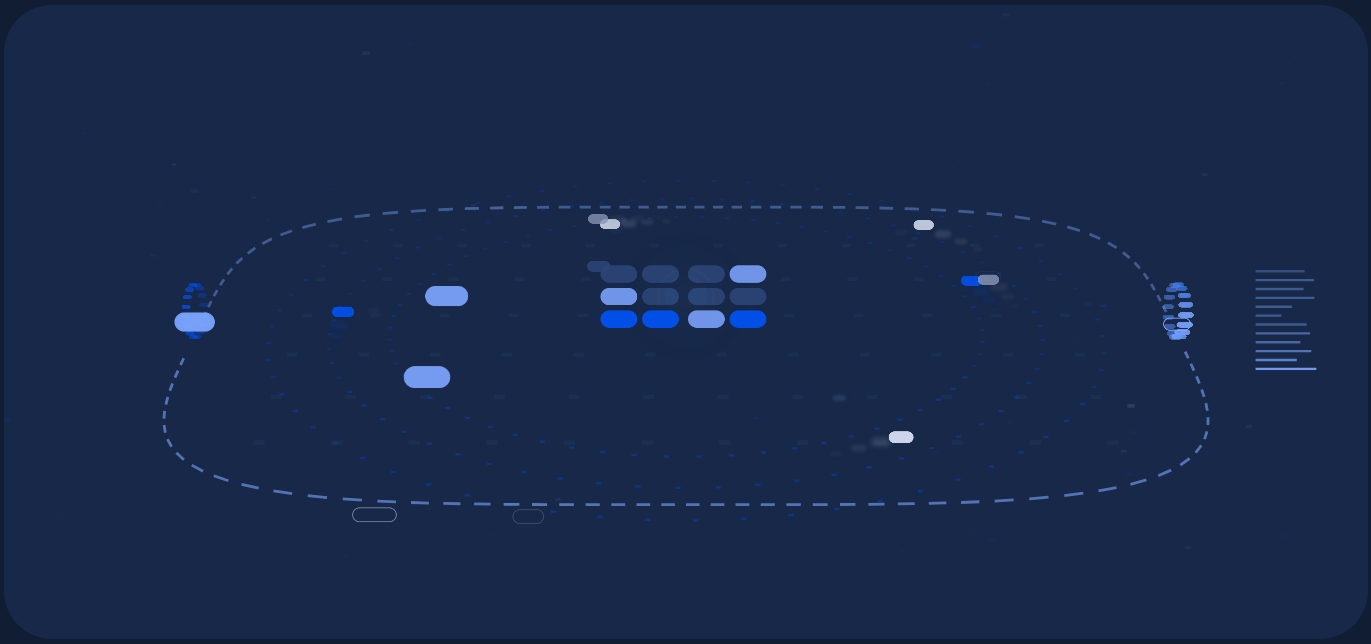
Rillor

METHODS NOTE

Self-hosted LLM vs API cost: a break-even method

A self-hosted model costs less than an API only above a break-even utilization, set by GPU hourly cost, tokens served within a latency limit and the API price.



Rillor

Contents

01 Summary

02 Background

03 Method

04 Findings

05 Implications

06 Limits of this analysis

07 Sources

Summary

- Self-hosted LLM vs API cost turns on the break-even utilization U* = H × 1,000,000 / (3,600 × Θ × API cost), with H the serving unit's hourly cost and Θ its output tokens per second within a latency limit. Self-hosting costs less only above U*, and an estimate that assumes full use understates cost by exactly 1/U [26].
- Throughput has to be read at the latency limit the product needs. In MLPerf logs, the tighter limit kept 42% of sustained output on an 8x H100 system running Llama 2 70B and 15% on an 8x B200 system running gpt-oss-120b [2, 3, 6, 7].
- At prices viewed on 24 October 2026, a rented 8x B200 system serving gpt-oss-120b breaks even against the same model sold per token at 16% to 28% utilization under MLPerf's Server limit for zero to three input tokens per output token, and never under its Interactive limit at those mixes [6, 7, 13, 14]. An owned estimate from filed and published inputs, before financing, facility, staff and residual value, breaks even at 6.4% and 42.3% [6, 7, 13, 20, 21, 22, 30].
- Pan et al. count only GPU purchase and electricity at full use, and their text ranges disagree with their own Table IV [25]. Patil measures cost against request rate and finds $0.21 to $15.25 per million output tokens on one H100 node [26].
- Both sides move. API prices at fixed capability fell 9x to 900x a year in Epoch AI's March 2025 analysis [27], one GPU cloud's H100 price rose 53% from May to October 2026 [15, 16], and one H100 system's MLPerf throughput rose 51% between rounds about a year apart [4, 5].

