GPU SMITH
COMPUTE ENGINEERING

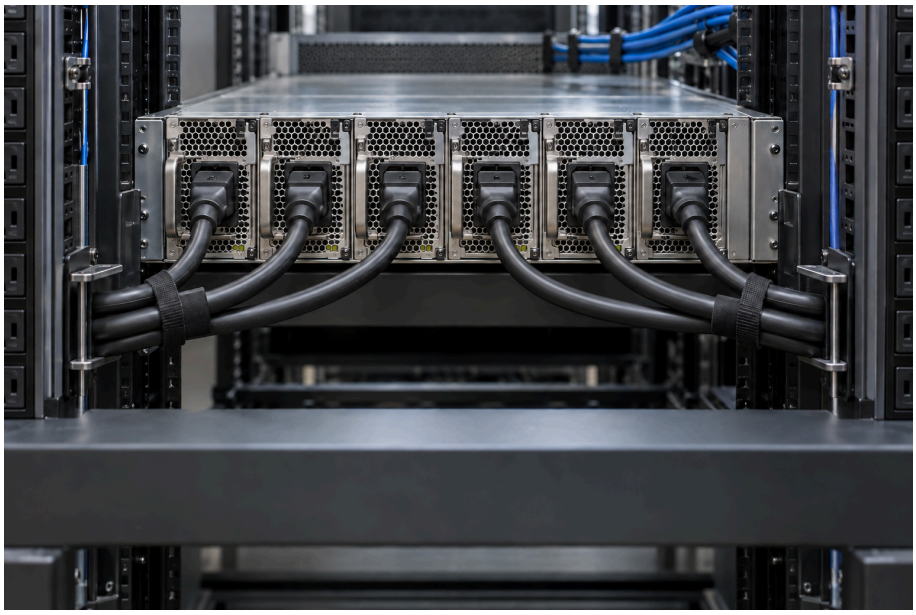
GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

GPU Power Cap Optimization for Inference: Cost and SLOs

> 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 · 2026-10-04 · gpu power cap optimization for inference · gpu power capping · inference latency · llm inference efficiency · energy per token · nvidia gpu power limits · slo goodput · private ai infrastructure · gpu capacity cost

Original article: https://gpusmith.com/articles/en/gpu-power-cap-economics-inference-slos



In this article

1. Executive Summary
2. Introduction and Background
3. Key Changes
4. Implementation Considerations and Process Changes
5. Measurement Boundaries and SLO-Qualified Output
6. Constructing the Power and Cost Frontier
7. Production Guardrails and Training Interpretation
8. Data Analysis and Evidence
9. Case Studies and Real-World Examples
10. Implications and Future Directions
11. Conclusion

Executive Summary

Graphics processing unit (GPU) power cap optimization for inference should minimize the cost of accepted output within a declared power envelope and service-level objective (SLO). The optimum is a measured operating point for one model, serving engine, request trace and hardware configuration. It cannot be inferred from thermal design power (TDP). NVIDIA's documented **DGX B200** control policy selects the most conservative applicable limit, so an operator must verify the effective limit rather than assume the host's requested setting governs the device. (docs.nvidia.com) Zeus similarly makes selection of an optimal limit depend on the user's criteria. (ml.energy)

The measurement boundary changes the answer. MLCommons' fetched power-policy document requires power and performance from the **same run** and measures alternating-current (AC) power at the wall. (github.com) (github.com) This report extends that principle to completed requests and accepted output tokens. Client-observed time to first token (TTFT), streaming latency and errors must be checked against declared deadlines, with aggregate percentile limits assessed separately. The versioned **vLLM 0.12.0** benchmark interface supports request-level goodput objectives for TTFT, time per output token (TPOT) and end-to-end latency. (docs.vllm.ai) A power reduction that increases rejected output, queueing or the number of replicas can increase total cost.

Published evidence illustrates the tradeoff without identifying a universal cap. Dell's **June 27, 2024** vendor study reports **15% lower system power** with **6% lower performance** at a **600 W per-GPU** setting in its tested offline workload; that result does not establish an interactive latency optimum. (infohub.delltechnologies.com) In the explicitly hypothetical calculation below, a middle cap needs **16 GPUs** and **13.92 kW** of estimated facility power, while a lower cap needs **20 GPUs** and **15.00 kW** to serve the same target. Those are declared arithmetic assumptions, not benchmark results. The middle point has the lowest modeled monthly cost in all three illustrated price and capacity-cost scenarios.

