GPU SMITH
COMPUTE ENGINEERING

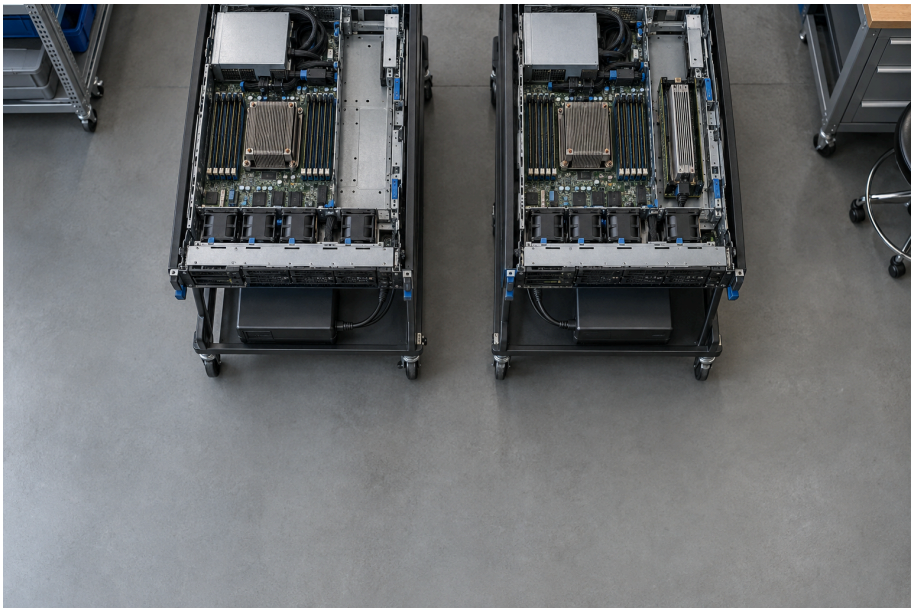
GPU SMITH / RESEARCH & PRACTICAL GUIDANCE

CPU vs GPU LLM Inference: Low-Concurrency Cost Calculator

> 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 · 2026-10-08 · cpu vs gpu llm inference · llm inference cost · cpu inference · gpu inference · low concurrency · llama.cpp · private ai · inference benchmarking · total cost of ownership

Original article: https://gpusmith.com/articles/en/cpu-vs-gpu-llm-inference-cost-calculator



In this article

1. Executive Summary
2. Introduction and Background
3. CPU Inference
4. GPU Inference
5. Feature Comparison
6. Comparable Test Contract
7. Performance and Benchmarks
8. Trace-Driven Cost Calculator
9. Data Analysis and Evidence
10. Implications and Future Directions
11. Frequently Asked Questions (FAQs)
12. Conclusion

Executive Summary

CPU vs GPU LLM inference is a workload-and-service-level decision. At low or intermittent concurrency, an available CPU server deserves the first test when its memory, instruction support and latency meet the application's requirements. A GPU becomes the stronger economic candidate when it delivers enough qualified capacity to replace multiple CPU nodes, or when the CPU cannot satisfy the response-time objective. Neither conclusion follows from peak floating-point operations per second. Current **vLLM** documentation explicitly supports CPU inference, and **llama.cpp** provides both x86 and CUDA backends. (docs.vllm.ai) (github.com) (github.com)

This report specifies a reproducible comparison using the official **Qwen2.5-3B-Instruct Q4_K_M GGUF** file, listed at **2.1 GB**, and one pinned llama.cpp revision. (huggingface.co) Its purpose is to make the experiment reproducible, rather than endorse that model for every enterprise task. The artifact's research license requires requesting a license for commercial use. (huggingface.co) An enterprise must resolve that prerequisite or replace the artifact in both test arms and rerun the comparison. The model card specifies **32,768 tokens** of context and **8,192 tokens** of generation, but the proposed test deliberately uses shorter, fixed inputs. (huggingface.co)

The main output is **qualified capacity**: the highest tested sustained arrival rate at which a declared time-to-first-token, inter-token-latency and error-rate service-level objective passes. The report uses **p95**, meaning the 95th percentile, and preserves the workload trace, context distribution and deployment configuration. MLCommons' public Endpoints overview also describes p95; newer submitter documentation describes a p90 development-rule change. These percentiles must remain distinct. Its throughput-per-kilowatt normalization uses provisioned system power, which must also remain distinct from measured energy for ownership costs. (mlcommons.org) (docs.mlcommons.org) (docs.mlcommons.org)

The calculator combines amortized hardware, measured whole-fleet wall energy, facility overhead and explicit operating assumptions. GPU component power is insufficient: **NVIDIA L4** is documented at **72 W** (www.nvidia.com), while its host, fans and memory remain inside the deployment boundary. Electricity

sensitivity can use EIA's preliminary **July 2026** sector averages, **14.53 cents/kWh commercial** and **9.77 cents/kWh industrial**, only as reference points. (www.eia.gov) Procurement decisions require local tariffs and dated complete-system quotes. (www.intel.com) The result is a staircase of whole-server costs, sometimes with several switches rather than one universal break-even rate. No matched benchmark or procurement price is invented here.

