Llama 4 Hardware Requirements: VRAM, Nodes and Costs (2026)

Published July 29, 2026 35 min read

Llama 4 Hardware Requirements: VRAM, Nodes and Costs (2026)

Executive Summary

Meta released **Llama 4** in April 2025 as a family of mixture-of-experts (MoE) models, and by July 2026 the practical question for most teams is no longer whether the models are capable but what hardware is required to run them. The family has two open-weight members: **Llama 4 Scout**, which holds 17 billion active parameters and 109 billion total parameters across 16 experts, and **Llama 4 Maverick**, which holds the same 17 billion active parameters spread across 400 billion total parameters and 128 experts (Source: ai.meta.com) (Source: huggingface.co). Because Llama 4 is a MoE architecture, the entire parameter set must sit in memory even though only 17 billion parameters compute per token, which is why Scout's video random access memory (VRAM) footprint is closer to a 109-billion-parameter dense model than to a 17-billion-parameter one. Meta states Scout "fits on a single H100 GPU (with Int4 quantization)" while Maverick "fits on a single H100 host," meaning an eight-GPU NVIDIA DGX-class server (Source: ai.meta.com). Meta's own developer site repeats this, calling Scout a model with "single H100 GPU efficiency" (Source: llama.com).

For self-hosting, Scout's bfloat16 (BF16) weights occupy roughly 218 gigabytes (GB), which drops to approximately 55 to 61 GB at 4-bit quantization, fitting a single 80 GB H100 or two consumer GPUs; Maverick's FP8 (8-bit floating point) release is designed to fit a single H100 DGX host with eight 80 GB GPUs (640 GB combined) (Source: huggingface.co). For multi-node inference, vLLM documentation shows that eight H100 GPUs can serve Scout with a full 1 million-token context and Maverick with about 430,000 tokens of context using tensor-parallel-size 8 (Source: vllm.ai). A vLLM GitHub issue reporter described Maverick as a model that "most people will be running across multiple GPU nodes"; this is user-submitted context for a pipeline-parallelism feature request, not vLLM deployment guidance (Source: github.com).

Cloud GPU rental for the underlying hardware runs from about **$2.89 to $4.29 per hour** for a single H100 on RunPod and Lambda (Source: www.runpod.io) (Source: lambda.ai), and roughly $5.89 to $9.86 per hour for a Blackwell B200 (Source: www.runpod.io) (Source: lambda.ai). Application programming interface (API) pricing is far cheaper per token: DeepInfra lists Maverick at $0.20 per million input tokens and $0.80 per million output tokens, and Scout at $0.10 and $0.30 respectively (Source: deepinfra.com) (Source: deepinfra.com), Together AI lists Maverick at

$0.27/$0.85 and Scout at $0.18/$0.59 (Source: www.together.ai) (Source: www.together.ai), and Meta itself projects Maverick can be served at "$0.19/Mtok (3:1 blended)...assuming distributed inference," rising to $0.30 to $0.49 per million tokens on a single host (Source: llama.com). On raw throughput, NVIDIA measured over 1,000 tokens per second per user on Maverick using a single DGX B200 node (Source: developer.nvidia.com), while Cerebras later reported 2,522 tokens per second on the same model, more than doubling NVIDIA's figure (Source: www.cerebras.ai). Meta's Llama 4 Community License requires companies with more than 700 million monthly active users to seek a separate license (Source: llama.com), leaving nearly every enterprise deployment unaffected. This report walks through the full hardware and cost landscape: VRAM tables by quantization, multi-GPU and multi-node topologies, per-token API economics across seven-plus providers, and named deployments ranging from Meta's own consumer apps to Cerebras' wafer-scale inference cloud.

Introduction and Background

When Meta introduced Llama 4 on April 5, 2025, it marked the company's first large language model (LLM) family built on a mixture-of-experts architecture rather than the dense transformer design used in Llama 2 and Llama 3 (Source: ai.meta.com). In an MoE model, each token is routed through only a subset of the network's total parameters rather than all of them, which is why Meta describes Llama 4 as activating "17 billion active parameters" regardless of whether the total parameter count is 109 billion (Scout) or 400 billion (Maverick) (Source: ai.meta.com). This design choice is the single most important fact for anyone estimating hardware requirements: the active-parameter count governs compute and generation speed, but the total-parameter count governs how much VRAM or system memory must be available before the model can run at all, because every expert's weights must be resident even if most go unused on any given token. Meta itself describes Scout as a "class-leading natively multimodal model that offers superior text and visual intelligence" (Source: llama.com), with Together AI recommending it specifically for "multi-document analysis, codebase reasoning, and personalized tasks" (Source: www.together.ai).

