How DMS measures GPU workload power

Method description, version 1.1, November 1, 2026. Dynamic Margin Systems. Contact: Liam Bronson, founder.

This document says what DMS reads, how often, what each recorded field means, how the offline tools compute their results and where the method stops being valid. It is written so that someone checking a DMS report can trace every number back to a reading.

What is read, and how often

Source	How it is read	Rate	Notes
NVIDIA GPUs	NVML, per card: average power, instantaneous power where the card exposes it, power limit, utilization, temperature, clocks	10 readings a second	Read-only unless a power budget is configured
GPU energy counters	The card's cumulative energy counter (NVML total energy, or AMD SMI energy count), differenced into watts	With every reading	The default power source on AMD, where published work found a 1 ms refresh on MI250X and MI300A (not yet confirmed on MI300X). On NVIDIA used only on request: published tests found it no faster than about 100 ms and under-reading on some cards
NVIDIA driver sample buffer	The driver's own buffer of power samples (NVML total power samples), drained about once a second	The card's own spacing: about 20 ms reported on one Blackwell card, not yet measured on Hopper or Ampere	Optional (`DMS_SAMPLE_BUFFER=1`). The default power source in reports when it was logged

Source	How it is read	Rate	Notes
AMD GPUs	AMD SMI, per card: socket power and the same companion fields	10 readings a second	Same record format as NVIDIA
CPU packages	Linux RAPL energy counters, differenced into watts	Once a second	Optional (`DMS_CPU_RAPL=1`)
Node or rack meter	Redfish chassis power, or a site command that prints watts	Every 5 seconds	Optional
Facility points	Modbus TCP or BACnet/IP reads of site meters, cooling, UPS output and PDUs	Every 5 seconds by default	Optional, read-only
UPS or battery output	A site command that prints watts, positive while discharging to the load	Every 5 seconds	Optional (`DMS_BUFFER_CMD`)
Site recorder files	COMTRADE or CSV from the site's disturbance or power-quality recorder	The recorder's own rate, often hundreds a second	Imported offline

Nothing is read from the network unless a meter, facility point, buffer or telemetry endpoint is configured. DMS opens no listening ports.

The telemetry record

One JSON line is written for each 0.1 second reading. The fields a reviewer needs:

Field	Meaning
`timestamp_utc`	The node's system clock at the reading, UTC. DMS logs the clock's sync status and offset every 10 minutes (`clock_status` events)
`per_gpu.<id>.power_draw_w`	The vendor library's power reading for that card, averaged over a short window inside the card
`per_gpu.<id>.power_draw_instant_w`	The instantaneous reading where the card provides one. Empty where it does not
`per_gpu.<id>.energy_mj_total`	The card's cumulative energy counter, millijoules
`node_sum_w`	Sum of the cards on the node at the last control step
`node_budget_w`	The power budget in force, if any. Empty when DMS is only recording
`cpu_package_w`	Sum of CPU package power from RAPL, when enabled
`external_meter_w`, `external_meter_utc`	Latest node or rack meter reading and when it was taken
`facility_by_role_w`	Latest facility readings by role (site meter, cooling, UPS output, PDU)
`site_buffer`	What the site declared sits between the computers and the grid: none, ups, bess, ups+bess, or unknown

At every start, DMS also records the node's driver version, kernel, card models, firmware (VBIOS) versions and power limits, and logs any difference from the last start as an equipment change.

The offline tools

Each tool reads telemetry files and, where needed, a meter, recorder or model file, and writes a JSON result with a plain-text summary. Any result can be sealed with a signature so later edits are detectable. Every result states which power source it used and what each card's reading covers.

Fast capture (`--fast-capture`). Reads every card as fast as it answers, along with its energy counter and the driver's own sample buffer, and reports how often each reading really changes. The result is kept on the node and cited by every later report as the measured sensor behavior.

Burst test (`nvml_burst_test.py`, in the kit). Runs the same short burst load at ten phase offsets on one NVIDIA card. The true average is identical in every phase, so a source whose phase averages spread out is missing time, and one that stays flat is covering it. This settles, per card, whether its energy counter or sample buffer sees what the instant field misses.