Introduction and Background

A private endpoint can be inexpensive per generated token at sustained load yet expensive over a month of intermittent use. Conversely, a CPU endpoint can have acceptable single-request behavior but accumulate a queue during bursts. The relevant question is whether each proposed fleet can serve the same arrivals, produce acceptable task outputs and meet the same latency commitment.

A **large language model (LLM)** processes input during **prefill** and generates subsequent output during **decode**. The Sarathi-Serve research separates these phases and explains how batching interleaves them. This distinction matters because an interactive chat request with a long prompt is a different workload from a short classification request, even when both use the same weights. (arxiv.org) (arxiv.org)

The intended deployment range is small private models, from roughly **1B to 14B parameters**, serving chat, classification or agent-support tasks. That range is the report's scope, not a claim that every model within it performs well on CPUs. Embedding and reranking services require separate experiments: llama-server documents dedicated embedding models and a reranking mode, so a chat model's output-token throughput cannot establish either workload's economics. (github.com) (github.com)

GPU Smith's first-party engineering description derives topology and serving-stack choices from workload models and throughput targets. That provides an appropriate advisory perspective for this comparison: characterize the demand before choosing the processor. The consultancy is an adjacent engineering advisor, rather than a CPU or GPU option in the comparison table. (gpusmith.com)

Red Hat's workload-placement discussion proposes smaller models for classification, extraction and routing. This is a **vendor observation**, useful for selecting test candidates, but it does not prove a private-server cost crossover. The analysis below uses primary documentation to establish compatibility and measurement boundaries, then supplies a calculation that depends on the buyer's own traces, power readings and quotes. (www.redhat.com)

CPU Inference

Capabilities

CPU inference is a documented serving path. vLLM's x86 installation guidance lists **FP32, FP16 and BF16** data types, instruction requirements and architecture-dependent quantization support. Those facts establish software availability, not a guaranteed response time for the selected model. llama.cpp independently documents x86 instruction support including AVX, AVX2, AVX512 and AMX. (docs.vllm.ai) (docs.vllm.ai) (github.com)

Instruction set architecture (ISA) describes the instructions available to the runtime. **Non-uniform memory access (NUMA)** describes differing memory-access locality within a system. AMD's EPYC architecture documentation explains the connection between memory latency and memory-controller connectivity. vLLM recommends keeping the cores for a serving rank within the same NUMA node. Record the actual firmware configuration and operating-system topology rather than infer them from the processor name. (www.amd.com) (docs.vllm.ai)

Memory population is part of the configuration. The EPYC 9004 family supports **12 memory channels** (www.amd.com); the Lenovo SR635 V3 documents **12 channels with one DIMM per channel**. (lenovopress.lenovo.com) Intel's Xeon Gold 5418Y lists **eight channels**, **24 cores** and **185 W processor TDP**. (www.intel.com) (www.intel.com) These are hardware boundaries, not comparative inference results. A partially populated server must be benchmarked as purchased or reused.

Adoption

The sources establish supported implementation routes, not enterprise adoption percentages. The appropriate adoption question is therefore local: can the existing server estate provide a dedicated, reproducible serving allocation? PyTorch's tuning guide recommends socket-local inference instances and documents CPU-and-memory binding through numactl. This supports a concrete deployment experiment without implying that every multi-socket server should use identical settings. (docs.pytorch.org) (docs.pytorch.org)

A CPU-first trial is useful when the team has available RAM, available cores and a response-time budget that the measured configuration satisfies. Red Hat identifies hybrid workloads and off-peak processing as CPU-placement possibilities, but that remains an attributed placement view rather than a cost result. (www.redhat.com)

Strengths and Limitations

Reuse can lower incremental capital cost, but it does not make capacity free. Charge the workload for displaced applications, additional memory, support, operating effort and any increase in wall energy. Maintain both an incremental view and a fully allocated view so the accounting choice remains visible.