Background

Published pages on this question give answers that do not line up. A consultancy's guide, reviewed on 28 September 2026, reports a median break-even of 11 to 12 million tokens a day against mid-tier APIs across 900 modeled scenarios, and 10.5 million to 82.2 million a day across the APIs it names, for hardware cost alone [32]. It amortizes hardware over 36 months and adds an operations floor of $1,500 to $2,500 a month [32]. A developer's blog from 17 October 2026 finds break-even for one rented H100 at about 2 billion tokens a month against a hosted open model at $1.04 per million, and about 530 million against a commercial family at a blended $4, with a 3:1 input to output split [33]. A May 2026 article puts break-even against two frontier families at roughly 100 to 256 million tokens a month, with a 50/50 split [34]. A fourth page now calls such a figure "an accounting intersection" and calls fixed thresholds unreliable [35].

Converted at 30 days a month (this report's arithmetic), the first three pages span about 100 million to 2.5 billion tokens a month, a factor of about 25. They differ in the unit (per day or per month, blended or output tokens), the input to output mix, the API tier compared and whether the GPU is assumed busy.

The two studies differ in the same ways. Pan et al. report break-even in months, from 0.3 to 69.3, across nine open-weight models and five commercial APIs [25]. Patil reports cost per million output tokens against offered load, and argues that treating utilization as a fixed input misprices self-hosting and over-sells it for low-traffic workloads [26].

Method

What was examined. The cost of serving output tokens from dedicated GPUs, rented or owned, against the price of the same tokens from a per-token API.

The formula, in US dollars per million output tokens:

- C_self = H × 1,000,000 / (3,600 × Θ_L × U)
- C_api = p_out + r × ((1 - h) × p_in + h × p_cache), times the batch factor where requests can wait, plus cache-write premiums where a provider charges them [11, 12]
- U* = H × 1,000,000 / (3,600 × Θ_L × C_api)
- V* = H × 730 / C_api, in millions of output tokens a month for a unit rented around the clock, valid below its capacity of Θ_L × 3,600 × 730 / 1,000,000

H is the hourly cost of the whole serving unit, Θ_L its output tokens per second within latency limit L, U the share of that throughput used over the hours paid for, r the input tokens per output token and h the share of input served from cache. 730 is 8,760 hours divided by 12.

Units. MLPerf counts generated tokens and checks the reported count against each answer [8]. In the Llama 2 70B log, tokens per second divided by samples per second gives 292.9 tokens per request, against the reference's 294.45 [2, 8]. Patil's costs are per million output tokens, with input tokens consuming compute outside the denominator [26]. Input charges enter through r.

Utilization. Patil defines U as achieved over peak throughput and notes that the latency limit caps the utilization a system can reach [26]. Here the denominator is throughput within the stated limit.

Sources. Benchmark and measurement inputs come from MLPerf [1, 2, 3, 4, 5, 6, 7, 8], vLLM [9] and DistServe [24]. Prices come from three commercial model families and an open-weight host [10, 11, 12, 13] and from GPU rental listings [14, 15, 16, 17, 18, 36]. Owned-cost inputs come from filings, federal data, a survey and a system specification [19, 20, 21, 22, 30]. The studies are [23, 25, 26, 27, 28, 29], with one vendor statement [31]. Four web pages [32, 33, 34, 35] appear only in the Background.

How. Each input was read with its unit and date and converted to an eight-GPU unit. Throughput comes from MLPerf logs, each with its own latency limits. Pan et al.'s Table IV was recomputed from the paper's equations and Tables I to III, and Patil's costs were checked against his throughputs. This report's own arithmetic is labelled as such. Model families stay unnamed in the text, and providers appear in the source list.

Findings

1. API prices, batch discounts and cached input

Three commercial families each list a model at $2.00 per million input tokens and $10.00 per million output tokens, which makes their discounts easy to compare.

Per million tokens	Family A [12]	Family B [11]	Family C [10]
Standard input / output	$2.00 / $10.00	$2.00 / $10.00	$2.00 / $10.00 to 30 November 2026
Batch input / output	$1.00 / $5.00	50% off both	$1.00 / $5.00
Cached input (hit)	$0.10	$0.10	$0.10
Cache write	$2.50 (30 minutes), $5.00 (12 hours)	1.25x input (5 minutes), 2x (1 hour)	Implicit caching only
Faster tier, output	$20.00 Fast, $60.00 Ultrafast	Fast mode at 2x, other model	$18.00 Priority

Sources: [10, 11, 12], viewed 24 October 2026.

Batch. All three sell batch processing at half price [10, 11, 12], "a 50 percent discount on both input and output tokens" in Family B's words [11]. Batch halves C_api and so doubles U*.

Caching. Family B charges 0.1x base input for a cache hit on most models and 0.05x or 0.025x on some newer ones, and these multipliers stack with batch and residency pricing [11]. With r = 3 and half the input cached, C_api on the $2.00/$10.00 model is $13.15 against $16.00 uncached (this report's arithmetic from [11, 12]).

Speed and residency. Faster tiers cost 1.8x to 6x standard [10, 11, 12]. Data residency adds 1.1x at Family B [11] and 10% on newer models at Family A [12].