That distinction explains why "Llama 4 hardware requirements" searches so often surface conflicting numbers. A superficial reading of "17 billion active parameters" suggests a model that should run on a mid-range consumer GPU, similar to other 17-billion-parameter dense models. In practice, Scout's 109 billion total parameters mean the unquantized (BF16) weights alone occupy roughly 218 GB, and Maverick's 400 billion total parameters push that figure past 800 GB (Source: huggingface.co). Meta anticipated this confusion and built quantization support directly into the release: Scout ships with instructions for on-the-fly Int4 (4-bit integer) quantization to fit a single 80 GB H100, and Maverick ships with both BF16 and FP8 (8-bit floating point) checkpoints, with the FP8 version sized to fit a single eight-GPU H100 DGX host (Source: huggingface.co). This quantization-driven flexibility also explains why per-token API pricing across providers, examined in detail later in this report, can vary by a factor of five or more for functionally similar output quality: providers that serve Maverick at FP8 precision on cost-optimized hardware can charge meaningfully less per token than those defaulting to less efficient configurations or bundling the model into a broader, higher-margin managed platform. Both models are natively multimodal, accepting text and image inputs, and both were trained on Meta's custom infrastructure using NVIDIA H100-80GB GPUs, with a cumulative 7.38 million GPU-hours across the two models (Source: huggingface.co). Meta also previewed, but has not released, **Llama 4 Behemoth**, a roughly 2 trillion-parameter teacher model with 288 billion active parameters used to distill Maverick (Source: ai.meta.com). Meta also states Llama 4 was pre-trained on 200 languages, including over 100 languages with more than 1 billion tokens each, representing "10x more multilingual tokens than Llama 3" (Source: ai.meta.com), and that both released models are "intended for commercial and research use in multiple languages" (Source: deepinfra.com). This report addresses the practical hardware and cost questions that follow from that architecture: how much VRAM Scout and Maverick need at each quantization level, what a multi-node deployment looks like, what API providers charge per token as of July 2026, and how the economics compare to running the models on rented or owned GPU infrastructure.

Self-Hosted Hardware Requirements: VRAM and GPU Configurations

The starting point for any self-hosting decision is total parameter count, not active parameter count, because a mixture-of-experts model must keep every expert's weights in memory even though routing sends each token to only a handful of them. Scout has 16 experts and 109 billion total parameters; Maverick has 128 experts (plus a shared expert) and 400 billion total parameters (Source: ai.meta.com). Meta's own architecture description is explicit on this point: "while all parameters are stored in memory, only a subset of the total parameters are activated while serving these models," which is precisely why VRAM requirements track total parameters rather than the 17-billion active figure (Source: ai.meta.com).

At full BF16 precision, Scout's 109 billion parameters occupy roughly 218 GB and Maverick's 400 billion parameters occupy roughly 800 GB, according to deployment guidance published alongside the model cards (Source: huggingface.co). Quantization is what makes local hosting realistic. Meta's official position is unambiguous about the two supported paths: Scout "is released as BF16 weights, but can fit within a single H100 GPU with on-the-fly int4 quantization," while Maverick "is released as both BF16 and FP8 quantized weights," with "the FP8 quantized weights fit[ting] on a single H100 DGX host while still maintaining quality" (Source: huggingface.co). Meta's consumer-facing model page repeats the Scout claim almost verbatim, describing it as offering "single H100 GPU efficiency, and a 10M context window for seamless long document analysis" (Source: llama.com).

Understanding why a single GPU is not always enough requires a brief detour into how multi-GPU inference actually works. Tensor parallelism, the technique underlying most of the multi-GPU configurations described in this report, splits each individual layer of a model's weights across several GPUs so that no single card needs to hold the entire model, trading a modest amount of inter-GPU communication overhead for the ability to run models that would otherwise not fit in any single accelerator's memory. This differs from data parallelism, which instead replicates the full model on each GPU purely to process more concurrent requests rather than to fit a larger model. Real-world Llama 4 deployments generally combine both approaches: enough tensor parallelism to fit Scout or Maverick's weights, plus additional replicated copies to scale total request throughput once the model itself already fits.

Table 1 below summarizes approximate VRAM requirements by quantization level, drawn from Meta's official model-card guidance and from NVIDIA's H200 and DGX B200 specification pages for the underlying hardware.

CONFIGURATION	APPROXIMATE VRAM NEEDED	TYPICAL HARDWARE	SOURCE BASIS
Scout, BF16 (full precision)	~218 GB	Multi-GPU (2 to 3x H100 80GB, or 2x H200)	Total parameter count at 2 bytes/parameter
Scout, FP8	~110 GB	2x H100 80GB or 1x H200 141GB	Halves BF16 footprint
Scout, Int4 (on-the-fly)	Fits 1x H100 80GB	Single NVIDIA H100	Meta's official quantization guidance (Source: huggingface.co)
Maverick, BF16 (full precision)	~800 GB	Multi-node (2x 8-GPU H100 hosts or more)	Total parameter count at 2 bytes/parameter
Maverick, FP8 (released checkpoint)	Fits 1x H100 DGX host (8x 80GB = 640GB)	Single 8-GPU H100 DGX host	Meta's official quantization guidance (Source: huggingface.co)

The table shows why community estimates for "how much VRAM does Llama 4 need" vary so widely: the answer depends entirely on which quantization level and which model variant is being discussed, and third-party guides do not always specify both. On developer forums, this ambiguity produced confusion at launch; one widely upvoted Reddit thread opens by asking "Zuck just said that Scout is designed to run on a single GPU, but how?" with a top reply clarifying "they said it's for a single H100 GPU" rather than a consumer card (Source: www.reddit.com). Another commenter in the same thread pinpointed the underlying constraint precisely: "you can fit 17B in single GPU but you still need to store all the experts somewhere first" (Source: www.reddit.com).