Choose a cap by constructing a **Pareto frontier** of input power, SLO-qualified throughput, energy per accepted output and deployable capacity. Use power usage effectiveness (PUE) only to translate an information technology (IT) energy boundary into a declared facility estimate; its definition already compares

facility and IT power. (www.energy.gov.au) Evaluate demand charges against the site's billing peak, not GPU average watts. (www.energy.gov) Keep the default limit when savings disappear after capacity rounding or the required SLO margin vanishes. For training, use energy and time to a fixed validation target, consistent with Zeus's stated objective, rather than treating inference tokens as a training outcome. (ml.energy)

Introduction and Background

A rack with spare electrical capacity and a rack constrained by its redundant power path face different decisions. An electricity budget adds another objective; an interactive application adds a latency constraint; a scheduled batch job adds a completion deadline. A useful experiment states which constraint is binding before changing a watt limit. Otherwise, a lower number on a dashboard can mask longer occupation of an expensive server.

Power is an instantaneous rate and **energy** accumulates over time. The US Energy Information Administration (EIA) distinguishes watts at a particular moment from electricity consumed over time, and defines a kilowatthour (kWh) as one kilowatt for one hour. (www.eia.gov) (www.eia.gov) Consequently, saving watts does not establish that the same completed work used fewer kWh. A longer run can consume the initial saving, and unused capacity can require another serving replica.

This report takes the perspective of an independent engineering advisor. GPU Smith's first-party description connects cluster sizing and serving-stack selection to workload models and throughput targets. (gpusmith.com) That perspective makes a reproducible acceptance test more useful than a recommended cap detached from the workload. The report does not claim access to private client results; all published measurements are attributed, and all scenario values are identified as assumptions.

Evidence was accessed on **October 4, 2026**; historical measurements retain their original scope. The scope is one declared inference workload at a time. Compare the same model weights, precision, engine, tokenization, request mix and acceptance definition across caps. A new model or engine is a new experiment. Original research on large language model (LLM) inference also finds that power, energy and performance objectives can select different frequency configurations; frequency tuning is related evidence, rather than an interchangeable watt-cap intervention. (arxiv.org)

The resulting decision is practical: select a supported setting that meets the rack envelope and application requirements at an acceptable full-capacity cost, or retain the baseline. Neither the lowest wattage nor the highest tokens per GPU automatically wins. The remainder specifies how to collect enough evidence to make that choice reviewable.

Key Changes

Replace a nominal cap with an effective control policy

There is no single universal layer at which every server applies a power limit. NVIDIA's DGX B200 guide lists firmware video BIOS (VBIOS), host **nvidia-smi** and out-of-band System Management Bus Post Box Interface (SMBPBI) sources. (docs.nvidia.com) Its worked example shows a **6 kW system limit** associated with a **562**

W per-GPU Redfish setpoint. (docs.nvidia.com) This is evidence of interaction in the documented system, not a conversion formula for arbitrary servers.

Table 1 summarizes the control surfaces and the verification each requires.

Control surface	Documented behavior	Operator verification
Host GPU limit	nvidia-smi requires a value within reported minimum and maximum power limits. (docs.nvidia.com)	Record device identity, requested setting and effective limit.
BMC and Redfish	The DGX B200 example queries a per-GPU power-limit setpoint through Redfish. (docs.nvidia.com)	Identify whether the resource limits the server, module or individual GPU.
Site scheduler	NERSC exposes power limiting through a site-developed Slurm plugin. (docs.nersc.gov)	Verify plugin availability, job scope and cleanup policy.
Frequency request	Slurm documents that system power caps can override requested GPU frequencies. (slurm.schedmd.com)	Treat frequency and watt-limit experiments as separate interventions.
AMD device tooling	AMD SMI can monitor GPU power usage and cap values in watts. (rocmdocs.amd.com)	Use the installed vendor interface and inspect supported limits.

These interfaces provide control, but they do not make their measurements equivalent. The experiment should have one owner for each setting and a record of all other active controls. A host request above a tighter system policy can produce a nominally different test with the same effective operating point.

Read supported limits from the exact machine

The DGX B200 guide shows **200 to 1000 W** allowable limits and a **700 W** default in an example response. (docs.nvidia.com) These values belong to that documented example. Inventory output and vendor guidance for the actual GPU, board, server and firmware must govern the sweep. Even supported settings are candidates for testing, not promises about application throughput.

A practical inventory uses the documented inspection interface before applying a limit. (docs.nvidia.com)

```
nvidia-smi -L
nvidia-smi -q -d POWER
```

For an authorized host-level change, the following template uses placeholders for the device's universally unique identifier (UUID) and a supported watt value. The documented power-limit operation requires root privileges. (docs.nvidia.com)

```
sudo nvidia-smi -i GPU_UUID -pl CAP_WATTS
nvidia-smi -i GPU_UUID -q -d POWER
```

Record the output with the run configuration. Resolve stable physical identities rather than assuming device numbering remains unchanged after maintenance. Keep the original settings available for rollback and verify all devices used by a multi-GPU model.