CPU performance also depends on runtime settings. vLLM describes the throughput-versus-latency batching tradeoff, while llama.cpp distinguishes BLAS effects on prompt processing from generation. Optimize the CPU path deliberately, then freeze it before the sweep. A low CPU utilization reading alone does not establish spare qualified serving capacity. (docs.vllm.ai) (github.com)

GPU Inference

Capabilities

llama.cpp documents CUDA kernels for NVIDIA GPUs and also supports hybrid CPU-plus-GPU inference. For the primary comparison, the GPU arm should keep the selected model's intended offloaded layers resident and record startup placement. A hybrid run remains a separately named candidate because its host-memory behavior and capacity differ from a fully offloaded configuration. (github.com) (github.com)

The **L4** is a concrete accelerator candidate for an input sheet, rather than a procurement recommendation. NVIDIA specifies a **72 W** low-power envelope (www.nvidia.com); Lenovo identifies **24 GB**, PCIe Gen4 and passive cooling. (lenovopress.lenovo.com) Neither specification establishes available memory after runtime allocation or the number of requests that meet the chosen objective. The selected artifact's file size is likewise only one part of memory demand. (huggingface.co)

The host configuration must be verified. Lenovo documents that L4 needs no auxiliary power cable, but its server rules specify risers for multi-GPU configurations and performance fans for GPU installations. A compatible slot and sufficient accelerator memory do not complete the bill of materials. (lenovopress.lenovo.com) (lenovopress.lenovo.com) (lenovopress.lenovo.com)

Adoption

The evidence supports deployment mechanisms, not a market-share claim. Kubernetes documents GPU drivers and device plugins as prerequisites and exposes vendor-specific schedulable resources after installation. These mechanisms establish resource placement; they do not establish actual accelerator utilization or economical sharing. (kubernetes.io) (kubernetes.io)

A single GPU service can therefore be tested as an operational candidate with its host, software and standby policy included. If several small endpoints are consolidated, benchmark their combined arrival trace. The savings from consolidation remain conditional on those endpoints meeting their objectives simultaneously.

Strengths and Limitations

The operational comparison should preserve the same availability commitment. A single GPU node and several CPU nodes create different consolidation and maintenance patterns, but node count alone does not establish redundancy. Decide whether the service must survive a node loss, model reload or scheduled patch, then include the necessary standby capacity in both fleets.

The GPU case is strongest when a measured fleet supplies required latency or avoids additional CPU nodes. Its low-demand limitation is **stranded capacity**: acquisition and support continue while little accepted work arrives. Power-management settings and shutdown policies change that cost only when their effects are measured.

Cooling and support can alter the quote. Dell's R660 thermal guidance requires a CPU shroud for air-cooled configurations and documents a **270 W CPU-TDP restriction** in specified three-L4 configurations. (www.dell.com) (www.dell.com) Lenovo states that an installed L4 assumes the server warranty. (lenovopress.lenovo.com) These examples show why an accelerator-only price cannot represent a complete, serviceable inference node.

Include software deliberately. NVIDIA describes AI Enterprise as a license addition for L4; record whether the proposed deployment uses it and obtain the applicable terms. Do not assume that every open-source serving configuration carries that license cost. Facility cooling also extends beyond the server, as DOE's cooling-system description illustrates. (www.nvidia.com) (www.energy.gov)

Feature Comparison

The comparison should separate documented capabilities from acceptance results. *Table 1* summarizes the procurement questions; “measure” means that source documentation does not settle the buyer’s result.

Criterion	CPU candidate	GPU candidate
Serving route	Documented x86 backends; verify actual ISA and build. (github.com)	Documented CUDA backend; verify actual layer placement. (github.com)
Memory boundary	System RAM, NUMA locality and channel population. (www.amd.com)	Accelerator memory plus host RAM and runtime allocations. (lenovopress.lenovo.com)
Scheduling boundary	CPU requests, limits and reserved system resources. (kubernetes.io)	GPU device resources, drivers and plugin configuration. (kubernetes.io)
Power boundary	Whole-server alternating-current input, including idle. (www.spec.org)	Whole host plus accelerator at that same measurement boundary. (spec.org)
Financial boundary	Reuse opportunity cost or a dated complete-node quote.	Complete host, accelerator, integration, support and optional licenses. (www.nvidia.com)
Acceptance result	Measure qualified arrivals and accepted output.	Measure the identical trace and acceptance criteria. (mlcommons.org)

The table does not rank architectures by specification. It identifies where a comparison can accidentally charge unlike boundaries. A reused CPU server and a newly purchased GPU server can be valid alternatives, but their acquisition bases and service obligations must remain explicit.

Concurrency means requests simultaneously in flight, rather than registered users or monthly requests. MLCommons’ Endpoints overview makes that distinction. A department with intermittent requests can still produce a burst with substantial in-flight demand; an average monthly request rate cannot capture that behavior. (mlcommons.org)

The choice also depends on the workload’s output definition. For a generative endpoint, count accepted output tokens and separately record input tokens. For embedding, count accepted items or input tokens under a fixed dimension and quality criterion. For reranking, count accepted query-document work under a fixed candidate-set distribution. Treat these as different acceptance contracts; a single monetary ratio cannot establish equivalent task value.