For context length, Scout's memory requirements scale further because Meta trained and shipped it with an "industry leading 10 million tokens" context window, expanded from the 128,000-token window in Llama 3 (Source: ai.meta.com). Maverick's native context window is 1 million tokens (Source: huggingface.co). The key-value (KV) cache required to hold that much context adds substantially to VRAM on top of the weight footprint, which is why most hosted providers cap the practical context length well below Scout's theoretical 10 million tokens; Together AI's own Scout deployment, for instance, currently supports "327,680 context length (will be increased to 10M)" (Source: www.together.ai). The NVIDIA H200 GPU, with 141 GB of high-bandwidth memory (HBM3e) at 4.8 terabytes per second, is frequently recommended for long-context Llama 4 workloads specifically because "the first GPU to offer 141 gigabytes (GB) of HBM3e memory" gives more headroom for KV cache than the H100's 80 GB (Source: www.nvidia.com), with NVIDIA framing the tradeoff as delivering "better energy efficiency and lower total cost of ownership" for memory-bound workloads of this kind (Source: www.nvidia.com).

Home-lab and community deployments have found that Maverick's MoE structure allows unconventional configurations that would be impossible for an equivalently sized dense model. One documented case study, discussed further below, showed that because "the experts on Maverick work out to only about 3B each," a system can offload the shared, always-active layers to a single consumer GPU while leaving the bulk of the expert weights in ordinary system RAM, since only a few billion parameters need to be evaluated per token even though the full 400 billion must be addressable (Source: www.reddit.com).

Beyond the H100 and H200, RunPod's public pricing lists several adjacent tiers relevant to Llama 4 planning: the NVIDIA H200 (141 GB VRAM) at "$4.39/hr" (Source: www.runpod.io), and the workstation-class NVIDIA RTX Pro 6000 at "$1.99/hr" (Source: www.runpod.io), illustrating that a heavily quantized Scout deployment does not strictly require a data-center-only GPU tier to get started.

Multi-Node and Enterprise-Scale Deployment Architectures

For production inference at scale, both Scout and Maverick can use tensor parallelism across multiple GPUs. Maverick fits on a documented single eight-GPU H100 host; additional nodes are a capacity- and context-dependent scaling choice, not a minimum fit requirement. Red Hat's day-zero integration notes describe how Maverick's FP8 release "com[es] with FP8 weights, enabling the model to fit on a single 8xH100 node, resulting in faster and more efficient inference" when served through vLLM, an open-source inference engine that developers "can install...seamlessly using pip" (Source: developers.redhat.com) (Source: developers.redhat.com). The vLLM project's own deployment guidance is specific about context-length trade-offs at that scale: "Using 8xH100, vLLM can serve Scout with 1M context and Maverick with about 430K" tokens, with example commands setting `--tensor-parallel-size 8` and a `--max-model-len` of 1,000,000 for Scout or 430,000 for the FP8 Maverick checkpoint, and separately recommending the environment variable "VLLM_DISABLE_COMPILE_CACHE=1" when serving either model at these context lengths (Source: vllm.ai) (Source: vllm.ai).

Not every parallelism strategy is supported for every Llama 4 variant. A GitHub feature request tracked by the vLLM maintainers records that pipeline parallelism, an alternative to tensor parallelism that splits a model's layers across nodes rather than splitting each layer across GPUs, initially returned a `NotImplementedError: Pipeline parallelism is not supported for this model` message for the Maverick checkpoint. As one affected developer summarized the experience, "I got this message saying that PP isn't supported" (Source: github.com), with the requester noting plainly that "Llama-4-Maverick-17B-128E is a large LLM that most people will be running across multiple GPU nodes," underscoring that tensor parallelism within a single high-bandwidth node, rather than looser multi-node pipeline splits, was the more mature path at launch (Source: github.com).

At the hardware-generation level, NVIDIA has published the clearest picture of what a fully optimized Maverick deployment looks like on its newest Blackwell architecture. A single NVIDIA DGX B200 node, which packs eight Blackwell GPUs with "1,440 GB total, 64 TB/s HBM3e bandwidth" of aggregate GPU memory (Source: www.nvidia.com), can achieve "over 1,000 tokens per second (TPS) per user on the 400-billion-parameter Llama 4 Maverick model," a result NVIDIA calls a world record and states "was independently measured by the AI benchmarking service Artificial Analysis" (Source: developer.nvidia.com). At its highest-throughput configuration rather than lowest-latency configuration, the same DGX B200 node "reaches 72,000 TPS/server," illustrating the classic trade-off between per-user latency and aggregate server throughput that shapes capacity planning for any multi-tenant Llama 4 deployment (Source: developer.nvidia.com). NVIDIA attributes the low-latency result to a stack of software optimizations including FP8 arithmetic for matrix multiplications, mixture-of-experts routing, and attention operations, plus a speculative-decoding implementation based on EAGLE-3 techniques that contributed "a 4x speed-up relative to the best prior Blackwell baseline" (Source: developer.nvidia.com). NVIDIA's DGX B200 also delivers "3X the training performance and 15X the inference performance" of the prior-generation DGX H100 system, a useful reference point for teams weighing whether to refresh existing H100 clusters before scaling a Maverick deployment (Source: www.nvidia.com).

Enterprises that prefer not to manage bare-metal clusters at all can reach Llama 4 through managed model catalogs on the three major hyperscalers: Amazon Bedrock, Azure AI Foundry, and Google Vertex AI each offer both "Maverick, Scout" variants (Source: www.bitslovers.com), with Azure delivering it through a Model-as-a-Service catalog alongside its OpenAI-exclusive offerings and Google offering it through its Model Garden alongside Gemini and Claude. For most enterprise buyers, this managed-catalog path removes the multi-node hardware question entirely, shifting the decision from "how many GPUs do we need" to "which managed per-token price and service-level agreement fits our workload," a trade-off explored in the next section.