Separate allocation from electrical control

Kubernetes GPU device plugins expose custom schedulable resources. (kubernetes.io) When GPU requests and limits are both specified, their values must match. (kubernetes.io) Those allocation declarations are not themselves proof of a watt limit. Similarly, Linux powercap represents controllable components as power zones, while its documented Intel Running Average Power Limit (RAPL) example concerns CPU-package domains. (cdn.kernel.org) A CPU-domain setting must not be substituted for a GPU control.

The operational change is to make electrical policy an explicit part of the deployment specification. Scheduler allocation, serving-engine placement and cap enforcement should be validated together, with ownership documented for each.

Implementation Considerations and Process Changes

Declare the workload and the acceptance contract

A cap sweep needs a workload manifest that remains fixed. State which output counts as useful and what constitutes rejection before examining results. **TTFT** is client-observed time from sending a request to the first streamed output in vLLM's benchmark definition. (docs.vllm.ai) **TPOT** excludes the first token and is calculated per request. (docs.vllm.ai) **Inter-token latency (ITL)** needs its own definition: under speculative decoding, one streamed output may contain multiple tokens, so stream gaps and per-token timing can differ. (docs.vllm.ai)

Keep the following manifest fields with every run:

- **Model:** weights identifier, tokenizer and allowed output-quality checks.
- **Precision:** weight, activation and cache formats.
- **Engine:** release or commit, build flags and runtime configuration.
- **Parallelism:** GPUs per replica and placement across nodes.
- **Requests:** input-length and requested-output-length distributions.
- **Traffic:** offered arrival process, rate and concurrency ceiling.
- **Acceptance:** request deadlines, error categories and useful-output rule.
- **Environment:** driver, firmware, clocks, temperature and cap sources.

Hugging Face Text Generation Inference (TGI) exposes an input-token-length histogram, which can help verify the trace's size distribution. (huggingface.co) Preserve the actual trace or a reproducible generator rather than recording only its mean. A mixture containing long prompts or long responses can behave differently from uniformly short requests; original inference research specifically observes queueing from longer outputs. (arxiv.org)

Preserve an honest baseline

The **baseline** is the supported default operating configuration without an additional experimental cap. It still has firmware and platform limits. (docs.nvidia.com) Record those limits and keep engine settings unchanged during the electrical sweep.

Run warmup before measurement and disclose its duration and energy treatment. Control compilation, loading and caches consistently. vLLM warns that repeated benchmarks against the same server can reuse prefix-cache contents and inflate throughput. (docs.vllm.ai) If production uses a warm cache, test a representative warm-cache state; if cold-start behavior matters, measure that separately. For PyTorch-based custom inference, verify evaluation mode for dropout and batch normalization before comparing outputs. (docs.pytorch.org)

Sweep safely and repeat independently

Use a small set of supported candidate caps, followed by refinement around promising points. Selection criteria remain workload-specific. (ml.energy) The procedure below is a proposed protocol rather than a claim about an optimal universal step size.

- **Inspect:** capture default, minimum, maximum and effective device limits.
- **Baseline:** replay the declared trace at the declared offered load.
- **Apply:** change only the intended electrical control.
- **Confirm:** inspect effective limits on every participating device.
- **Warm:** reach a comparable thermal and cache state.
- **Measure:** collect synchronized power and request records.
- **Restore:** return to the baseline between appropriate comparisons.
- **Repeat:** vary test order and retain run-level dispersion.
- **Reject:** discard configurations that fail declared eligibility checks.

Use separate tests for fixed-arrival-rate serving and saturated batch throughput. vLLM's benchmark documentation notes that a concurrency ceiling can reduce the actual arrival rate below the requested rate. (docs.vllm.ai) Report offered, admitted and completed traffic so a cap does not appear successful merely because the load generator supplied less work. Ray Serve's batching documentation also makes the queue explicit: requests wait to form batches. (docs.ray.io) Batch wait time should also be evaluated against the end-to-end SLO. (docs.ray.io) A batching change belongs in a separate experiment.

Measurement Boundaries and SLO-Qualified Output

Collect power and work over the same interval

The measurement record must align electrical input with the completed work it supported. (github.com) MLCommons uses benchmark start and end timestamps to select the corresponding power interval. (github.com) Zeus measurement windows likewise return elapsed time and energy for the same window. (ml.energy) Adopt that principle without claiming that this operational test is an MLPerf submission. Drain outstanding requests and include their completion tail in both records, or document a consistent boundary-attribution method.

