layer. **The uppercase AMD expressions are configuration-key pseudocode, not executable PromQL.** Before writing a query, substitute the exact series names observed in the deployment's `/metrics` output, including its configured prefix. Retain the unit-scale, source-type, sampling, and reset checks below.

Neutral quantity	NVIDIA expression	AMD expression	Interpretation and source
Used VRAM bytes	DCGM_FI_DEV_FB_USED * 1048576	Configuration-key pseudocode: GPU_USED_VRAM * 1000000	MiB versus MB, assuming vendor labels describe the runtime scale; verify with allocation. (github.com) (instinct.docs.amd.com) (physics.nist.gov)
Energy joules	DCGM_FI_DEV_TOTAL_ENERGY_CONSUMPTION / 1000	Configuration-key pseudocode: GPU_ENERGY_CONSUMED / 1000000	Cumulative energy; reset on driver reload versus restart or GPU reset. (docs.nvidia.com) (instinct.docs.amd.com) (rocm.docs.amd.com)
Average watts from energy	rate(energy_joules[window])	rate(energy_joules[window])	Counter rate in joules per second; calculate rate before aggregation. (prometheus.io) (prometheus.io) (opentelemetry.io)
XGMI bytes sent	Not a portable NVLink metric	Configuration-key pseudocode: GPU_XGMI_NBR_0_BEATS_TX * 32	Per-neighbor cumulative data size only; sample and rollover need validation. (instinct.docs.amd.com) (rocm.docs.amd.com)
PCIe throughput bytes/s	DCGM_FI_PROF_PCIE_TX_BYTES	No documented same-scope exporter series	NVIDIA direction is GPU transmit; AMD's aggregate Mb/s is not a directional replacement. (docs.nvidia.com) (instinct.docs.amd.com)

The table's **1,048,576** and **1,000,000** multipliers differ by about **4.86%** when translating an equal displayed count into bytes. That percentage is arithmetic from the stated units, not an independent vendor measurement. NIST also cautions that historical use of “MB” may vary, so a controlled allocation remains part of the validation. (physics.nist.gov) (grafana.com)

Counter treatment is similarly quantitative. A driver reload resets NVIDIA's energy total, and AMD's energy accumulator ties to restart or GPU reset. Prometheus `rate()` adjusts detected counter resets; it should be applied before summing across GPUs. It must not be applied to a power or temperature gauge. That distinction determines whether an energy-based power panel is an honest rate or an arbitrary slope through a reset. (docs.nvidia.com) (rocm.docs.amd.com) (prometheus.io) (prometheus.io) (prometheus.io)

Finally, counts in the metric names are not always counts of events. `DCGM_FI_DEV_XID_ERRORS` is a last-code gauge. Optional exporter XID count series count sampled records in a window, and totals reset with exporter lifecycle events. AMD publishes correctable, uncorrectable, and deferred error categories. A cross-vendor "error rate" requires a written event taxonomy and observation rules, not merely addition across all RAS fields. (github.com) (docs.nvidia.com)

Dashboard Migration Procedure

Inventory, normalize, validate

A migration should start with the live metric surface and end with a recorded decision for each old rule. The operator should retain the original vendor metric beside any neutral recording rule until both can be inspected under a controlled workload. This matters because NVIDIA's configured rows can be absent, and AMD can prefix fields or change what a partition exposes. Prometheus recording rules create new series; they do not change the evidence produced by the exporter. (docs.nvidia.com) (rocm.docs.amd.com) (prometheus.io)