API and Usage-Based Pricing Across Providers

For teams that do not want to own or rent GPU infrastructure directly, Llama 4 is available through numerous inference-as-a-service providers, each pricing input and output tokens separately. *Table 2* below compares confirmed, directly-sourced pricing across the providers examined for this report; figures are current as of July 2026 unless otherwise noted, and this is a fast-moving market where prices and even model availability change without notice.

PROVIDER	LLAMA 4 SCOUT (PER 1M TOKENS)	LLAMA 4 MAVERICK (PER 1M TOKENS)	NOTES
DeepInfra	$0.10 input / $0.30 output (Source: deepinfra.com)	$0.20 input / $0.80 output (Source: deepinfra.com)	Flex tier discounts input/output by 20 to 46 percent
Together AI	$0.18 input / $0.59 output (Source: www.together.ai)	$0.27 input / $0.85 output (Source: www.together.ai)	Serverless, FP8
Groq (launch pricing, April 2025)	$0.11 input / $0.34 output, blended $0.13 (Source: groq.com)	$0.50 input / $0.77 output, blended $0.53 (Source: groq.com)	Not listed on Groq's public pricing page as of July 2026, which currently shows only GPT-OSS, Llama 3.x, and Qwen models plus an enterprise-only tier (Source: groq.com)
Amazon Bedrock	Not separately confirmed	$0.55 input / $0.55 output (reported) (Source: www.bitslovers.com)	Managed catalog, bundled with AWS infrastructure billing
Azure AI Foundry	Not separately confirmed	$0.65 input / $0.65 output (reported) (Source: www.bitslovers.com)	Model-as-a-Service catalog
Google Vertex AI	Not separately confirmed	$0.50 input / $0.50 output (reported) (Source: www.bitslovers.com)	Model Garden
Meta's own cost projection	Not published separately	$0.19/Mtok blended (distributed) to $0.30 to $0.49/Mtok (single host) (Source: llama.com)	Meta's internal engineering estimate, not a hosted commercial offer

At a 3:1 input-to-output mix, the listed Maverick API prices are about $0.35 per million total tokens for DeepInfra, $0.415 for Together AI, and $0.55 for Google Vertex AI. These blended figures weight 75% input tokens and 25% output tokens; Google's published Vertex AI rates are $0.35 for input and $1.15 for output per million tokens. They are a point-in-time comparison of token charges, not a complete comparison of service levels, regional availability, support, or infrastructure integration. Meta's $0.19 to $0.49 per million-token estimate is an engineering projection rather than a commercial API price; Meta states that a single-host deployment specifically "can be served at $0.30 - $0.49/Mtok (3:1 blended)" (Source: llama.com).

Independent benchmarking firm Artificial Analysis, which evaluates models on standardized intelligence and cost metrics, separately measured Maverick's blended market pricing at "$0.27 per 1M input tokens (moderately priced, median: $0.43) and $0.85 per 1M output tokens" as of its most recent evaluation run, a figure that "cost $35.56 to evaluate Llama 4 Maverick on the Intelligence Index" (Source: artificialanalysis.ai). The same evaluation found Maverick generating output at "104.6" tokens per second, describing the model as "notably fast" relative to peers of similar size and price (Source: artificialanalysis.ai). Fireworks AI, one of the API providers evaluated, separately reported that its own Maverick endpoint, running on NVIDIA H200 hardware, delivers "145 tokens per second for streaming throughput," a result it describes as "10-20% faster than the closest competitor and more than double the speed of managed Azure endpoints" according to the same Artificial Analysis measurement methodology (Source: fireworks.ai) (Source: fireworks.ai).

One notable discrepancy worth flagging honestly: Groq publicly announced Llama 4 pricing on its April 2025 launch day at a blended $0.13 for Scout and $0.53 for Maverick, positioning itself as the "lowest cost per token in the industry" at the time (Source: groq.com), yet a direct fetch of Groq's current public pricing page in July 2026 shows no Llama 4 listing among its on-demand models, with large models beyond its standard catalog now routed to an "Enterprise-only" tier (Source: groq.com). This illustrates a broader theme for this hardware guide: provider catalogs rotate frequently, and any point-in-time pricing table should be treated as a snapshot rather than a permanent reference.

Comparative Context and Market Positioning

Llama 4's positioning is best understood against the two dimensions that matter for a hardware-and-cost decision: benchmarked capability and independently measured inference speed. On capability, Artificial Analysis places Maverick at a score of "14 on the Artificial Analysis Intelligence Index, placing it below average among comparable models (median: 17)" as of its most recent evaluation, summarizing the model as "below average

in intelligence, but reasonably priced when comparing to other open weight non-reasoning models of similar size" (Source: artificialanalysis.ai) (Source: artificialanalysis.ai), a notably more measured assessment than Meta's initial marketing claims. Meta's own marketing copy describes Maverick as an "industry-leading natively multimodal model for image and text understanding with groundbreaking intelligence and fast responses at a low cost," a framing that independent benchmarks only partly bear out (Source: llama.com). At launch, Meta had highlighted that an experimental Maverick chat variant scored an "ELO of 1417 on LMArena," a crowdsourced human-preference leaderboard, and that the model was "beating GPT-4o and Gemini 2.0 Flash across a broad range of widely reported benchmarks" (Source: ai.meta.com) (Source: ai.meta.com). That LMArena result later drew scrutiny from the community, since the submitted checkpoint was reportedly tuned differently from the publicly released weights, an important caveat for readers comparing Meta's headline benchmark claims against independent, reproducible evaluations of the shipped model rather than a promotional variant.