Tiers, schedules and units. In their current flagship tables, output prices at Families A and B run from $0.50 to $50.00 per million tokens, a factor of 100 [11, 12]. Family C's $2.00/$10.00 model doubles on 1 December 2026, two smaller models double from $0.75/$3.75 on 1 January 2027, and thinking tokens are billed as output [10]. Family B's newer tokenizer "produces approximately 30% more tokens for the same text" [11], so per-token prices compare different quantities across families.

The like-for-like price. A host listed gpt-oss-120B at $0.15 per million input tokens and $0.60 per million output tokens on 24 October 2026, and single-tenant dedicated HGX H100 and HGX B200 GPUs at $5.49 and $8.99 per GPU-hour [13]. The same model on both sides needs no claim about quality, so finding 7 uses it.

2. GPU hourly cost, rented or owned

Rented.

Offer	Per GPU-hour	Date
GPU cloud 1, 8x H100 SXM and 8x B200 instances	$3.99 and $6.69	Viewed 24 October 2026 [14]
GPU cloud 1, 1x H100 SXM and 1x B200	$4.29 and $6.99	Viewed 24 October 2026 [14]
GPU cloud 1, 8x H100 and 8x B200 instances, from 16 November 2026	$4.59 and $7.69	Viewed 7 November 2026 [36]
GPU cloud 2, HGX H100 and HGX B200 on demand	$4.50 and $8.50	Viewed 24 October 2026 [15]
GPU cloud 2, same	$2.95 and $5.50	Archived 7 May 2026 [16]
Hyperscaler, 2x H100 NVL node [26], pay-as-you-go	$6.98 ($13.96 per node)	Effective 1 March 2024 [17]
Same node, spot	$1.42	Effective 1 November 2026 [17]
Same node, 1-year and 3-year reservation	About $5.23 and $3.84 (derived)	Viewed 24 October 2026 [17]

The hyperscaler's API returns reservation prices of $91,717 for one year and $201,778 for three [17]. Read as term totals, they equal $10.47 and $7.68 an hour per node, exactly 25.0% and 45.0% below pay-as-you-go, which supports that reading. GPU cloud 2 offered up to 35% off on-demand rates for multi-month cluster reservations in May 2026 [16]. Patil costed single GPUs at half the $13.96 node, assuming a co-tenant uses the second GPU, and says a dedicated operator should double those figures [26]. How posted prices relate to paid prices is set out in what posted GPU cloud prices tell you.

Owned. No filing gives an hourly cost for owned GPUs, so this report builds one from dated inputs.

- Purchase. IREN bought 2,400 Blackwell GPUs, 1,300 B200 and 1,100 B300, "for ~$130m, including fit-out costs", which covers servers, storage, labor, InfiniBand, racks and cabling [20]. That is about $54,167 per GPU.
- Life. Amazon moved a subset of its servers from six years to five from 1 January 2025, citing "an increased pace of technology development, particularly in the area of artificial intelligence and machine learning" [21].
- Power. IREN plans on 1.93 kW per GPU including ancillary draw, at a PUE of 1.1 [20]. NVIDIA lists a maximum of 10.2 kW for an eight-GPU DGX H100 [19].
- Facility. Uptime Institute's 2025 survey reports a weighted average PUE of 1.54, 1.48 for facilities built within five years and 1.44 at 20 MW and above [30].
- Electricity. U.S. averages for the 12 months to August 2026 were 13.87 cents per kWh commercial and 8.91 cents industrial, with 2026 values preliminary [22].

Life	PUE 1.1, industrial	PUE 1.54, industrial	PUE 1.54, commercial
4 years	$1.74	$1.81	$1.96
5 years	$1.43	$1.50	$1.65
6 years	$1.22	$1.30	$1.44

Owned cost per GPU-hour: straight-line purchase cost plus 1.93 kW × PUE × power price. This report's arithmetic from [20, 21, 22, 30].

The table leaves out financing, facility space, maintenance and spares, software, staff and end-of-life value, and it charges full power at every hour. Financing terms are described in GPU financing structures explained (https://rillor.com/insights/gpu-financing-structures).

3. Tokens per GPU hour within a latency limit

MLPerf's Server and Interactive scenarios send queries "according to a Poisson distribution", require 99% of them to meet the latency limits and report the peak Poisson rate the system supports [1]. The limits are time to first token (TTFT) and time per output token (TPOT). MLPerf disallows caching across queries, so every input is computed in full [1].

Model	Looser limit (TTFT / TPOT)	Tighter limit
Llama 2 70B	2,000 ms / 200 ms	450 ms / 40 ms

Model	Looser limit (TTFT / TPOT)	Tighter limit
gpt-oss-120b	3,000 ms / 80 ms	2,000 ms / 20 ms

Source: MLPerf Inference rules, file last changed 20 August 2026 [1].

The v6.0 gpt-oss-120b Interactive log records a 15 ms TPOT limit [7], tighter than the 20 ms in the current rules [1]. Each result below uses its own log's limit.

