Stop Comparing GPU Providers. Count Your Tokens.

Teams spend weeks comparing GPU clouds to host an open-source model. Most of them shouldn't be renting a GPU at all.

The comparison spreadsheet that feels like rigor

The scenario is common. A team picks an open model, say gpt-oss-120b, and starts shopping for a home. Someone opens the Hugging Face deploy menu and finds six options. Someone else builds a spreadsheet: dollars per GPU-hour on Modal, on RunPod, on Hugging Face Endpoints, on a bare-metal server rented by the month. The rows multiply. The meeting gets scheduled.

It feels like due diligence. Hourly rates are concrete, easy to rank, satisfying to compare. And the spread is real: the same H100 costs around $2 per hour on a reserved bare-metal cloud and up to $8 per hour on the most convenient managed endpoint.

But ranking GPU prices answers a question most teams never ask out loud: *should this workload be on a dedicated GPU in the first place?* Usually, it shouldn't. Flip the comparison.

Where an open model should run is decided by two numbers: how many million tokens you process per month, and how those tokens spread across the day. Not by which provider has the best pricing page.

Volume picks the tier. Timing picks the machine. Everything else is detail.

Why the spreadsheet wins anyway

Hardware prices are visible; usage patterns are not. A GPU rate fits in a cell. "How many tokens will our users actually generate in March, and at what hours?" requires estimation, and estimation feels like guessing. So teams optimize what they can see. It is rational behavior, and it routinely produces a $1,500-per-month server doing the work of a $40 API bill.

The three rungs

Below roughly 50 million tokens per month, renting hardware is a mistake at any price. Open models are served per-token by a dozen competing providers: DeepInfra, Together, Fireworks, Groq, or all of them at once through a router like OpenRouter. For gpt-oss-120b, input tokens start around $0.03 to $0.04 per million. A chat application serving a few hundred users burns single-digit dollars per month at those rates. Even the cheapest rentable GPU, a consumer RTX 4090 at about $132 per month, loses that comparison by an order of magnitude before you count the hours spent maintaining it. One caution from our own benchmarking: the same model varies up to 7.8x in price and dramatically in reliability across these providers, so pick one deliberately rather than routing blind.

Work that arrives in batches belongs on a GPU that dies when the job ends. A daily pipeline of 100,000 inferences does not need an always-on server; it needs two or three hours of compute, once a day. Rent a spot GPU on a marketplace like Vast.ai or RunPod for roughly a dollar an hour, run the batch through an inference engine, shut the machine down. The monthly bill lands between $30 and $90, below both the per-token equivalent and any standing infrastructure. This is the rung the spreadsheet never finds, because it isn't a provider. It's a schedule.

Steady, high-volume traffic is the only case for owning the serving stack. When interactive users keep a GPU busy most of the day (in practice, past 40 to 50% utilization, a threshold our own measurements put at 45%), the economics invert. A reserved H100 at about $2 per hour, running an optimized engine like vLLM, serves billions of tokens a month for a fixed ~$1,500. The same volume billed per-token costs a multiple of that. Public analyses place the crossover between 100 and 500 million tokens per month. Above it, the fixed cost stops being a burden and becomes the discount.

Four situations override the math entirely. If the model is your own fine-tune, per-token providers cannot serve it; you are in GPU territory regardless of volume. If data cannot leave your jurisdiction, you self-host even at a loss. If tail latency or output reliability carries a contract, dedicated tenancy wins on merit: in our field benchmark, a self-hosted endpoint produced zero malformed structured outputs across 344,000 requests, while one API provider on a routed endpoint failed on nearly a third of the same task. And if nobody on the team can own an inference engine, the managed premium (Hugging Face Endpoints, Friendli, Baseten) is not waste. It is the salary of the engineer you didn't hire.

Four questions before you rent anything

1. How many million tokens did you actually serve last month? Measured, not projected.
2. Do those tokens arrive all day, or in a burst you could schedule?
3. Could a per-token provider legally and technically serve your exact model?
4. Who on your team restarts the inference engine at 2 a.m.? If the answer is "nobody," which premium are you choosing to pay instead?

The provider list is long. The decision is two numbers. Count your tokens before you count your GPUs.

Leyton CognitX helps teams run exactly this arithmetic on their own workloads, before the infrastructure bill does it for them.

Sources

- gpt-oss-120b provider pricing and spread: Artificial Analysis, provider benchmark, July 2026 (https://artificialanalysis.ai/models/gpt-oss-120b/providers)
- Reserved H100 rates: Spheron GPU catalog / Lambda pricing, July 2026 (https://www.spheron.network/blog/modal-gpu-pricing-2026-per-second-billing/)
- Serverless vs dedicated break-even analyses, 30 to 61% utilization: Beam, Modal pricing breakdowns, June 2026 (https://www.beam.cloud/blog/modal-pricing-explained)
- Self-hosting inflection, 100 to 500M tokens/month: independent Hugging Face pricing analysis, June 2026 (https://techjacksolutions.com/ai-tools/hugging-face/hugging-face-pricing/)
- RTX 4090 monthly rental range: GetDeploying price tracker, July 2026 (https://getdeploying.com/gpus/nvidia-rtx-4090)
- Structured-output failure rates, 45% break-even, tail-latency findings: Leyton CognitX field benchmark, July 2026 (link to benchmark article/repo).

Revision #2
Created 15 July 2026 15:39:07 by EMB
Updated 10 August 2026 10:21:13 by EMB