1. **Freeze versions and topology.** Record GPU model, firmware, driver, DCGM or AMD SMI version, exporter image digest, collector configuration, partition mode, and privilege set. Pin NVIDIA's paired exporter/DCGM release and AMD's supported ROCm matrix. (docs.nvidia.com) (instinct.docs.amd.com)
2. **Inventory effective series.** Save `/metrics` samples from each deployment mode, including primary and secondary partitions or guests. Compare configured fields with observed names, types, help text, and labels. An enabled field is not proof of emission. On dense MI250 nodes with PCIe bandwidth enabled, account for delayed or empty metric responses when assessing scrape completeness. (instinct.docs.amd.com)
3. **Define stable identity.** Join by physical device identifier only where present; preserve partition, link, and pod labels. NVIDIA MIG adds instance labels, AMD CPX can create multiple handles per socket, and Kubernetes pod attribution uses the PodResources API. (rocm.docs.amd.com) (kubernetes.io)
4. **Normalize units and types.** Create distinct neutral names for bytes, joules, watts, degrees Celsius, and ratios. Convert MiB and MB explicitly. Keep cumulative values as counters and instantaneous values as gauges. (physics.nist.gov) (prometheus.io) (prometheus.io) (opentelemetry.io)
5. **Annotate availability.** Mark each mapping by GPU generation, driver, bare metal or guest, hypervisor role, privilege, and collector mode. For example, NVIDIA profiling needs `SYS_ADMIN`, AMD profiler access can need `CAP_PERFMON`, and AMD per-VF VRAM use is not visible on a GIM host.
6. **Check cardinality.** Count series before adding link, partition, process, pod, or code labels. Each distinct label set consumes Prometheus resources; a per-link or per-event dashboard can multiply series unexpectedly. (prometheus.io)

7. **Re-baseline alerts.** Run vendor-native and neutral panels together through representative idle, loaded, and partitioned conditions. Keep separate missing-series alerts because Prometheus marks disappearing series stale. Use a for duration chosen from the real sample cadence and response objective. (prometheus.io) (prometheus.io)
8. **Document exceptions.** Leave XID, AFID, health status, NVLink, and XGMI as vendor-native names unless a tested higher-level question justifies an explicit derived rule. Preserve the grade and source URL in the dictionary. (github.com) (github.com) (rocm.docs.amd.com)

Alert evidence templates

The following Prometheus Query Language (PromQL) expressions are **templates**, not universal limits. Replace placeholder thresholds with values derived from a pinned fleet's observed distribution and operational objective. The expressions assume neutral recording rules already normalize units, identity, and counter resets. Prometheus recommends `rate()` before aggregation, and `absent()` is available for missing-series detection. (prometheus.io) (prometheus.io)

```
max by (cluster, physical_gpu) (gpu_temperature_celsius{sensor="chosen"}) > TEMPERATURE_LIMIT_C

sum by (cluster, physical_gpu) (rate(gpu_energy_joules_total[WINDOW])) > POWER_LIMIT_W

max by (cluster, physical_gpu) (expected_physical_gpu{cluster="CLUSTER"})
  unless on (cluster, physical_gpu)
  max by (cluster, physical_gpu) (gpu_temperature_celsius{cluster="CLUSTER", sensor="chosen"})
```

Populate the template `expected_physical_gpu` series from inventory, with one `cluster` and `physical_gpu` label pair per expected device. A fixed `absent()` selector checks only its specified metric-and-label combination. (prometheus.io)

A temperature limit must name the sensor: AMD documents edge and hotspot or junction readings, while NVIDIA's GPU and memory temperatures are separate fields. A power limit should identify whether it is board, socket, or partition scope. If an AMD secondary partition reports zero because the physical sensor belongs to the primary partition, a zero-power alert is a schema error. Grafana's display unit option changes presentation, not the stored metric, so conversion belongs in the query or recording rule. (instinct.docs.amd.com) (github.com) (grafana.com)

Implications and Future Directions

A mixed-fleet procurement review should ask for an **observability acceptance matrix** alongside hardware and software compatibility. The matrix needs the actual `/metrics` payload for each ordered accelerator and deployment mode, a field-by-field support statement, version-pinned documentation, and a path to expose the selected labels to the monitoring plane. This is a buyer's evidence request, not a prediction about any vendor's future releases. AMD's current exporter release notes list supported ROCm combinations; NVIDIA links an exporter release to a DCGM version. (instinct.docs.amd.com) (docs.nvidia.com)