System, round, model	Limit in log	Output tokens/s	Per GPU	p99 TTFT / TPOT
8x H100 (XE9680), v5.0, Llama 2 70B	2,000 / 200 ms	30,903.64	3,863	1,939 / 146 ms
8x H100 (XE9680), v5.0, Llama 2 70B interactive	450 / 40 ms	12,979.44	1,622	269 / 37 ms
8x B200, v6.0, gpt-oss-120b Server	3,000 / 80 ms	87,444.22	10,931	1,306 / 45 ms
8x B200, v6.0, gpt-oss-120b Interactive	2,000 / 15 ms	13,155.34	1,644	146 / 14.6 ms

Sources: [2, 3, 6, 7]. All four runs are marked VALID. Per-GPU figures divide by eight.

The tighter limit kept 42% of output on the H100 system and 15% on the B200 system, so cost per token rises 2.4x and 6.6x [2, 3, 6, 7]. Each gpt-oss-120b request produced about 1,309 output tokens (this report's division of tokens per second by samples per second) [6, 7].

DistServe describes the cause: under stringent latency requirements, serving systems "have to prioritize one latency over the other, or over-provision compute resources to meet both" [24]. It measures performance as the rate served within both TTFT and TPOT limits on each GPU [24], which vLLM's benchmark tool reports as goodput, citing DistServe [9].

MLPerf systems are tuned by their submitters, on MLPerf's prompts. The vLLM tool sends requests at a set Poisson or burstier rate and reports goodput against TTFT, TPOT and end-to-end limits [9]. Patil's workflow feeds goodput measured at the operator's own rates and limit into the cost formula [26]. His runs used vLLM defaults without prefix caching, and he reads his costs as upper bounds [26].

4. Utilization and the 1/U penalty

Patil writes effective cost as the GPU price over achieved throughput and naive cost as the GPU price over peak throughput, with "The ratio Ceff /Cnaive = Θmax /Θachieved = 1/U" [26]. A calculator that leaves utilization at 100% understates cost by exactly that factor. The identity follows from the definitions. What the paper measures is how far U falls at realistic loads.

On one H100 serving Llama 3.1 8B at $6.98 an hour, with a 512:256 input to output shape, Patil's Table 3 gives [26]:

Requests per second	p99 TTFT	$ per million output tokens	Penalty
1	54.3 ms	$7.60	24.4x
5	713.3 ms	$1.54	5.0x
10	84.5 ms	$0.79	2.5x
25	259.0 ms	$0.37	1.18x
50	994.8 ms	$0.32	1.02x
100	5,011 ms	$0.31	1.00x
200	7,437 ms	$0.31	1.00x

Patil attributes the 5 requests per second tail to a few outlier requests in one unrepeated run [26].

At peak throughput of 6,238 tokens per second the naive cost is $0.311 [26]: $6.98 / (6,238 × 3,600 / 1,000,000) = $0.311. At one request per second the penalty ran from 17.5x to 36.3x across configurations, converging to about 1.0x by 50 requests per second, and across six configurations and all loads cost spanned $0.21 to $15.25 per million output tokens [26].

Patil then takes, for each configuration, the highest swept load that meets 300 ms TTFT and 50 ms TPOT at p99 [26]. For Llama 3.1 8B in FP16 that is 25 requests per second, at $0.368 or 1.18x its floor. For Mixtral 8x7B in FP16 it is 10 requests per second, at $1.666 or 1.91x its floor of $0.871 [26].

Real load is uneven. In SageServe's study of an enterprise productivity suite with over 10 million LLM requests a day, latency-sensitive workloads "exhibit clear diurnal periodicity with weekends quiescing", and separate pools for fast and slow work leave "significant under-utilization" off-peak [23]. Its scheduler, which shares one

pool and defers flexible work, saved up to 25% of GPU-hours against the baseline and cut autoscaling waste by 80% [23]. The paper gives no single average utilization.

Utilization counts over the hours paid for. A load that runs at the limit for 8 hours on 22 working days, 176 hours, and idles otherwise has U = 176 / 730 = 24% on a unit rented around the clock (this report's arithmetic). Patil does not model GPU sharing among tenants, which can raise utilization for providers that serve many customers [26].

5. What two published studies assume and find

Pan et al. Version 3 of 11 November 2025 sets hardware cost as GPU count times price, an A100 80 GB at $15,000 and 400 W or an RTX 5090 at $2,000 and 575 W, and electricity at "8 hours/day and 20 days/month" and $0.15 per kWh [25]. API cost is "normalized by the token generation capacity" of the local hardware, priced at one third input and two thirds output tokens, and break-even is the month at which the two costs meet [25]. Five commercial APIs, listed from $1.25/$10.00 to $15.00/$75.00 per million input/output tokens, are compared with open-weight models scoring "within 20% of top commercial models" [25]. Table IV runs from 0.3 months, for EXAONE 4.0 32B and Qwen3-30B against the $15.00/$75.00 API, to 69.3 months, for Kimi-K2 against a $1.25/$10.00 API, a spread of 231x [25].