There may be **multiple crossovers**. Immediately after an added CPU node, the GPU fleet may become cheaper; immediately after an added GPU node, the ranking can change again. Find every sign change in total cost over tested arrival rates, rather than force a continuous straight-line intersection.

- **CPU first:** available capacity, acceptable observed latency and a credible reuse-cost basis.
- **GPU first:** the CPU trial misses the required objective, or measured GPU capacity reduces fleet cost.
- **Consolidation trial:** several endpoints have complementary demand and can share a tested failure domain.
- **Neither accepted:** model quality, licensing, capacity or operational requirements remain unresolved.

These are experiment-selection rules. The final decision must come from compliant measured fleets, rather than the labels in the table.

Comparable Test Contract

Model, Precision and Engine

Use one exact downloadable artifact in both arms: **qwen2.5-3b-instruct-q4_k_m.gguf**, from Qwen's official repository. Its published SHA256 is **626b4a6678b86442240e33df819e00132d3ba7dddfe1cdc4fbb18e0a9615c62d**. (huggingface.co) Verify the complete hash after download and retain it in the run manifest. The model card lists Q4_K_M among its supplied quantizations. (huggingface.co)

This is a reproducibility fixture. Its official license requires a commercial-license request, so procurement approval must precede commercial deployment. If an approved model replaces it, replace the file in both arms and repeat every capacity measurement. Equal file bytes control the weight artifact; they do not guarantee identical floating-point reductions across backends. (huggingface.co)

Pin **llama.cpp b11505**, identified on the fetched release page as a prerelease at short commit **ff5888f**, solely as this report's reproducibility revision. (github.com) Resolve and record the full commit before building. Archive compiler, CUDA toolkit, driver, operating-system image and container digest. No container digest or completed benchmark is asserted here. llama-server's version option reports build information. (github.com)

The commands below define the proposed reproducibility recipe. They were not executed on benchmark hardware. The thread, slot and context settings are explicit experiment inputs. CUDA builds also include the CPU backend; separate CPU-only builds help audit the CPU arm. (github.com)

```
git clone https://github.com/ggml-org/llama.cpp
cd llama.cpp
git checkout b11505
git rev-parse HEAD
cmake -B build-cpu -DCMAKE_BUILD_TYPE=Release -DGGML_CUDA=OFF
cmake --build build-cpu --config Release -j
cmake -B build-gpu -DCMAKE_BUILD_TYPE=Release -DGGML_CUDA=ON
cmake --build build-gpu --config Release -j
hf download Qwen/Qwen2.5-3B-Instruct-GGUF   qwen2.5-3b-instruct-q4_k_m.gguf --local-dir models
sha256sum models/qwen2.5-3b-instruct-q4_k_m.gguf
build-cpu/bin/llama-server --version
build-gpu/bin/llama-server --version
```

Launch the CPU arm with zero GPU layers and the GPU arm with sufficient GPU layers to offload the intended model. Set the same context, server-slot count and prompt template, then verify logs. Record generation threads separately from batch-processing threads. These are documented independent controls. (github.com) (github.com)

Workload and Hardware Manifest

For an **initial generative trial (Hypothetical Example)**, the following numbers are **declared test assumptions**, not measured capacities: a **4,096-token** serving context, **512-token** prompts, **128-token** requested outputs and client-concurrency sweeps of **1, 2, 4 and 8**. Tokenize the prompts with the selected model, preserve their exact bytes and report actual output lengths. Repeat with the production distribution before buying capacity.

- **Artifact:** file name, full hash, tokenizer and chat-template revision.
- **Precision:** weight quantization, cache types and any backend conversions.
- **Runtime:** full commit, executable version, compiler and dependency versions.
- **Processor:** SKU, ISA flags, firmware, socket and NUMA layout.
- **Memory:** DIMM population, configured speed, locality and reserved RAM.
- **Accelerator:** SKU, driver, placement, offloaded layers and memory allocation.
- **Trace:** arrivals, input lengths, output limits, cache policy and cancellations.
- **Power:** meter identity, sampling interval and included equipment.

The current vLLM CPU quantization documentation lists architecture-specific configurations; it does not establish this exact GGUF CPU path. Use its CPU benchmark-suite guidance for a separately contracted vLLM experiment, rather than silently substitute engines or precision in the primary comparison. (docs.vllm.ai) (docs.vllm.ai)

Performance and Benchmarks

Measure Qualified Capacity

Declare the **service-level objective (SLO)** before running the sweep. **Interactive-chat SLO inputs (Hypothetical Example)** are **p95 TTFT at most 1,000 ms**, **p95 ITL at most 100 ms** and **errors at most 1%**. These are example acceptance inputs, not industry guarantees. **Time to first token (TTFT)** measures first response delay; **inter-token latency (ITL)** measures spacing between generated tokens. Streaming must expose those events. (mlcommons.org) (developer.nvidia.com) (developer.nvidia.com)

Define percentiles explicitly. This report's calculator uses the nearest-rank percentile of successful requests, with errors assessed separately over all scheduled arrivals. It uses each request's mean observed inter-token gap as its ITL value. Teams requiring a percentile across every token gap should replace that aggregation consistently and preserve the raw gaps. Never compare these two definitions as if they were equivalent.