On speed, the comparative picture is sharper and less contested because the same 400-billion-parameter Maverick checkpoint was benchmarked identically across hardware vendors by Artificial Analysis. Cerebras reported the fullest cross-vendor comparison: "SambaNova 794 t/s, Amazon 290 t/s, Groq 549 t/s, Google 125 t/s, and Microsoft Azure 54 t/s," against Cerebras' own 2,522 tokens per second and NVIDIA Blackwell's 1,038 tokens per second (Source: www.cerebras.ai) (Source: www.cerebras.ai). That spread, more than 45-fold between the slowest and fastest measured deployment of the identical model weights, is a reminder that "Llama 4 Maverick speed" is inseparable from the specific hardware and serving stack behind any given API endpoint, and buyers should never assume a uniform experience across providers.

Table 1 earlier in this report and the pricing comparison in *Table 2* suggest a practical segmentation of the market. Teams with latency-critical, high-volume workloads should validate measured throughput and availability for the specific endpoint they plan to use. Teams optimizing for token cost should compare current input and output prices using their expected input-to-output ratio; the listed 3:1 Maverick examples in Table 2 range from about $0.35 to $0.55 per million total tokens, based on the providers with published rates in the table. Teams prioritizing cloud integration or procurement consistency should obtain a current provider quote and evaluate it alongside their operational requirements. Dedicated self-hosting may be worth evaluating for strict data-residency, air-gapped, or very high sustained-volume workloads, but requires a workload-specific capacity and operations analysis. Procurement teams evaluating these segments should also weight how quickly a given provider's catalog and pricing have historically changed, since a vendor's cheapest published rate on the day of evaluation is not a guarantee of its rate a quarter later.

Data Analysis and Evidence

Bringing together the pricing and performance data collected for this report, several quantitative patterns stand out. First, on raw GPU rental cost, a single NVIDIA H100 currently rents for between $2.89 and $4.29 per hour depending on interconnect (PCIe versus SXM) and provider, with Lambda listing $3.29 to $4.29 per hour depending on instance size (Source: lambda.ai). At the high end of the hardware curve, a single Blackwell B200 (180 GB VRAM) rents for roughly $6.69 to $6.99 per hour on Lambda depending on configuration, while an eight-GPU HGX B200 cluster on Lambda's reserved 1-click clusters starts near $9.86 per GPU-hour for a 16-GPU commitment and falls to $8.87 per GPU-hour at 256-plus GPUs (Source: lambda.ai). NVIDIA frames its own newest DGX platform in similar terms, noting that "Powered by the NVIDIA Blackwell architecture's advancements in computing, DGX B200 delivers 3X the training performance and 15X the inference performance of DGX H100" (Source: www.nvidia.com).

Using the stated on-demand H100 range of $2.89 to $4.29 per hour, a continuously rented single H100 for 730 hours costs about $2,110 to $3,132 per month before storage, networking, or utilization losses. An eight-GPU H100 host for the FP8 Maverick checkpoint would cost about $16,878 to $25,054 per month at the same per-GPU rates, before volume discounts. For Scout, DeepInfra's standard $0.10-per-million input and $0.30-per-million output rates yield $0.15 per million total tokens at a 3:1 input-to-output mix. Comparing only those token charges with the single-H100 rental range yields roughly 14.1 to 20.9 billion tokens per 30-day month, or 469 to 696 million tokens per day; for all-output traffic, the comparable range is about 234 to 348 million tokens per day. This hardware-only comparison excludes storage, networking, engineering, and utilization costs, so dedicated-hardware economics require a workload-specific capacity and operations analysis.

Second, on training-side compute, Meta disclosed that Llama 4 pre-training consumed a combined "7.38M" GPU-hours on H100-80GB hardware, broken out as 5.0 million GPU-hours for Scout and 2.38 million GPU-hours for Maverick, alongside "1,999 tons" of estimated location-based greenhouse gas emissions for training, offset to zero on a market basis through Meta's renewable energy purchases (Source: huggingface.co) (Source: deepinfra.com). Scout was pretrained on approximately 40 trillion tokens and Maverick on approximately 22 trillion tokens of multimodal data (Source: deepinfra.com), against a total pre-training data mixture Meta describes as "more than 30 trillion tokens" across the family when accounting for shared and distinct portions of the corpus (Source: ai.meta.com). During the Behemoth teacher-model training run, Meta reported achieving "390 TFLOPs/GPU" using FP8 precision across a 32,000-GPU cluster, a data point that illustrates the scale of infrastructure behind the smaller, publicly released Scout and Maverick checkpoints (Source: ai.meta.com).

Third, on inference throughput, the cross-vendor benchmark data compiled by Cerebras shows a wide dispersion even among well-resourced hyperscalers running the identical Maverick weights: Google Cloud's measured 125 tokens per second and Microsoft Azure's 54 tokens per second sit roughly an order of magnitude below Cerebras' 2,522 tokens per second and NVIDIA Blackwell's 1,038 tokens per second on the same benchmark (Source: www.cerebras.ai). Cerebras CEO Andrew Feldman framed the stakes of this gap directly, stating that on GPU-bound deployments "generation speeds as low as 100 tokens per second on GPUs" cause "wait times of minutes and mak[e] production deployment impractical" for agentic and reasoning workloads that require many sequential model calls (Source: www.cerebras.ai).