Three features of the setup matter. The paper lists cooling, maintenance, personnel and software licensing as operating costs, but its equations count only GPU purchase and electricity, and its Future Work calls for adding "staffing and maintenance" [25]. Pricing the API at the hardware's capacity sets U to 100% in operating hours, which Patil describes as "full utilization during 160 operational hours per month" [26]. Throughput carries no latency limit [25].

This report's recomputation found five internal mismatches.

1. The text gives medium-model break-even as "3.8 to 34 months" and large-model break-even as 3.5 to 69.3 months, while Table IV gives 2.3 to 34.0 for the medium group and 4.3 to 69.3 for the large group [25].
2. The conclusion gives recovery within 6 to 24 months for medium models, and the introduction "2 years for medium models and 5 years for larger models" [25]. Neither matches the table.
3. The stated hours are 160 a month. Table III implies 176: Llama-3.3-70B at 190 tokens per second gives the stated 120.4 million tokens a month only at 176 hours [25]. At 176 hours, Llama-3.3-70B against a $1.25/$10.00 API recomputes to 17.81 months, matching the table's 17.8, and at 160 hours to 19.59.
4. The text describes "54 deployment scenarios" and "six commercial APIs" [25]. Tables I and IV show five APIs and 45 cells.
5. Two API columns that Table I lists at the same $1.25/$10.00 differ by about 9% in every row, for example 63.1 against 69.3 for Kimi-K2 [25]. The paper does not explain this.

At 176 hours the recomputation matches the higher of those two columns and the other three within about 1%. This report cites Table IV's values.

Patil. The paper is a 26-page single-author arXiv preprint by an independent researcher, submitted on 10 June 2026 with no journal reference [26]. It uses one cloud node with two H100 NVL 96 GB GPUs at $13.96 an hour, three models at two precisions, and six configurations swept at seven arrival rates for 42 runs, on a uniform 512:256 workload [26]. Its API comparison uses list output prices of 9 June 2026 with no discounts [26]. The "n=3" caveat in its abstract refers to the three architectures behind its quantization finding [26], and does not qualify the 1/U identity.

Patil adds that published per-token prices are "unconditioned on any production service-level objective", since providers publish no enforceable p99 guarantees on TTFT or inter-token latency [26]. A latency-limited unit and a per-token API sell different products at the tail.

6. Prices move on both sides

API prices. Epoch AI measured the price of reaching fixed benchmark scores falling 9x to 900x a year, median 50x, on a 3:1 input to output blend, and 200x at the median using data from January 2024 on [27]. The authors note that the larger drops came in the past year, "so it's less clear that those will persist", and found no clear evidence that thinner margins explain them [27]. Scheduled increases run the other way, with Family C's doublings on 1 December 2026 and 1 January 2027 [10].

Rental prices. GPU cloud 2's on-demand H100 rose from $2.95 per GPU-hour on 7 May 2026 to $4.50 on 24 October 2026, up 53%, and its B200 from $5.50 to $8.50, up 55% [15, 16]. GPU cloud 1 lists its 8x H100 and 8x B200 instances at $4.59 and $7.69 per GPU-hour from 16 November 2026, 15% above the rates in finding 7 [14, 36]. A hyperscaler cut on-demand prices from 1 June 2025 by up to 45% for P5, 26% for P5en and 33% for P4d and P4de instances on Amazon Linux [18].

Throughput. The same DGX H100 system served Llama 2 70B at 20,556.40 tokens per second in MLPerf v4.0 (March 2024) and 31,106.28 in v5.0 (April 2025), under the same limits [4, 5]. That is 51% more output on unchanged hardware, about a third off cost per token (this report's arithmetic). NVIDIA states that B200 cost per million tokens on gpt-oss-120b fell from $0.11 at launch to $0.02 within two months, citing SemiAnalysis InferenceX as of April 2026, a 5x gain it attributes to software [31]. That is a vendor statement.

Hardware life. Amazon's cut from six years to five [21] matters for owned cost: in the finding 2 table, a four-year life adds about $0.51 an hour per GPU over a six-year life.

7. Worked example with dated public prices

The rented B200 unit costs $53.52 an hour ($6.69 × 8) and the H100 unit $31.92 ($3.99 × 8) [14]. At full use, the B200 unit costs $0.170 per million output tokens under the Server limit and $1.130 under the Interactive limit, and $0.567 and $3.767 at 30% utilization. The H100 unit costs $0.287 and $0.683 at full use, and $0.956 and $2.277 at 30% (this report's arithmetic from [2, 3, 6, 7, 14]).

Break-even utilization against the same model sold per token.

r	API cost per million output tokens	U*, Server limit	U*, Interactive limit
0	$0.60	28.3%	188%
0.5	$0.675	25.2%	167%
1	$0.75	22.7%	151%
2	$0.90	18.9%	126%
3	$1.05	16.2%	108%

Rented 8x B200 at $6.69 per GPU-hour. This report's arithmetic from [6, 7, 13, 14].

At GPU cloud 2's $8.50 [15], U* is 36.0% to 20.6% under the Server limit and 239% to 137% under the Interactive limit (this report's arithmetic from [6, 7, 13, 15]). At the host's own dedicated B200 price of $8.99 [13], cost is $0.228 per million output tokens at full use under the Server limit, and U* is 38.1% at r = 0 and 21.8% at r = 3 (this report's arithmetic from [6, 13]).