GPU board power, server AC input, rack power distribution unit (PDU) input and facility input are different boundaries. nvidia-smi's documented instantaneous draw concerns the entire board. (docs.nvidia.com) MLCommons instead requires system-level power measurement, including the workload's supporting

components. (github.com) A baseboard management controller (BMC) reading is useful only when its documented sensor boundary, timing and accuracy are understood.

- **Primary energy meter:** state the AC boundary and meter identity.
- **Telemetry:** retain per-device draw, effective limit, utilization and clocks.
- **Time alignment:** record shared timestamps and missing samples.
- **Sampling:** disclose interval, averaging and integration method.
- **System overhead:** include central processing units (CPUs), memory, fans and supporting equipment.
- **Idle state:** measure a loaded, ready-to-serve system separately.
- **Transient checks:** retain peaks relevant to the platform constraint.
- **Uncertainty:** report repeat variation and meter limitations.

SPECpower distinguishes an active-idle interval with no workload transactions while power continues to be measured. (www.spec.org) Its CPU-oriented benchmark is not an LLM result, but the distinction is useful: a model-loaded idle server still belongs in the capacity budget. (www.spec.org) Do not subtract idle electricity from a total operating-cost estimate merely because incremental GPU efficiency looks better without it.

Translate to facility energy once

PUE relates total facility energy to IT equipment energy. (datacenters.lbl.gov) If the measurement is server or rack IT energy, a declared PUE assumption can produce a facility estimate. If the meter already records facility input, the facility-energy numerator already includes IT equipment energy and supporting overhead. (datacenters.lbl.gov) Multiplying that reading by PUE again would double-count overhead.

A fixed annual PUE is only an approximation for an individual workload change. The Green Grid notes that PUE varies with IT load. (datacenters.lbl.gov) Prefer measured facility deltas when attribution is practical, or disclose the coefficient and run a sensitivity analysis. Metering inputs and outputs is consistent with Australia's energy-department guidance on data-centre efficiency. (www.energy.gov.au)

Define goodput without misusing percentiles

SLO-qualified throughput, or goodput, counts useful completed work meeting the declared acceptance rule. For this report's protocol, a request contributes output tokens only if it completes successfully, passes the declared quality check and meets every applicable request-level deadline. Count rejected work's energy in the numerator; exclude its tokens from the accepted-output denominator.

Separately require the observation window to satisfy aggregate **p95 or p99** latency and error-rate limits. A percentile is a distribution statistic, not an individual request's attribute. Do not call a request “p99 compliant.” If a window fails the deployment SLO, mark that configuration ineligible even if many individual requests passed.

Prometheus warns that averaging quantiles across instances is statistically nonsensical. (prometheus.io) Aggregate request records or appropriate histograms instead. (prometheus.io) Save detailed responses and errors where practical; vLLM's versioned benchmark interface supports per-request information. (docs.vllm.ai) Also report accepted requests per second alongside accepted tokens per second. This prevents an output-length change from being mistaken for a capacity gain.

Constructing the Power and Cost Frontier

Compute completed-work efficiency

For each eligible run, integrate measured power over the measurement interval. With constant average power, the equivalent expression is:

```
energy_kWh = average_measured_kW * elapsed_hours
kWh_per_million_accepted_tokens =
    energy_kWh / accepted_output_tokens * 1000000
qualified_tokens_per_second =
    accepted_output_tokens / elapsed_seconds
```

These are dimensional definitions and analyst calculations, not measured results. The kWh unit follows EIA's power-times-time definition. (www.eia.gov) If the numerator is measured facility energy, report **facility kWh per million accepted output tokens**. If it is server AC energy, label it accordingly. Keep input tokens separate, preserve the requested-output distribution, and explain how truncated responses and retries are counted. Input-length metrics can help audit this workload manifest. (huggingface.co)

A cap is energy-improving for equal completed work only when its power ratio is smaller than its throughput ratio. This follows from dividing average power by useful throughput. SPECpower's documented performance-to-power metric uses the reciprocal relationship, throughput divided by average power. (www.spec.org) Neither ratio establishes monthly savings until the required capacity and actual service schedule are considered.

Round capacity in deployable units

For independently serving single-GPU workers, required GPUs equal the ceiling of target qualified throughput divided by measured qualified throughput per GPU. For tensor-parallel serving, first round full replicas:

```
required_replicas = CEILING(target_qualified_rate / rate_per_replica)
required_GPUs = GPUs_per_replica * required_replicas
```

This arithmetic assumes the tested workload scales across replicas without another bottleneck. Test that assumption under the intended routing and shared-network configuration. The original LLM study comparing tensor-parallel degrees reports nonlinear latency scaling, supporting caution about treating each GPU as an independent capacity unit. (arxiv.org) Kubernetes device-plugin resources likewise use integer quantities and cannot be overcommitted at that resource level. (kubernetes.io)

Include reserve capacity explicitly. Spare replicas needed for maintenance, availability or burst traffic belong in the deployed fleet; a ceiling based only on average demand is not a complete production plan. A fixed installed fleet has a different economic question from new procurement: a cap may avoid a power upgrade without avoiding existing hardware cost.