Fourth, on licensing economics, Meta's Llama 4 Community License Agreement caps free commercial use at scale: any licensee whose products or services exceed "700 million monthly active users in the preceding calendar month" must separately request a license "which Meta may grant to you in its sole discretion" (Source: llama.com). The full terms are published as "a custom commercial license, the Llama 4 Community License Agreement" alongside each model card rather than buried in a separate legal repository (Source: deepinfra.com). For the overwhelming majority of enterprises evaluating Llama 4 hardware requirements, the 700-million-user threshold is irrelevant since it targets only the handful of platforms operating at hyperscale consumer-internet scale.

Case Studies and Real-World Examples

Meta's own consumer product surface. The largest known deployment of Llama 4 is Meta's own family of consumer applications. On launch day, Meta directed users to "Try Meta AI built with Llama 4 in WhatsApp, Messenger, Instagram Direct, and on the web," putting a mixture-of-experts model with hundreds of billions of total parameters in front of a user base numbering in the billions (Source: ai.meta.com). Running that scale of inference is precisely the deployment context the MoE architecture was designed for: Meta's own architecture notes state this design "improves inference efficiency by lowering model serving costs and latency," which is the stated rationale for choosing a 400-billion-parameter sparse model over a smaller, fully-dense alternative for a product used by so many people simultaneously (Source: ai.meta.com). For a hardware planner, this case illustrates the upper bound of the scale question: Meta's own production stack is effectively the largest single reference deployment of Maverick anywhere, run across infrastructure well beyond what any individual enterprise customer would provision.

Cerebras Systems' wafer-scale inference cloud. Cerebras Systems, a chip startup building wafer-scale processors as an alternative to GPU clusters, added Llama 4 Maverick and Scout to its inference cloud shortly after launch. By May 2025, Cerebras announced that "at over 2,500 t/s, Cerebras has set a world record for LLM inference speed on the 400B parameter Llama 4 Maverick model, the largest and most powerful in the Llama 4 family" (Source: www.cerebras.ai), with co-founder and chief executive Andrew Feldman stating the company had "led the charge in redefining inference performance across models like Llama, DeepSeek, and Qwen, regularly delivering over 2,500 TPS/user" (Source: www.cerebras.ai). Notably, Cerebras' press release also points out an operational nuance relevant to any buyer comparing published benchmarks: "unlike the Nvidia Blackwell used in the Artificial Analysis benchmark, the Cerebras hardware and API are available now," and it goes on to suggest that NVIDIA's headline figure may have required an unusually small batch size to achieve, since "none of the Nvidia's inference providers offer a service at Nvidia's published performance" (Source: www.cerebras.ai), an example of the vendor rivalry that colors much of the public Llama 4 benchmarking discourse. Cerebras added that its result required no special kernel optimizations and that comparable performance "will be available to everyone through Meta's API service coming soon" (Source: www.cerebras.ai), a reason to weight independently reproduced numbers over any single vendor's press materials.

NVIDIA's Blackwell DGX B200 latency record. NVIDIA used Llama 4 Maverick as its flagship demonstration model for the Blackwell architecture's inference capabilities, engineering a stack of kernel fusions, FP8 arithmetic, and EAGLE-3-based speculative decoding specifically to minimize per-user latency on the 400-billion-parameter model, described in its technical writeup as helping "even massive models...deliver the speed and responsiveness required for seamless, real-time user experiences and complex AI agent deployment scenario[s]" (Source: developer.nvidia.com). NVIDIA published a companion deployment guide on GitHub so that any organization with access to B200 hardware could replicate the configuration itself (Source: developer.nvidia.com). Unlike the Meta and Cerebras cases, this deployment is best read as a capability demonstration rather than a production service: it shows the ceiling of what optimized Blackwell hardware can achieve on Maverick, which any team purchasing or renting B200 capacity should treat as a target to approach through its own optimization work rather than a result guaranteed out of the box.

Hyperscaler managed-service adoption. Rather than each building bespoke infrastructure, Amazon Web Services, Microsoft Azure, and Google Cloud each added both Llama 4 Scout and Maverick to their respective managed model catalogs, with Amazon Bedrock offering the pair alongside Anthropic's Claude with "enterprise indemnification" (Source: www.bitslovers.com), Azure AI Foundry offering Llama 4 through its Model-as-a-Service catalog as a complement to its OpenAI-exclusive models, and Google Vertex AI offering it through Model Garden alongside its own Gemini family. This three-way availability signals that, roughly a year after launch, Llama 4 had become table-stakes infrastructure across the major clouds rather than a niche open-weight curiosity.

Community and home-lab self-hosting. At the opposite end of the deployment spectrum, individual developers demonstrated that Maverick's sparse activation pattern makes even a repurposed, decade-old server usable. One widely discussed case documented running the full 400-billion-parameter Maverick model on a homemade rig built from "dual e5-2690 v2 ($10/each), random Supermicro board ($30), 256GB of DDR3 Rdimms ($80)" plus a consumer GPU, achieving 24.53 tokens per second once a single NVIDIA V100 handled the model's always-active shared layers while the CPU and system RAM handled the sparsely activated experts (Source: www.reddit.com) (Source: www.reddit.com). Even without any GPU at all, the same hardware produced a usable, if slow, 3.10 tokens per second in CPU-only mode, and other commenters in the same thread called the result "pretty amazing" for "a 12 year old system without a gpu" (Source: www.reddit.com), illustrating just how far Maverick's sparse mixture-of-experts design lowers the floor for what counts as "runnable" hardware compared with an equivalently sized dense model **(the CPU-only and consumer-GPU figures above describe an individual hobbyist's documented setup, not a vendor-certified reference configuration)**.