Break-even volume. Around the clock, the B200 unit costs $53.52 × 730 = $39,070 a month. It pays off above about 65.1 billion output tokens a month at r = 0, and about 37.2 billion at r = 3, against a capacity of about 229.8 billion within the Server limit (this report's arithmetic). The working-hours load from finding 4, 24% of the month, falls below U* at r = 0 and above it at r = 3.

Owned. At $12.01 an hour for eight GPUs (five years, PUE 1.54, industrial power), cost is $0.038 per million output tokens under the Server limit and $0.254 under the Interactive limit at full use, with break-even against $0.60 at 6.4% and 42.3% (this report's arithmetic from [6, 7, 13, 20, 21, 22, 30]). The costs left out in finding 2 would raise this figure and residual value would lower it, by amounts these sources do not give. The IREN purchase mixes B200 and B300 GPUs [20], while the throughput is for B200 [6, 7].

Against commercial families. Against a family listed at $10 per million output tokens [10, 11, 12], the rented B200 unit breaks even at 1.7% utilization under the Server limit and 11.3% under the Interactive limit (this report's arithmetic from [6, 7, 10, 11, 12, 14]). Against $50 [11, 12], at 0.34% and 2.3% (this report's arithmetic from [6, 7, 11, 12, 14]). These hold only where the open-weight model does the task as well. Epoch AI found in 2024 that frontier open models trailed frontier closed models by 5 to 22 months on benchmarks [28], and in October 2026 placed the open frontier on its cyber index where the closed frontier stood 6 to 7 months earlier [29]. None of this establishes parity on a reader's task.

Approximations. No per-token price for Llama 2 70B was confirmed, so the H100 figures show the latency effect only. MLPerf throughput includes MLPerf's own prompts, whose input lengths the sources opened do not state, so varying r on the API side is approximate, and a reader's own Θ [9] replaces it.

Implications

For CTOs and platform engineers. The answer rests on three measurements a team can make: Θ on its own prompts at the latency limit the product needs [9, 26], utilization by the hour from traffic logs, since load follows the working day and week [23], and U* for each option. The latency limit can decide the outcome alone: in the B200 example, the same hardware and prices break even at 16% to 28% under one limit and never under the other [6, 7, 13, 14]. Work that can wait has two cheaper paths, a batch tier at half price [10, 11, 12] or deferrable jobs pooled onto self-hosted capacity to raise U [23]. Sizing the hardware, platform and models is the Design step when Rillor deploys a private superintelligence (SI) cloud (https://rillor.com/compute/private-si), a dedicated environment on the customer's servers, colocation space or dedicated GPU capacity that Rillor sources and contracts for the customer.

For FinOps leads. Date every input and plan reruns at known changes, such as Family C's doublings on 1 December 2026 and 1 January 2027 [10]. Compare like tiers: batch against batch, priority against a latency-limited deployment. Count sales tax where a price page adds it [14]. Reservations lowered H by 25% and 45% on one hyperscaler node [17], which lowers U* by the same share.

For readers of published figures. Check five things before using a break-even number: the unit, the input to output mix, the API tier, the utilization assumed and the costs counted. Pan et al.'s Table IV assumes full use in operating hours (stated as 160 a month, though its Table III capacities imply 176) and counts GPU purchase and electricity only [25]. The pages in the Background mix 3:1 and 50/50 splits and daily and monthly units [32, 33, 34], and one now rejects fixed thresholds [35].

Limits of this analysis

- **Staff.** No primary, dated source opened gives the staff cost of running a self-hosted inference stack. The only figure found is one consultancy's assumed operations floor of $1,500 to $2,500 a month [32]. Pan et al. list personnel as a cost and leave it out of their equations [25].
- **Power.** IREN's 1.93 kW per GPU is a planning figure that includes ancillary draw [20], applied here at every hour. EIA's prices are U.S. sector averages [22], and a given site's tariff differs.
- **Quality parity.** No source opened shows an open-weight model matching a commercial family on a reader's task, and Epoch AI's lags vary by domain and date [28, 29].
- **Throughput.** MLPerf systems are tuned by their submitters [2, 3, 6, 7], MLPerf disallows caching across queries [1], and its input lengths are not stated in the sources opened.
- **Utilization.** No source gives a typical production utilization for self-hosted serving. SageServe shows cycles without an average [23], and Patil uses fixed request rates on a synthetic workload [26].
- **Owned cost.** It excludes financing, facility space, maintenance, software, staff and residual value, and rests on one July 2025 purchase of mixed B200 and B300 GPUs [20].
- **Service levels.** Per-token APIs publish no enforceable p99 latency guarantees [26].
- **Prices and units.** Prices were viewed 24 October 2026, and the November rows in finding 2 on 7 November 2026. They change often [10, 15, 16, 36]. Tokenizers differ across families [11]. Reservation hourly rates are derived from term totals [17].
- **Studies.** Patil's paper is a single-author preprint [26]. Pan et al.'s text and tables disagree [25].
- **Scope.** This is a method for engineering cost comparisons. It gives no investment, accounting or procurement advice.