The same review should identify who can access device files, host-engine sockets, Kubernetes pod resources, and profiling counters. Kubernetes says pod-resource monitoring agents require a privileged security context; NVIDIA profiling and AMD profiler collection have their own documented capability requirements. A signal that exists in an API but cannot be collected under the site's security model is operationally unavailable. SR-IOV hosts and guests deserve separate rows because their visible fields and labels differ. (kubernetes.io) (docs.nvidia.com)

Common conventions may reduce naming friction, but they do not erase sensor differences. OpenTelemetry's GPU convention is still marked as developing and distinguishes usage from memory utilization; its common hardware convention models energy as a joule counter and power as a watt gauge. A portable layer can adopt those units while retaining vendor, source_api, sensor, physical_gpu, partition, and support_grade as explicit metadata. The remaining vendor-native panels are a feature of honest observability, not migration debt. (opentelemetry.io) (opentelemetry.io) (opentelemetry.io) (prometheus.io)

GPU Smith's public operations scope includes telemetry and procedures for existing facilities. In that advisor posture, the defensible recommendation is a measured mapping record and a small live validation on the specified fleet before an alert becomes production policy. The report supplies the schema questions, while actual thresholds and reliability behavior remain properties of the deployed systems and workloads. (gpusmith.com) (prometheus.io)

Frequently Asked Questions (FAQs)

Is AMD SMI the same as ROCm SMI?

AMD calls **AMD SMI** the successor to ROCm SMI. Its API and command-line output should be treated as a source of AMD telemetry, while the Device Metrics Exporter schema determines the Prometheus names used by a dashboard. As of this report, the exporter documentation is labeled **1.5.2**. (rocm.docs.amd.com)

What is the AMD equivalent of NVIDIA GPU utilization?

GPU_GFX_ACTIVITY is the closest general AMD exporter field for graphics-engine busy percentage, but the mapping is **approximate**. GPU_UMC_ACTIVITY measures memory-engine activity, not VRAM occupied. NVIDIA DCGM_FI_PROF_SM_ACTIVE is a profiling interval ratio with another scope. Validate all three under the same workload before reusing a threshold. (rocm.docs.amd.com) (github.com) (docs.nvidia.com)

Can DCGM Exporter rules be reused with AMD Device Metrics Exporter?

Rules that normalize device identity, bytes, joules, and watts can often be **composed** from documented fields after unit conversion. Alerts for temperature, utilization, PCIe, NVLink or XGMI, ECC, and event identifiers need a new scope statement or a vendor-native branch. An exporter label prefix, optional DCGM collector, unsupported field, or guest-mode omission can make a syntactically correct query return no series. (docs.nvidia.com)

Are missing GPU metrics equivalent to zero?

No. Prometheus marks a previously present series stale when a target stops returning it, and `absent()` is intended for missing-series alerts. AMD's documented partition modes can yield zero on a secondary partition for a sensor available only on the primary, while other fields may disappear entirely on a guest or host. Maintain explicit missing-series and scope checks. (prometheus.io) (prometheus.io)

Conclusion

The DCGM to AMD SMI migration is a translation of **measurement contracts**, not a search-and-replace of metric names. Capacity and energy can be normalized through stated units and reset rules; activity, temperature, PCIe, interconnect, health, and event codes carry important scope differences. A dashboard can display both vendors under shared headings only after the underlying series have been graded and tested. (docs.nvidia.com) (rocm.docs.amd.com)

For an SRE or platform team, the deliverable is a versioned dictionary, a saved sample of each exporter's live output, and an alert review that records which thresholds were re-baselined. Keep physical and partition identity, missing-series behavior, and privilege requirements explicit. Preserve vendor-native panels where no equivalent exists. The current documents provide a strong specification for that work, but they do not replace validation on the actual GPUs, drivers, partition modes, and monitoring cadence. (docs.nvidia.com) (rocm.docs.amd.com) (prometheus.io)