Implications and Future Directions

The clearest implication of Llama 4's mixture-of-experts design is that "hardware requirements" can no longer be answered with a single VRAM number per model; the answer depends on quantization level, target context length, and whether the deployment optimizes for latency or throughput. This is a durable shift, not a temporary quirk of the Llama 4 release: Meta's own architecture notes frame MoE adoption as a deliberate compute-efficiency choice for future models, not a one-off experiment, since "MoE architectures are more compute efficient for training and inference and, given a fixed training FLOPs budget, deliver[] higher quality compared to a dense model" (Source: ai.meta.com). Teams building internal hardware-sizing playbooks should expect this total-versus-active parameter distinction to recur in future model generations, including the still-training, roughly 2 trillion-parameter Llama 4 Behemoth, whose eventual release would raise the total-parameter VRAM floor by an order of magnitude again (Source: ai.meta.com). Any capacity plan built around today's Scout or Maverick VRAM figures should therefore be documented with its assumptions explicit, so it can be revisited quickly once a larger or differently architected successor model arrives.

A second implication concerns the widening gap between GPU-based and non-GPU inference hardware. The Cerebras-versus-NVIDIA benchmark dispute, in which Cerebras measured its own wafer-scale hardware at more than double NVIDIA's best publicly demonstrated Blackwell configuration on the identical Maverick weights, suggests that the inference-hardware market is no longer a single-axis GPU-generation race; specialized

accelerators are now credibly competitive on tokens-per-second for MoE-style architectures specifically, because their memory-bandwidth-heavy designs suit the sparse expert-routing pattern well. Buyers evaluating "best GPU for Llama 4" in isolation risk missing that some of the fastest measured deployments of the model use no conventional GPU at all.

Third, hardware planning cannot be separated from safety tooling in production deployments. Meta recommends that any Llama 4 deployment pair the base model with system-level guardrails "like Llama Guard, Prompt Guard and Code Shield" (Source: huggingface.co), which adds modest additional compute and memory overhead on top of the base model sizing discussed throughout this report and should be included in any production capacity plan rather than treated as an afterthought.

Fourth, the discrepancy between Groq's headline April 2025 launch pricing and its absence from the public pricing page fetched for this report in July 2026 is a useful cautionary signal for procurement teams: per-token API pricing for open-weight models is unusually volatile, with providers adding, repricing, and occasionally removing models from self-serve catalogs as their own capacity and margin priorities shift. Any organization committing to a specific provider for Llama 4 inference should build in periodic repricing reviews rather than assuming launch-day figures remain valid a year or more later.

Finally, the community-documented home-lab deployments suggest that the accessibility gap between "frontier-class" and "hobbyist-runnable" continues to narrow for MoE architectures specifically, even as it widens for dense models of comparable total scale. This has commercial implications for edge and on-premises deployment strategies: a regulated enterprise unable to send data to an external API may find a quantized Scout deployment on a single H100, or even a well-specified workstation, considerably more attainable than the raw 109-billion parameter count would suggest at first glance.

Frequently Asked Questions (FAQs)

How much VRAM does Llama 4 need? It depends on the variant and quantization level. Llama 4 Scout needs roughly 218 GB at full BF16 precision, or fits a single 80 GB NVIDIA H100 GPU with on-the-fly Int4 quantization, per Meta's own model documentation (Source: huggingface.co). Llama 4 Maverick needs roughly 800 GB at full BF16 precision, or fits a single eight-GPU H100 DGX host (640 GB combined) using the officially released FP8 checkpoint (Source: huggingface.co).

What are Llama 4's GPU requirements in practice? Meta says Scout can fit on a single NVIDIA H100 with Int4 quantization (Source: ai.meta.com); this is a fit statement, not a published minimum or support policy. The usable configuration also depends on the selected runtime, context length, and KV-cache headroom. For Maverick, Meta's released FP8 checkpoint fits on a single eight-GPU H100 host; vLLM documents 8x H100 configurations serving both models with tensor parallelism set to 8 (Source: vllm.ai).

What is the cost to run Llama 4? Self-hosting a single H100 costs roughly $2.89 to $4.29 per hour on demand (Source: www.runpod.io) (Source: lambda.ai), or about $2,110 to $3,132 per month for 730 hours; an eight-GPU host scales that roughly eightfold before discounts. Via API, the listed Maverick rates range from $0.20/$0.80 at DeepInfra (input/output per million tokens) to $0.35/$1.15 on Google Vertex AI (Source: deepinfra.com) (Source: cloud.google.com).

What is Llama 4 Maverick's inference cost per token? Meta's own engineering estimate projects "$0.19/Mtok (3:1 blended)" for distributed inference, rising to "$0.30 - $0.49/Mtok (3:1 blended)" when served from a single host rather than distributed across nodes (Source: llama.com). At a 3:1 input-to-output mix, Together AI's listed $0.27 input and $0.85 output rates equal about $0.415 per million total tokens, while Google Vertex AI's listed $0.35 input and $1.15 output rates equal about $0.55 (Source: www.together.ai) (Source: cloud.google.com).