Build and select the frontier

Plot measured input watts against qualified throughput. Mark ineligible SLO windows and distinguish repeated measurements from the mean. A point is **dominated** if another eligible point requires no more input power and delivers no less useful throughput, with a strict improvement in at least one. Keep energy per accepted output, whole-fleet power and total cost as additional decision dimensions.

Use the following screening sequence:

- **Electrical feasibility:** stay inside the declared operational envelope.
- **Application feasibility:** meet percentile, deadline and error requirements.
- **Work efficiency:** compare kWh per accepted output at the same trace.
- **Capacity:** round complete replicas and add declared reserves.
- **Economics:** calculate electricity plus deployed capacity cost.
- **Margin:** retain headroom for the tested traffic and thermal variation.

Energy cost per accepted token equals the applicable energy price multiplied by facility kWh per million accepted tokens, divided by one million. Monthly cost should include facility energy charges, applicable demand charges, GPU or server capacity cost and other costs that change between configurations. DOE describes demand charges in terms of maximum kW over a billing period. (www.energy.gov) Its guidance also describes ratchets tied to current and previous peaks. (www.energy.gov) Lower experiment-average power does not automatically reduce that bill component.

Use the customer's actual tariff. EIA's average price methodology includes charges and does not equal a customer's per-kWh rate. (www.eia.gov) EIA also identifies locality as a source of price variation. (www.eia.gov) A national average is context, not a substitute for the incremental price used in the decision.

Production Guardrails and Training Interpretation

Keep redundancy and cap persistence explicit

A performance-efficient operating point can still be unsuitable for a required power-supply unit (PSU) redundancy state. NVIDIA's DGX B200 guide links **N+N redundancy** to system power capping. (docs.nvidia.com) The H100/H200 guide states that the relevant redundancy feature is disabled by default starting with **version 24.07.1**. (docs.nvidia.com) Verify the exact system's firmware, configuration and supported power budget rather than transferring these statements to another chassis.

Host configuration persistence also needs testing. nvidia-smi documents that persistence mode defaults to disabled after reboot; that statement does not establish whether every device's watt limit survives reboot. (docs.nvidia.com) It separately documents module power limits returning to default after driver unload. (docs.nvidia.com) Capture effective settings after each lifecycle event relevant to the deployment.

A production change record should contain:

- **Owner:** which service or operator applies the cap.
- **Identity:** physical GPU and model-replica mapping.
- **Policy:** host, firmware, BMC and scheduler settings.

- **Persistence:** observed state after reboot and driver reload.
- **Redundancy:** approved PSU state and input-power envelope.
- **Rollback:** saved settings and a verified restoration procedure.
- **Trigger:** the latency, error or thermal condition that restores baseline.
- **Evidence:** configuration, trace, power samples and acceptance report.

Track clock-event reasons alongside latency. nvidia-smi distinguishes external power-brake assertion from other clock-limiting reasons. (docs.nvidia.com) Do not interpret every frequency reduction as the intended software cap. Identify thermal constraints before treating a capped run as representative.

Integrate with deployment and scheduling

Test restoration on a canary replica before fleet rollout. Drain or route around the canary as the service design requires, observe effective limits, and replay the acceptance trace. Scale only after confirming that the deployed configuration reproduces the measured goodput and margin.

Account for scheduler behavior during rollback. Slurm documents that prolog and epilog hooks run outside the job's Linux control group, so their device view can differ from the job's view. (slurm.schedmd.com) Resolve physical identity carefully in site hooks. Kubernetes can reduce allocatable resources when a device is marked unhealthy; capacity calculations should use deployable resources rather than inventory alone. (kubernetes.io) These are operational constraints on the calculation, not evidence that either scheduler universally manages GPU watts.

Interpret training separately

Training should optimize **kWh to a target validation result or accepted checkpoint**, subject to a completion deadline and reproducibility requirements. Zeus explicitly combines energy-to-accuracy and time-to-accuracy for a user-selected validation target. (ml.energy) Reducing watts for a fixed number of steps is insufficient if those steps do not deliver the same outcome.

Record validation data, optimizer state, checkpoint criteria and any changed numerical behavior. PyTorch's checkpoint guidance includes saving optimizer state when resuming training. (docs.pytorch.org) Measure the actual completed run, including required checkpointing and evaluation, rather than reporting only its busiest kernel interval. Keep inference and training frontiers separate: their denominators, acceptance rules and capacity patterns answer different questions.