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/dcgmlatest/release-notes/dcgml-exporter.html>
2. <https://instinct.docs.amd.com/projects/device-metrics-exporter/en/release-v1.5.2/installation/kubernetes-helm.html>
3. <https://github.com/NVIDIA/dcgml-exporter/blob/main/etc/default-counters.csv>
4. <https://instinct.docs.amd.com/projects/device-metrics-exporter/en/latest/configuration/metricslist.html>
5. <https://docs.nvidia.com/datacenter/dcgmlatest/dcgml-api/dcgml-api-field-ids.html>
6. <https://rocm.docs.amd.com/projects/amdsmi/en/latest/reference/amdsmi-py-api.html>
7. <https://physics.nist.gov/cuu/Units/binary.html>
8. <https://docs.nvidia.com/datacenter/dcgmlatest/learn/modules/profiling.html>
9. <https://prometheus.io/docs/practices/instrumentation/>
10. <https://prometheus.io/docs/prometheus/latest/querying/basics/>
11. <https://prometheus.io/docs/prometheus/3.9/querying/functions/>
12. <https://prometheus.io/docs/practices/naming/>
13. <https://opentelemetry.io/docs/specs/semconv/hardware/gpu/>
14. <https://gpusmith.com/>
15. <https://gpusmith.com/articles/en/dcgml-exporter-gpu-monitoring-metrics-alerts-triage>
16. <https://docs.nvidia.com/datacenter/dcgmlatest/reference/command-line-reference/dcgml-exporter.html>

17. <https://rocm.docs.amd.com/projects/amdsmi/en/latest/index.html>
18. <https://docs.nvidia.com/datacenter/dcgm/latest/reference/dcgm-exporter-metrics.html>
19. <https://rocm.docs.amd.com/projects/amdsmi/en/latest/conceptual/partition.html>
20. <https://gpusmith.com/articles/en/good-gpu-utilization-percentage>
21. <https://docs.nvidia.com/datacenter/dcgm/latest/installation/install-dcgm-exporter.html>
22. <https://aws.amazon.com/blogs/machine-learning/enable-pod-based-gpu-metrics-in-amazon-cloudwatch/>
23. <https://docs.cloud.google.com/kubernetes-engine/docs/how-to/dcgm-metrics>
24. <https://instinct.docs.amd.com/projects/device-metrics-exporter/en/latest/releasenotes.html>
25. <https://instinct.docs.amd.com/projects/device-metrics-exporter/en/release-v1.5.2/configuration/configmap.html>
26. <https://grafana.com/docs/grafana/latest/visualizations/panels-visualizations/configure-standard-options/>
27. <https://raw.githubusercontent.com/NVIDIA/dcgm-exporter/4.6.0-4.8.3/etc/default-counters.csv>
28. <https://instinct.docs.amd.com/projects/device-metrics-exporter/en/release-v1.5.2/configuration/metricslist.html>
29. https://raw.githubusercontent.com/NVIDIA/DCGM/v4.6.0/dcgmlib/dcgm_fields.h
30. <https://grafana.com/docs/grafana/latest/visualizations/dashboards/variables/add-template-variables/>
31. <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-metrics-EKS.html>
32. <https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/device-plugins/>
33. <https://rocm.docs.amd.com/projects/amdsmi/en/latest/reference/amdsmi-cpp-api/functions.html>
34. <https://opentelemetry.io/docs/specs/semconv/hardware/common/>
35. <https://rocm.docs.amd.com/projects/rocprofiler-systems/en/latest/how-to/xgmi-pcie-sdma-sampling.html>
36. https://prometheus.io/docs/prometheus/latest/configuration/recording_rules/
37. https://prometheus.io/docs/prometheus/latest/configuration/alerting_rules/
38. https://prometheus.io/docs/instrumenting/writing_exporters/

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.