Start with a warm single-request run, then increase **offered arrival rate** independently of server speed. Preserve queue delay in the client-visible measurement. A closed-loop client that waits for one response before issuing another can hide queue growth. AIPerf documents configurable constant, Poisson and gamma arrival patterns; production trace replay remains preferable when available. (developer.nvidia.com)

- **Warmup:** distinguish process-ready, model-ready and first accepted response.
- **Arrival sweep:** preserve scheduled arrivals, including timeouts and dropped requests.
- **Latency:** retain per-request TTFT, ITL and completion timestamps.

- **Errors:** count transport, timeout, malformed-output and application-quality failures.
- **Output:** retain actual generated lengths and accepted-token totals.
- **Energy:** integrate whole-fleet power over the complete trace, including gaps.
- **Repetition:** repeat boundary points and report run-to-run variation.
- **Release record:** publish manifests, raw traces and percentile summaries.

llama-server performs an empty warmup by default. That behavior is not a cold-start timing measurement. Test restart and scale-up separately with the intended image, model storage and placement policy, and include startup energy and cost in monthly assumptions when relevant. (github.com)

Keep Published Evidence Comparable

MLPerf Inference **v6.0** results were released on **April 1, 2026**, with **five of eleven** datacenter tests new or updated. (mlcommons.org) That release establishes benchmark scope, not the selected small-model crossover. Restrict any external comparison to matching model, scenario, rules and measurement boundaries. Endpoints requires a common model, endpoint and software stack within a submitted curve. (mlcommons.org)

As fetched for this report, the public Endpoints overview describes p95. Submitter documentation instead describes p90 for v1.0 and says its verification was against development rules. Preserve that discrepancy and pin the methodology version; the article's p95 acceptance rule is an independent declared choice. (mlcommons.org) (docs.mlcommons.org)

Do not average node percentiles to obtain a fleet percentile. Prometheus explicitly warns that averaging quantiles is statistically nonsensical and explains histogram aggregation. Use aggregated compatible histograms or the original client samples. Report uncertainty near the fleet-sizing boundary; a point estimate that barely passes should not receive assumed growth headroom. (prometheus.io) (prometheus.io)

Trace-Driven Cost Calculator

The following standard-library Python calculator consumes client request records and measured energy for each tested whole-node fleet. It compares only candidates replaying the same trace identifier and rate. It does not extrapolate an untested capacity point or create missing prices.

```
import json, math, sys

def p95(values):
    if not values:
        return math.inf
    return sorted(values)[math.ceil(0.95 * len(values)) - 1]

def evaluate(run, config):
    nodes = run["nodes"]
    if not isinstance(nodes, int) or nodes < 1:
        raise ValueError("nodes must be a positive integer")
    seconds = run["trace_seconds"]
    if seconds <= 0 or run["amortization_months"] <= 0:
        raise ValueError("positive durations are required")
```

```
    requests = run["requests"]  # every scheduled arrival, including errors
    good = [r for r in requests if not r["error"] and r["quality_ok"]]
    errors = (len(requests) - len(good)) / len(requests) if requests else 1
    slo = config["slo"]
    compliant = (
        len(requests) >= config["minimum_requests"]
        and errors <= slo["error_fraction"]
        and p95([r["ttft_ms"] for r in good]) <= slo["ttft_ms"]
        and p95([r["itl_ms"] for r in good]) <= slo["itl_ms"]
    )
    accepted = sum(
        r["output_tokens"] for r in good
        if r["ttft_ms"] <= slo["ttft_ms"]
        and r["itl_ms"] <= slo["itl_ms"]
        and r["quality_ok"]
    )
    scale = config["monthly_hours"] * 3600 / seconds
    fixed = nodes * (
        (run["acquisition_basis"] - run["residual_value"])
        / run["amortization_months"]
        + run["maintenance_monthly"] + run["software_monthly"]
        + run["labor_monthly"]
    ) + run["shared_monthly"]
    energy = run["fleet_it_kwh"] * run["facility_multiplier"] * scale
    cost = fixed + energy * config["tariff_per_kwh"]
    cost += run["monthly_startup_cost"]
    tokens = accepted * scale if compliant else 0
    return {
        "candidate": run["candidate"], "nodes": nodes,
        "trace_id": run["trace_id"], "rate_rps": run["rate_rps"],
        "compliant": compliant, "monthly_cost": cost,
        "accepted_output_tokens": tokens,
        "cost_per_million": cost * 1_000_000 / tokens if tokens else None
    }

if __name__ == "__main__":
    config = json.load(sys.stdin)
    results = [evaluate(r, config) for r in config["runs"]]
    groups = {}
    for r in results:
        key = (r["trace_id"], r["rate_rps"])
        if r["compliant"] and r["accepted_output_tokens"] > 0:
            groups.setdefault(key, []).append(r)
    winners = [min(rows, key=lambda r: r["monthly_cost"])
               for rows in groups.values()]
    print(json.dumps({"results": results, "winners": winners}, indent=2))
```