Sources

1. MLPerf Inference Rules (inference_rules.adoc). MLCommons, last changed 20 August 2026. https://github.com/mlcommons/inference_policies/blob/master/inference_rules.adoc

2. MLPerf Inference v5.0, closed: XE9680 8x H100-SXM-80GB, llama2-70b-99.9, Server log. MLCommons, April 2025. https://github.com/mlcommons/inference_results_v5.0/blob/main/closed/Dell/results/XE9680_H100_SXM_80GBx8_TRT/llama2-70b-99.9/Server/performance/run_1/mlperf_log_summary.txt

3. MLPerf Inference v5.0, closed: XE9680 8x H100-SXM-80GB, llama2-70b-interactive-99.9, Server log. MLCommons, April 2025. https://github.com/mlcommons/inference_results_v5.0/blob/main/closed/Dell/results/XE9680_H100_SXM_80GBx8_TRT/llama2-70b-interactive-99.9/Server/performance/run_1/mlperf_log_summary.txt

4. MLPerf Inference v4.0, closed: DGX H100 8x H100-SXM-80GB, llama2-70b-99.9, Server log. MLCommons, March 2024. https://github.com/mlcommons/inference_results_v4.0/blob/main/closed/NVIDIA/results/DGX-H100_H100-SXM-80GBx8_TRT/llama2-70b-99.9/Server/performance/run_1/mlperf_log_summary.txt

5. MLPerf Inference v5.0, closed: DGX H100 8x H100-SXM-80GB, llama2-70b-99.9, Server log. MLCommons, April 2025. https://github.com/mlcommons/inference_results_v5.0/blob/main/closed/NVIDIA/results/DGX-H100_H100-SXM-80GBx8_TRT/llama2-70b-99.9/Server/performance/run_1/mlperf_log_summary.txt

6. MLPerf Inference v6.0, closed: nebius_b200_n1 8x B200-SXM-180GB, gpt-oss-120b, Server log. MLCommons, March 2026. https://github.com/mlcommons/inference_results_v6.0/blob/main/closed/Nebius/results/nebius_b200_n1/gpt-oss-120b/Server/performance/run_1/mlperf_log_summary.txt

7. MLPerf Inference v6.0, closed: nebius_b200_n1 8x B200-SXM-180GB, gpt-oss-120b, Interactive log. MLCommons, March 2026. https://github.com/mlcommons/inference_results_v6.0/blob/main/closed/Nebius/results/nebius_b200_n1/gpt-oss-120b/Interactive/performance/run_1/mlperf_log_summary.txt

8. Llama 2 70B reference implementation (language/llama2-70b/README.md). MLCommons, viewed 24 October 2026. https://github.com/mlcommons/inference/blob/master/language/llama2-70b/README.md

9. vllm/benchmarks/serve.py, benchmark options. vLLM project, viewed 24 October 2026. https://github.com/vllm-project/vllm/blob/main/vllm/benchmarks/serve.py

10. Gemini Developer API pricing. Google AI for Developers, last updated 22 October 2026, viewed 24 October 2026. https://ai.google.dev/gemini-api/docs/pricing

11. Pricing. Claude Platform documentation, Anthropic, viewed 24 October 2026. https://platform.claude.com/docs/en/about-claude/pricing

12. Pricing. OpenAI API documentation, OpenAI, viewed 24 October 2026. https://developers.openai.com/api/docs/pricing

13. Pricing. Together AI, viewed 24 October 2026. https://www.together.ai/pricing

14. GPU cloud pricing, archived capture of 25 October 2026 showing the rates viewed on 24 October 2026. Lambda, via the Internet Archive Wayback Machine, viewed 7 November 2026. https://web.archive.org/web/20261025040216/https://lambda.ai/pricing

15. AI Cloud pricing. Nebius, viewed 24 October 2026. https://nebius.com/prices

16. AI Cloud pricing, archived capture. Nebius, via the Internet Archive Wayback Machine, 7 May 2026. https://web.archive.org/web/20260507111158/https://nebius.com/prices

17. Azure Retail Prices API, query for Standard_NC80adis_H100_v5 in East US. Microsoft Azure, viewed 24 October and 7 November 2026. https://prices.azure.com/api/retail/prices?$filter=armSkuName%20eq%20%27Standard_NC80adis_H100_v5%27%20and%20armRegionName%20eq%20%27eastus%27

18. Pricing and usage model updates for Amazon EC2 instances accelerated by NVIDIA GPUs. Amazon Web Services, 5 June 2025. https://aws.amazon.com/about-aws/whats-new/2025/06/pricing-usage-model-ec2-instances-nvidia-gpus/

19. Introduction to NVIDIA DGX H100/H200 Systems (DGX H100/H200 User Guide). NVIDIA, viewed 24 October 2026. https://docs.nvidia.com/dgx/dgxh100-user-guide/introduction-to-dgxh100.html