Data Analysis and Evidence

What published measurements establish

Table 2 separates measured findings from their limits. It deliberately compares evidence types rather than ranking hardware.

Source and scope	Quantitative observation	Decision limit
Dell XE9680 vendor test	Llama2-70b Datacenter 99.9 Offline; at 600 W per GPU, reported system power fell 15% and performance fell 6%. The designated 450 W configuration was a verified MLPerf Inference v4.0 submission; the remaining sweep results were internal and unverified. (infohub.delltechnologies.com) (infohub.delltechnologies.com)	Offline throughput does not establish interactive TTFT or tail-latency compliance.
NERSC Perlmutter documentation	A100 40 gigabytes (GB) allowed range: 100 to 400 W; A100 80 GB: 100 to 500 W. (docs.nersc.gov) (docs.nersc.gov)	These are platform-specific allowed ranges, not efficiency optima.
ASPLOS 2024 research	Frequency locking yielded up to 20% lower peak power with up to 7% performance loss across evaluated inference configurations. (www.microsoft.com)	Frequency locking differs from a watt cap; the maxima are not a universal paired operating point.
EIA commercial-price context	Preliminary 2025 US average commercial retail price: 13.41 cents/kWh, as reported on the fetched page. (www.eia.gov)	Average realized prices do not specify the site's tariff or marginal charge.

The evidence supports profiling and explicit boundaries. It does not support transferring a vendor's offline operating point to an interactive service or combining extrema from different research configurations. The retrieved MLCommons policy identifies an older rules version; its same-run principle is used here without describing it as a newly issued policy. (github.com)

Dell's reported power and throughput ratios imply an energy-per-work ratio of **0.85 / 0.94 = 0.9043**, or approximately **9.6% less energy per unit work**, if both ratios describe the same steady workload and measurement boundary. This is analyst arithmetic from the published percentages, not an additional Dell measurement. The reduction in watts is larger than the inferred energy saving because work completes more slowly. (infohub.delltechnologies.com)

Reproducible monthly-cost scenarios (Hypothetical Example)

All values in the following example are **synthetic assumptions**, not Dell, NERSC or GPU Smith benchmarks. Each model replica uses **4 GPUs**. The target is **15,000 accepted output tokens/s**, with all three settings assumed to pass the same SLO. The assumed operating month is **730 hours** of the stated load, with **PUE 1.20** used to estimate facility input from server AC power. There are no reserve replicas, demand charges or fixed shared infrastructure costs in this illustration. PUE's facility-to-IT boundary motivates the conversion; the chosen coefficient is hypothetical. (www.energy.gov.au)

The monetary unit is US dollars (USD). Scenario **A** assumes **USD 0.10/kWh** and **USD 300/GPU-month** capacity cost. Scenario **B** changes electricity to **USD 0.30/kWh**, retaining that capacity cost. Scenario **C** assumes **USD 0.10/kWh** and **USD 1,500/GPU-month**. These are sensitivity inputs, not quoted tariffs or hardware offers. “Capacity cost” means the allocated monthly hardware and capacity expense before the separately modeled electricity charge.

Table 3 applies replica rounding and the same formulas to each assumed operating point.

Cap W/GPU	Server kW/replica; accepted tokens/s/replica	Replicas; GPUs; facility kW	Facility kWh/million accepted tokens	Monthly A; B; C, USD
700	3.40; 4,000	4; 16; 16.32	0.2833	5,991.36; 8,374.08; 25,191.36
550	2.90; 3,900	4; 16; 13.92	0.2479	5,816.16; 7,848.48; 25,016.16
450	2.50; 3,100	5; 20; 15.00	0.2688	7,095.00; 9,285.00; 31,095.00

The lowest cap reduces each replica's instantaneous server power, but it requires another replica. Its modeled full-fleet power and cost exceed the middle setting in every scenario. The table assumes each replica sustains its measured-point power and capacity for the illustrated month; actual demand tracking, idle time, routing and nonlinear scaling require their own measurements.

The following text plots display those same synthetic coordinates, sorted by increasing cap. The cost lines share a vertical scale within their panel; all coordinates are available in the table.

```
Qualified throughput vs estimated facility input per replica
4000 |                              H (700 W, 4000)
3900 |               M (550 W, 3900)
3100 | L (450 W, 3100)
     +-------------------------------------------
       3.00          3.48           4.08   facility kW/replica

Monthly cost, USD, by electricity/capacity scenario
32000 | C: L
25000 | C:               M-------------H
 9500 | B: L
 8400 | B:                               H
 7800 | B:               M
 7100 | A: L
 6000 | A:               M-------------H
      +-------------------------------------------
        450            550           700 cap W/GPU
Point labels L/M/H denote low/middle/high cap positions.
Scenario labels A/B/C denote the three cost assumptions.
```

The plots are coordinate sketches, not fitted curves. Retain unrounded calculations in a spreadsheet and use real scatter plots for the production report.