Save the code as calculator.py and run it with a JSON input redirected to standard input. The top-level object requires **monthly_hours**, **tariff_per_kwh**, **minimum_requests**, **slo** and **runs**. The SLO object supplies **ttft_ms**, **itl_ms** and **error_fraction**; runs is a list of measured fleet records. Set the minimum request count before testing, and report statistical uncertainty separately.

All cost fields are required user inputs in one currency. **acquisition_basis** is a dated quote for new hardware or an explicitly assigned reuse value. **fleet_it_kwh** includes every charged node during the entire trace, including warm standby. **facility_multiplier** is one when the energy reading already includes allocated facility overhead. **monthly_startup_cost** includes restart energy and any associated monthly cost, without duplication.

Set **quality_ok** false for application-quality rejection; the calculator includes those requests in the error fraction. Count only successfully completed, individually latency-compliant output in a passing run. The denominator excludes prompt tokens and rejected output; the numerator still includes the compute and power spent on them. A zero denominator produces no finite token-cost result.

The monthly extrapolation assumes the complete trace represents the month's activity and idle distribution. If it covers only busy hours, extend the representative interval to include inactive periods before running the comparison. Set **trace_seconds** to the full duration of that interval and **fleet_it_kwh** to the total fleet energy over the same interval, including busy and inactive periods. Do not add an already monthly-scaled idle-energy total to **fleet_it_kwh**; supply energy for the representative interval and let the calculator scale it once to **monthly_hours**. Trace identity should include prompt bytes, arrival schedule and generation settings, not merely a descriptive name.

Data Analysis and Evidence

Hardware and Cost Input Sheet

Table 2 specifies the inputs needed before the calculator can produce a procurement result. Blank acquisition fields remain unknown; they are never filled with estimated street prices.

Input	Reused CPU	New CPU	GPU fleet
Acquisition basis per node	User-entered reuse or opportunity value, plus upgrades	Dated complete-system quote	Dated host, accelerator and integration quote
Residual value and months	Explicit remaining life and residual assumption	Explicit amortization life and residual assumption	Same accounting policy, stated separately
Monthly operating allocation	Maintenance, labor, software and facilities	Same categories	Same categories, including any selected licenses
Energy over complete trace	Measured fleet AC-input kWh	Measured fleet AC-input kWh	Measured host-plus-GPU fleet AC-input kWh
Facility multiplier	Measured or allocated facility-to-IT energy factor	Same boundary	Same boundary; avoid double counting (www.energy.gov)
Qualified capacity	Sustained arrivals meeting the declared SLO	Same trace and SLO	Same trace and SLO
Tariff and fixed charges	Local time-weighted tariff and allocated fixed charges	Same site assumptions	Same site assumptions

Interpret the sheet as **three separate alternatives**, even when the reused and newly purchased CPU configurations have identical measured performance. Intel states that recommended customer pricing is guidance rather than a formal offer. ENERGY STAR's general three-to-five-year server replacement guidance can inform sensitivity analysis, but does not prescribe the amortization life of this deployment. (www.intel.com) (www.energystar.gov)

Whole-Server Cost and Crossover

Let q_j be the measured compliant requests per second of one node for the fixed trace family. Let h_j be the fraction of qualified capacity permitted after reserving growth headroom. For peak arrival rate lambda, the initial node count is:

```
n_j = ceil(lambda / (h_j * q_j))
monthly_fixed_j =
    n_j * ((acquisition_basis_j - residual_j) / amortization_months_j
           + maintenance_j + software_j + labor_j)
    + shared_monthly_j
monthly_cost_j =
    monthly_fixed_j
    + measured_IT_kWh_j * facility_multiplier_j
      * (monthly_hours / trace_hours) * tariff
    + monthly_startup_cost_j
cost_per_million_accepted_output_tokens_j =
    monthly_cost_j / (accepted_output_tokens_month_j / 1,000,000)
```

These are this report's accounting definitions. Revalidate the selected fleet count with the routed trace, because multiplying isolated node capacity does not establish fleet latency. Add hot spares to the purchased count and energy boundary. Do not reduce required burst capacity by the average duty cycle.