Planning profile (`--planning-profile`). Builds a description of the load for planners: levels as a share of nameplate, the largest changes within 0.5, 1, 5 and 60 second windows, classified steps (job starts, drops to idle, dips that recover), and frequency content by Welch's method.

Tenant report (`--tenant-report`). Puts every card on one time grid and groups the cards by tenant, from a file naming each tenant's nodes or GPUs. For the site and each tenant it gives the largest change inside any 5 seconds and the largest rise and fall inside any 60 seconds. Optionally it scales the measured site to a stated size and checks it against a 5-second swing limit and a per-minute ramp limit. For the site's worst 5-second swings it shows each tenant's change and share. Scaling multiplies every reading by one factor, which assumes every node moves the way the measured ones did, at the same moment.

Recovery report (`--recovery-report`). For each voltage dip, from dip times supplied by the utility or recorder, or from a voltage channel, it finds the load before the dip and how long the site and each tenant took to get back to a stated share of it (90% by default) and stay there. A mode that finds candidate events from power drops alone is available, labeled as candidates, because a job ending looks the same.

Change report (`--change-report`). Compares the load before and after an equipment change: levels, swings and ramps. It notes the 180-day verification window in NERC's draft CLO-001. Whether a change qualifies is the site's and the planner's decision.

Recorder import (`--recorder-import`). Reads COMTRADE (1991, 1999, 2013; ASCII and binary formats) or CSV from a site recorder. It computes power from a power channel or from three-phase voltage and current, averaged over each power-system cycle. It writes a power series that site attribution lines up with every node.

Site attribution (`--site-attribution`). Lines up node telemetry with a site meter or recorder. It finds the clock offset from the data and lists every meter swing above a threshold, with how much of it the GPUs moved, which node and which cards. For meters that read faster than twice a second, it also reports the meter's own frequency content by band.

Proxy verification (`--verify-proxy`) and **model validation** (`--validate-model`) are unchanged from version 1.

Limits

These are stated in every result file, and they apply to every number DMS publishes.

1. **Ten readings a second sees up to 5 Hz.** Card readings cannot describe the 5 to 55 Hz range grid operators measure. That range comes from the site's own recorder, which DMS imports and lines up with the nodes.
2. **Each power reading covers only part of the time between readings.** Published measurement found that on A100 and H100 cards the driver updates power about every 100 ms, and each reading covers only the last 25 ms (Yang, Adamek and Armour, arXiv 2312.02741). On our own A100 PCIe run the instant field matched the averaged one in 99.97% of samples. To see more than that, reports use the best source the card offers: on AMD the energy counter, on NVIDIA the driver's own sample buffer when it was logged, otherwise the instant field. On NVIDIA the energy counter is not trusted to cover the gaps until a burst test on that card shows it does: published tests found it updates no faster than about 100 ms and under-reads on some cards. Every report states, per card model, what its readings cover, from fast capture where it was run and published figures otherwise.
3. **Device readings are not revenue meters.** Their accuracy is the vendor's. Proxy verification against a meter is how a site finds out how close they are on its own hardware.
4. **Card readings leave things out.** CPUs, memory, networking, storage, cooling and power conversion are outside them unless CPU readings, facility points or a meter are added.
5. **A UPS or battery changes what the grid sees.** Behind one, GPU readings describe the compute load, not the grid connection.
6. **A controlled load is not a raw load.** When DMS holds cards below their limits, the record describes the load as DMS shaped it, and says how often that happened.
7. **Lining up with a recorder needs agreeing clocks.** Node timestamps are only as good as NTP or PTP on the host. Reports state the worst logged offset.

Measured data so far

Run	Hardware	Length	Rate	Setting
October 26, 2026	2 × NVIDIA A100 80GB PCIe	10.7 hours, 386,103 readings	10 a second	Rented cloud machine, scripted training and inference load. No meter

The tenant, recovery, change and recorder tools have run only on simulated data. Their first real inputs will be a customer site's telemetry and recorder files.