20. IREN Expands AI Cloud with 2.4k NVIDIA Blackwell GPUs (Exhibit 99.1 to Form 8-K). IREN Limited, 3 July 2025. https://www.sec.gov/Archives/edgar/data/1878848/000114036125024747/ef20051566_ex99-1.htm

21. Amazon.com, Inc. Annual Report on Form 10-K for 2024. Amazon.com, Inc., filed with the U.S. Securities and Exchange Commission 7 February 2025. https://www.sec.gov/Archives/edgar/data/1018724/000101872425000004/amzn-20241231.htm

22. Electric Power Monthly, Table 5.3, Average Price of Electricity to Ultimate Customers, data for August 2026. U.S. Energy Information Administration, 23 October 2026. https://www.eia.gov/electricity/monthly/epm_table_grapher.php?t=epmt_5_3

23. Jaiswal et al. SageServe: Optimizing LLM Serving on Cloud Data Centers with Forecast Aware Auto-Scaling. Proceedings of the ACM on Measurement and Analysis of Computing Systems 9(3), Article 61, December 2025. https://arxiv.org/abs/2502.14617

24. Zhong et al. DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving. arXiv:2401.09670, 18 January 2024, version 3 of 6 June 2024. https://arxiv.org/abs/2401.09670

25. Pan, Chodnekar, Roy and Wang. A Cost-Benefit Analysis of On-Premise Large Language Model Deployment: Breaking Even with Commercial LLM Services. arXiv:2509.18101v3, 11 November 2025. https://arxiv.org/html/2509.18101v3

26. Patil. Beyond Per-Token Pricing: A Concurrency-Aware Methodology for LLM Infrastructure Cost Estimation. arXiv:2606.11690v1, 10 June 2026. https://arxiv.org/abs/2606.11690

27. LLM inference prices have fallen rapidly but unequally across tasks. Epoch AI, 12 March 2025. https://epoch.ai/data-insights/llm-inference-price-trends

28. How far behind are open models? Epoch AI, 4 November 2024. https://epoch.ai/blog/open-models-report

29. Open-weight models trail closed models in cyber capabilities by 6 to 7 months. Epoch AI, 16 October 2026. https://epoch.ai/data-insights/open-closed-cyber-eci-gap

30. Uptime Institute Global Data Center Survey 2025 (UII Keynote Report 180). Uptime Institute, July 2025. https://datacenter.uptimeinstitute.com/rs/711-RIA-145/images/2025.Annual.Survey.Report.pdf

31. AI Inference Performance Benchmarks. NVIDIA Developer, viewed 24 October 2026. https://developer.nvidia.com/deep-learning-performance-training-inference/ai-inference

32. Cost to Host a Private LLM: Self-Hosted vs API (2026). NavyaAI, last reviewed 28 September 2026. https://www.navyaai.com/guides/self-host-llm-vs-api-cost

33. Self-Hosted LLM vs API: The Real Break-Even (2026). MD Rakibul Islam Rakib, 17 October 2026. https://www.rakibulinux.com/blogs/self-hosted-llm-vs-api-cost-break-even

34. Self-Hosted LLM vs API: The Real Cost and Security Trade-offs for Enterprise in 2026. Marka Development, 13 May 2026. https://www.marka-development.com/news/self-hosted-llm-vs-api-the-real-cost-and-security-trade-offs-for-enterprise-in-2026/

35. Self-Hosted LLM vs API: Cost, Break-Even and Performance. Braincuber Technologies, 10 March 2026, reviewed 21 October 2026. https://www.braincuber.com/blog/self-hosted-llms-vs-api-based-llms-cost-performance-analysis

36. GPU cloud pricing, instance rates taking effect 16 November 2026. Lambda, viewed 7 November 2026. https://lambda.ai/pricing

About this report

Published by Rillor on 7 November 2026. Superintelligence (SI) tools assist research and drafting, and every figure cites its source. Rillor builds and secures SI systems, supplies the data they learn from and delivers the infrastructure they run on. We apply SI to financial and compute markets.

Cite as: Rillor. Self-hosted LLM vs API cost: a break-even method. 7 November 2026. https://rillor.com/insights/self-hosted-llm-vs-api-cost

Notices

Rillor is not a registered investment adviser or commodity trading advisor and does not provide investment or trading advice. Research, forecasts, data and software described on this site are for research and engineering use. Nothing here is an offer or recommendation to buy or sell any security, commodity interest or digital asset. Past or simulated results do not indicate future results.

NVIDIA, AMD, Ceph, Kubernetes and related product names are trademarks of their owners and are used only to identify products. No affiliation or endorsement is implied.

Superintelligence (SI) outputs can be wrong. Important decisions should include human review.

Work with Rillor

Send us the workloads you want to run, the data they use, who uses them and where your hardware sits today. Contact us at rillor.com/contact?type=private-si or contact@rillor.com. More on private SI cloud: rillor.com/compute/private-si