Average wall energy includes idle. SPECpower's measurement rules place the analyzer between AC supply and the tested system; its active-idle definition retains workload readiness. These boundary definitions are useful, but its Java workload consumption is not an LLM power measurement. DOE defines power usage effectiveness (PUE) from annual facility and IT energy. A site-average PUE is an allocation assumption when applied to a short experiment, not a measured marginal cooling response. (spec.org) (spec.org) (www.energy.gov)

Sensitivity Without Invented Prices

- **Duty cycle:** replay identical bursts separated by longer idle intervals; keep burst capacity unchanged.
- **Electricity price:** substitute the local tariff, then explore the EIA reference range separately.
- **Reuse value:** vary opportunity cost and upgrades, rather than treat existing hardware as universally free.
- **Growth headroom:** reduce h_j and recompute integer fleet counts, then validate the resulting fleet.
- **Amortization:** vary remaining service life and residual assumptions together.
- **Standby policy:** measure always-ready and restart-based traces separately, including startup delay.
- **Quality:** repeat the same task acceptance evaluation after model or precision changes.

The worksheet identifies inputs that change the ranking. Unmatched public throughput cannot establish the buyer's break-even rate.

Implications and Future Directions

Kubernetes enforces CPU limits through throttling and schedules against requests rather than current measured utilization. Its allocatable node resources are reduced by system-daemon needs. Accordingly, a CPU benchmark on an unconstrained host does not establish performance for a tightly limited production container. Record the deployed limits and reservations in the manifest. (kubernetes.io) (kubernetes.io) (kubernetes.io)

GPU scheduling has a different allocation boundary: documented GPU requests and limits must match when both are specified. Treat device allocation, driver compatibility and sharing policy as deployment inputs. A booked device is not evidence that its memory, compute or power is fully used. (kubernetes.io) (kubernetes.io)

- **Scheduling:** repeat acceptance under the intended shared-load and placement policy.
- **Patching:** archive dependencies, update procedures and a tested rollback configuration.
- **Failure domain:** charge the standby capacity required by the availability commitment.

Table 3 turns the measured comparison into an operational decision and a retest trigger.

Observed condition	Decision supported by the experiment	Retest trigger
Reused CPU meets SLO and has lower complete cost	Retain the measured CPU allocation	Competing workload, memory population or tariff changes
CPU misses TTFT or ITL even before queue growth	Test GPU or a separately approved smaller model	Model, context distribution or runtime changes
GPU replaces enough CPU nodes at compliant demand	Select the lower-cost validated GPU fleet	Added endpoints, headroom or standby changes
Sparse demand makes restart policy attractive	Compare complete restart-and-idle traces	Cold-start latency or request distribution changes
Neither fleet passes the application acceptance test	Revise the model or service contract and rerun	Quality threshold or task distribution changes

The table requires observed evidence for the first two columns. Retest triggers describe changes capable of invalidating that evidence; they are not automatic reasons to buy a different architecture.

ENERGY STAR advises keeping server power-management settings engaged. Preserve those settings during acceptance, and measure any alternative setting under the same latency objective. A power-saving mode that changes response behavior belongs in a new candidate rather than an unrecorded production adjustment. (www.energystar.gov)

GPU Smith's first-party assessment description includes workload modelling and total cost of ownership. The useful independent-advisor role is to reconcile the manifest, dated bill of materials and acceptance record. Its own service description is not evidence that either CPU or GPU wins the experiment. (gpusmith.com)

Retain an offline registry of approved artifacts, build dependencies and rollback images where disconnected operation is required. Publish the exact revision, commands, raw percentile summary and energy boundary with the internal decision. Retest when a relevant input changes; archive what the decision covered.

Frequently Asked Questions (FAQs)

When should a team use CPU for LLM inference?

Test CPU first when existing resources and the application's latency budget make it a credible candidate. Supported software paths exist, but hardware-specific throughput and response time still require measurement. Keep NUMA locality and instruction support in the test manifest. (docs.vllm.ai) (docs.vllm.ai)

What is the CPU versus GPU inference break-even rate?

It is the tested arrival rate at which complete compliant fleet costs become equal or change ordering. Integer node counts can create several switches. Without local capacity, energy and acquisition inputs, this report establishes no numerical crossover.

Is low concurrency enough to justify CPU?

No. Concurrency counts simultaneous in-flight requests, while the workload also includes prompt length, generation length and queueing. MLCommons explicitly separates concurrency from the number of users. Replay the burst trace even when the monthly average looks small. (mlcommons.org)

Which electricity tariff should the calculator use?

EIA's preliminary **July 2026** commercial and industrial averages were released **September 24, 2026**. (www.eia.gov) (www.eia.gov) Use the cited **14.53** and **9.77 cents/kWh** as tariff sensitivity references only. (www.eia.gov) Local demand charges, time-of-use pricing and contracted rates belong in the input sheet; the national figures do not replace them.