CSV schema and spreadsheet construction

Copy this measurement schema into a comma-separated values (CSV) file and retain one row per repeat:

```
run_id,workload_id,engine_version,driver_version,firmware_version,gpu_uuid_set,gpus_per_replica,cap_requeste
d_W,cap_effective_W,system_cap_W,meter_boundary,pue_assumed,elapsed_s,energy_kWh,avg_input_kW,idle_input_kW,
```

```
offered_requests,completed_requests,accepted_requests,accepted_output_tokens,ttft_p95_ms,ttft_p99_ms,itl_p95
_ms,itl_p99_ms,error_rate,slo_window_pass,thermal_state,psu_redundancy_state
```

Keep detailed request and power files referenced by run identifier. MLCommons asks for documentation of power-management settings used to achieve measured performance and power, which supports retaining requested and effective settings in this record. (github.com)

For a spreadsheet, place **cap, server kW, accepted tokens/s, GPUs per replica, target tokens/s, PUE, hours, price/kWh and cost/GPU-month** in columns A through I. Starting on row 2, calculate:

```
J2 = CEILING(E2/C2,1)
K2 = D2*J2
L2 = B2*F2*J2
M2 = B2*F2*1000000/(C2*3600)
N2 = L2*G2*H2 + K2*I2
```

J is replicas, K is GPUs, L is estimated facility kW, M is kWh per million accepted tokens, and N is monthly cost. Copy each operating point under each scenario's H and I assumptions. Plot A against C for throughput and A against N for each cost series. Replace this constant-load model with time-weighted power states for a variable-load deployment; it must not silently become an assumed annual-load-factor calculation.

Case Studies and Real-World Examples

Dell: an efficiency result with a narrow scope

Dell reports **3.005 tokens/s per system watt** for its **450 W** Config 1 point, with the **700 W** point reaching **84.2%** of that efficiency. (infohub.delltechnologies.com) This vendor example demonstrates that the cap producing the largest efficiency ratio need not be the uncapped baseline. It does not provide a production traffic trace establishing this report's acceptance contract.

An operator can reuse the experimental question without copying the answer: does the lower cap retain enough useful throughput at the intended arrival process? Add client-side latency, error and acceptance records to the electrical sweep. Then calculate complete replicas and cost. If the application needs more capacity at the lower setting, the ratio observed on one server is only an intermediate result.

Treat changes to system performance profiles as distinct variables. An experiment that changes both a cap and another performance setting cannot attribute all differences to the cap alone. Preserve the baseline profile when isolating the electrical intervention, and run a separate profile comparison when that is itself the decision.

NERSC: a job-level operational example

NERSC documents a **200 W** job request through the following site-specific directive. (docs.nersc.gov)

```
#SBATCH --gpu-power=200
```

The documented scope lasts throughout the job and its job steps. (docs.nersc.gov) Requests outside an allowable range are adjusted to the nearest valid value. (docs.nersc.gov) NERSC also exposes the cap in the **AdminComment** field of job accounting. (docs.nersc.gov) These details demonstrate why a job file alone is insufficient evidence of the actual operating point.

A private cluster can apply the same governance principle: identify the supported control, log its effective state, attach that state to accounting, and restore the intended baseline after the experiment. The directive is a NERSC capability, not a claim that every Slurm installation supports it. A site without the plugin needs its own documented integration and permission model.

For batch inference, include completion deadline and accepted-output counts with job-level kWh. For training, include validation outcome and checkpoint acceptance instead. The scheduler's accounting record makes the experiment traceable; it does not decide which objective is economically appropriate.

Implications and Future Directions

The useful production result is a **validated operating policy**, not an isolated watt value. It identifies the workload envelope, effective cap sources, meter boundary, acceptance rules, tested traffic range and rollback conditions. That package can be rerun when the model, firmware, driver or serving engine changes. Zeus's power-limit optimizer makes selection criteria explicit, which is the appropriate principle for automation as well as manual sweeps. (ml.energy)

Use a decision record to communicate uncertainty:

- **Selected point:** why it survives electrical and application checks.
- **Runner-up:** which assumption would change the choice.
- **Sensitivity:** energy price, capacity cost and required reserve replicas.
- **Boundary:** measured facility input or estimated facility energy.
- **Evidence gap:** untested bursts, transitions or multi-node scaling.
- **Refresh trigger:** the configuration change that requires another sweep.