What are Llama 4 Maverick's specific VRAM requirements? Maverick holds 400 billion total parameters and 17 billion active parameters across 128 experts (Source: ai.meta.com), which places its unquantized VRAM footprint near 800 GB and its officially released FP8 footprint at roughly the 640 GB available on a single eight-GPU H100 DGX host (Source: huggingface.co).

What are Llama 4 Scout's hardware requirements? Scout holds 109 billion total parameters and 17 billion active parameters across 16 experts, with a 10-million-token context window (Source: ai.meta.com) (Source: ai.meta.com). Meta specifies a single NVIDIA H100 GPU as the minimum supported hardware using on-the-fly Int4 quantization (Source: llama.com), though the full 10-million-token context requires far more memory for the KV cache than the weights alone, which is why most hosted providers cap the practically usable context well below that theoretical maximum (Source: www.together.ai).

How does Llama 4's hardware profile compare to Llama 3? The shift to a mixture-of-experts architecture is the key difference: Llama 3's largest dense models concentrated all of their parameters into active compute, whereas Llama 4 spreads far more total parameters across experts that are only partially activated per token, and expands the context window from Llama 3's 128,000 tokens to Scout's 10 million tokens (Source: ai.meta.com).

In practical terms, this means a hardware plan built for a dense Llama 3 model of similar active-parameter size cannot simply be reused for Llama 4: the total-parameter VRAM floor is substantially higher even though generation speed per token is broadly comparable, since both generations activate a similar order of magnitude of parameters on any given forward pass.

What is the best GPU for Llama 4? For single-GPU Scout deployments, the NVIDIA H100 80GB is Meta's officially cited baseline (Source: ai.meta.com), while the H200's 141 GB of HBM3e memory offers more headroom for longer context windows on the same architecture (Source: www.nvidia.com). For latency-critical Maverick deployments at scale, NVIDIA's own Blackwell-based DGX B200 posts the fastest GPU-based result independently confirmed at over 1,000 tokens per second per user (Source: developer.nvidia.com), though Cerebras' non-GPU wafer-scale hardware measured more than double that figure on the same model (Source: www.cerebras.ai).

Can Llama 4 run on a multi-node setup? Yes. vLLM's official guidance describes an 8xH100 configuration using tensor-parallel-size 8 as the standard single-node topology for both models, and confirms that scaling Maverick specifically means "most people will be running across multiple GPU nodes" for production-grade throughput and context length (Source: github.com), while pipeline parallelism, an alternative multi-node splitting strategy, was not supported for the Maverick checkpoint at the time of that report (Source: github.com).

Conclusion

Llama 4's hardware and cost profile cannot be summarized with a single number, because its mixture-of-experts design deliberately decouples the parameter count that governs compute speed from the parameter count that governs memory footprint. Scout's 109 billion total parameters and Maverick's 400 billion total parameters, not their shared 17 billion active parameters, are what determine whether a deployment needs a single GPU, a full eight-GPU host, or a multi-node cluster. Meta built quantized checkpoints specifically to make single-GPU and single-host deployment realistic, and the ecosystem responded quickly: Llama 4 reached managed cloud catalogs, specialized inference clouds, general-purpose API providers such as DeepInfra, and even hobbyist hardware repurposed from a decade-old server.

For budgeting purposes, the practical range this report has documented spans from roughly $0.10 per million input tokens on the cheapest API tier to over a thousand dollars per month for dedicated single-GPU rental, and up to eight times that for a full DGX-class host, with throughput varying by more than 45-fold across vendors serving the identical Maverick weights. Buyers should treat every specific price and benchmark figure in this space as a snapshot rather than a constant, verify current catalog availability before committing budget, and match their choice of hardware or provider to whether their priority is lowest cost per token, lowest latency per user, or organizational fit with an existing cloud relationship. The underlying architectural lesson, that total parameters govern memory while active parameters govern compute, will very likely persist into whatever comes after Llama 4, including Meta's still-unreleased, far larger Behemoth model, and should anchor any hardware budget built today against tomorrow's larger and more capable successors.

Tags: llama 4 hardware requirements, llama 4 vram, llama 4 gpu requirements, llama 4 maverick, llama 4 scout, llama 4 inference cost, multi-node gpu setup, nvidia h100 h200

DISCLAIMER

The information contained in this document is provided for educational and informational purposes only. We make no representations or warranties of any kind, express or implied, about the completeness, accuracy, reliability, suitability, or availability of the information contained herein.

Any reliance you place on such information is strictly at your own risk. In no event will GPUSmith or its representatives be liable for any loss or damage including without limitation, indirect or consequential loss or damage, or any loss or damage whatsoever arising from the use of information presented in this document.

This document may contain content generated with the assistance of artificial intelligence technologies. AI-generated content may contain errors, omissions, or inaccuracies. Readers are advised to independently verify any critical information before acting upon it.

All product names, logos, brands, trademarks, and registered trademarks mentioned in this document are the property of their respective owners. All company, product, and service names used in this document are for identification purposes only. Use of these names, logos, trademarks, and brands does not imply endorsement by the respective trademark holders.

This document does not constitute professional or legal advice. For specific guidance related to your business needs, please consult with appropriate qualified professionals.