What belongs in self-hosted inference total cost of ownership?

Include assigned acquisition cost, upgrades, residual value, amortization, whole-fleet energy, facility overhead, maintenance, selected licenses and operating allocations. Use the same service and availability boundary for every candidate.

Can embedding and chat token costs be compared directly?

Use the accepted-work unit defined for each service. Run separate calculators with accepted items, input tokens or query-document work as appropriate; label the denominator visibly. The comparison should preserve task quality rather than equate unlike work.

Conclusion

The procurement decision follows three questions: can the chosen artifact satisfy the task, can the fleet satisfy the service objective, and what does that accepted service cost over the complete demand trace? Answer them in that order. A financially attractive fleet that fails the latency or quality contract is not a qualified alternative.

CPU deserves a measured first trial when capacity already exists and the required response behavior leaves room for it. New CPU capacity is a distinct financial case. GPU deserves a measured trial when latency or consolidated qualified capacity can justify its host, accelerator and operating costs. None of these positions requires a universal claim about processor superiority.

The reproducibility contract controls model bytes, precision, engine revision, prompt distribution and power boundary. The sweep records arrivals, queue behavior, percentiles, errors and accepted output. The calculator charges whole servers, accounts for idle periods and leaves prices as dated quote or user inputs. A crossover is consequently a result of the declared workload and accounting policy, with possible additional switches as fleets grow.

The final decision record should include the hardware input sheet, deployment manifest, replay trace, measured energy, accepted-output definition and sensitivity results. It should also state the changes that require retesting. That package gives enterprise architects a reviewable CPU-versus-GPU decision for their private endpoint, while preserving the limits of what the available documentation and their own measurements establish.

Source links

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

1. https://docs.vllm.ai/en/latest/getting_started/installation/cpu/
2. https://github.com/ggml-org/llama.cpp/blob/master/README.md?plain=1
3. https://huggingface.co/Qwen/Qwen2.5-3B-Instruct-GGUF/blob/main/qwen2.5-3b-instruct-q4_k_m.gguf
4. https://huggingface.co/Qwen/Qwen2.5-3B-Instruct-GGUF/blob/main/LICENSE
5. https://huggingface.co/Qwen/Qwen2.5-3B-Instruct-GGUF
6. https://mlcommons.org/benchmarks/endpoints/
7. https://docs.mlcommons.org/endpoints-submission-docs/reference/metrics-and-regions/
8. https://www.nvidia.com/en-us/data-center/l4/
9. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_es1a
10. https://www.intel.com/content/www/us/en/products/sku/232379/intel-xeon-gold-5418y-processor-45m-cache-2-00-ghz/specifications.html
11. https://arxiv.org/abs/2403.02310
12. https://github.com/ggml-org/llama.cpp/blob/master/tools/server/README.md?plain=1
13. https://gpusmith.com/
14. https://www.redhat.com/en/blog/agentic-ai-demands-new-infrastructure-stack-amd-and-red-hat-deliver

15. https://www.amd.com/content/dam/amd/en/documents/products/epyc/4th-gen-amd-epyc-processor-architecture-whitepaper.pdf
16. https://lenovopress.lenovo.com/lp1609-thinksystem-sr635-v3-server
17. https://docs.pytorch.org/tutorials/recipes/recipes/tuning_guide.html
18. https://github.com/ggml-org/llama.cpp/blob/master/docs/build.md
19. https://lenovopress.lenovo.com/lp1717-thinksystem-nvidia-l4-24gb-pcie-gen4-passive-gpu
20. https://gpusmith.com/articles/en/good-gpu-utilization-percentage
21. https://kubernetes.io/docs/tasks/manage-gpus/scheduling-gpus/
22. https://gpusmith.com/articles/en/gpu-cluster-reliability-budget
23. https://www.dell.com/support/manuals/en-ca/oth-r660/r660_ism_pub/thermal-restriction-matrix?guid=guid-546869aa-85a1-41db-b266-e3cd55550c51&lang=en-us
24. https://www.energy.gov/cmei/femp/cooling-water-efficiency-opportunities-federal-data-centers
25. https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/
26. https://www.spec.org/power_ssj2008/
27. https://spec.org/power/docs/specpower_ssj2008-run_reporting_rules/
28. https://github.com/ggml-org/llama.cpp/releases
29. https://developer.nvidia.com/blog/benchmarking-llm-inference-at-scale-with-aiperf/
30. https://mlcommons.org/2026/04/mlperf-inference-v6-0-results/
31. https://prometheus.io/docs/practices/histograms/
32. https://www.energystar.gov/products/enterprise_servers
33. https://gpusmith.com/articles/en/mig-vs-mps-time-slicing-gpu-passthrough
34. https://gpusmith.com/articles/en/air-gapped-llm-deployment-guide

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.