Automation should first reproduce a fixed-policy test. Later, a workload-aware controller can be evaluated against the same acceptance contract, with its own transition behavior and control overhead measured. Do not assume that a setting selected under steady offered load remains suitable during burst traffic. Ray Serve's guidance connects batching wait time to end-to-end latency SLOs, illustrating how a non-electrical control can consume the same latency budget. (docs.ray.io)

For facilities constrained by input power, compare the useful output achievable by the whole admissible fleet. For procurement, compare complete capacity cost. For an installed cluster, disclose whether capacity cost is avoidable, allocated or sunk. Those distinctions can change the economic answer without changing a single benchmark measurement.

GPU Smith's first-party method describes acceptance testing against written criteria. (gpusmith.com) Applied to this topic, the relevant perspective is to specify the criterion before recommending a cap and preserve the measurements that support it. That does not turn consultancy positioning into performance evidence.

Publishing the manifest, raw records and calculation assumptions makes later reassessment possible. The strongest reusable result is the experiment design and its observed frontier, with the conditions under which the selected point remains valid.

Conclusion

The optimal GPU power limit for inference is the eligible point that best serves a declared objective under the tested workload. The objective may be facility-power feasibility, energy per accepted output, total monthly cost or a completion deadline. Each produces a different comparison unless the same operating point happens to satisfy them all.

A defensible decision starts with an uncapped experimental baseline that records the platform's existing limits. It then sweeps supported caps while holding the workload and serving configuration steady. Power and accepted output belong to the same interval; GPU telemetry, server input and facility energy retain their respective labels. Latency percentiles determine service eligibility, while request-level acceptance determines useful-output counts.

After measurement, round capacity in whole deployable replicas. Include required reserves and the tariff components that actually change. The hypothetical scenarios show why lowering each GPU's cap can increase fleet power and monthly cost once another replica is required. That arithmetic is a reason to measure capacity, not a forecast for any named system.

Deploy only with verified effective settings, redundancy state, lifecycle behavior and rollback. Retain the baseline when no tested cap improves the relevant frontier with adequate margin. Keep training's validation-based outcome separate from inference's accepted-output outcome. The decision is complete when another operator can reproduce the configuration, measurements and cost calculation and understand why the selected point satisfies the stated constraints.

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/dgx/dgxb200-user-guide/power-capping.html
2. https://ml.energy/zeus/optimize/power_limit_optimizer/
3. https://github.com/mlcommons/inference_policies/blob/master/power_measurement.adoc
4. https://docs.vllm.ai/en/v0.12.0/cli/bench/serve/
5. https://infohub.delltechnologies.com/en-us/p/enabling-efficient-llm-inferencing-power-capping-strategies-for-poweredge-xe9680/
6. https://www.energy.gov.au/business/equipment-guides/data-centres
7. https://www.energy.gov/cmei/femp/evaluating-your-utility-rate-options
8. https://ml.energy/blog/energy/research/ml-energy-performance-and-accuracy/
9. https://gpusmith.com/articles/en/data-center-electrical-design-gpu-clusters
10. https://www.eia.gov/energyexplained/electricity/measuring-electricity.php
11. https://gpusmith.com/

12. https://arxiv.org/html/2403.20306v1
13. https://docs.nvidia.com/deploy/nvidia-smi/index.html
14. https://docs.nersc.gov/jobs/power-capping/
15. https://slurm.schedmd.com/sbatch.html
16. https://rocmdocs.amd.com/projects/amdsmi/en/latest/how-to/amdsmi-cli-tool.html
17. https://kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/
18. https://cdn.kernel.org/doc/html/latest/power/powercap/powercap.html
19. https://docs.vllm.ai/en/latest/benchmarking/cli/
20. https://huggingface.co/docs/text-generation-inference/reference/metrics
21. https://docs.pytorch.org/tutorials/beginner/saving_loading_models
22. https://docs.ray.io/en/latest/serve/advanced-guides/dyn-req-batch.html
23. https://ml.energy/zeus/measure/
24. https://www.spec.org/power/docs/1.12/SPECpower_ssj2008-Result_File_Fields.html
25. https://datacenters.lbl.gov/sites/default/files/WP49-PUE%20A%20Comprehensive%20Examination%20of%20the%20Metric_v6.pdf
26. https://prometheus.io/docs/practices/histograms/
27. https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/device-plugins/
28. https://gpusmith.com/articles/en/gpu-cluster-reliability-budget
29. https://www.eia.gov/electricity/annual/pdf/tech_notes.pdf
30. https://www.eia.gov/energyexplained/electricity/prices-and-factors-affecting-prices.php
31. https://docs.nvidia.com/dgx/dgxh100-user-guide/power-capping.html
32. https://slurm.schedmd.com/gres.html
33. https://www.microsoft.com/en-us/research/wp-content/uploads/2024/03/GPU_Power_ASPLOS_24.pdf

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.