LLM Inference Hardware Sizing: Enterprise GPU Guide 2026

Published July 10, 2026 41 min read

Executive Summary

Sizing hardware for large language model (LLM) inference comes down to a single dominant constraint: graphics processing unit (GPU) memory, commonly called video random-access memory (VRAM). The baseline formula used across the industry is Memory (GB) = Parameters (billions) x Bytes per parameter x (1 + Overhead), where overhead of 20 to 50% covers the key-value (KV) cache, activations, and framework buffers (Source: modal.com) (Source: bentoml.com). Applied to a **70 billion (B)** parameter model such as Llama 3.1 70B at 16-bit precision, this yields roughly **140 GB** of weights alone, meaning a single **80 GB NVIDIA H100** is insufficient and at least two are required (Source: modal.com). As of July 2026, the three dominant accelerators for enterprise inference are the **NVIDIA H100** (80 GB HBM3, 3.35 TB/s bandwidth), the **NVIDIA H200** (141 GB HBM3e, 4.8 TB/s bandwidth, roughly $30,000 to $40,000 per card, or $3.72 to $10.60 per GPU-hour to rent as of January 2026) (Source: nvidia.com (Source: jarvislabs.ai), and the **NVIDIA GB200 NVL72** rack-scale Blackwell system, which delivers up to **3.4 times** higher per-GPU throughput than an eight-GPU H200 system on the Llama 3.1 405B MLPerf benchmark (Source: developer.nvidia.com). AMD's competing **Instinct MI300X**, built on an 8,192-bit memory bus interface running at a 5.2 gigabit-per-second (Gbps) data rate (Source: amd.com), offers 192 GB of HBM3 at 5.3 TB/s bandwidth for an estimated $10,000 to $15,000 per unit, undercutting the H200 on sticker price at the cost of a less mature CUDA-equivalent software ecosystem (Source: amd.com (Source: spheron.network).

Inference workloads split into two computational phases with opposite bottlenecks: prefill, which processes the prompt and is compute-bound, and decode, which generates tokens one at a time and is memory-bandwidth-bound because every generated token requires re-streaming the full weight set from high-bandwidth memory (HBM) (Source: rfriedmann.de). This means memory bandwidth, not raw teraFLOPS, sets the practical ceiling on tokens-per-second for most production chat and agentic workloads, and it is the central reason the industry's roadmap (H200, B200, and beyond) prioritizes bandwidth growth over compute growth (Source: arxiv.org). Open-source serving engines such as **vLLM**, using its PagedAttention algorithm, and **NVIDIA TensorRT-LLM** both exploit this by minimizing KV cache waste and maximizing concurrent-request batching; the original PagedAttention paper reported throughput improvements of 2 to 4 times over prior systems at the same latency (Source: arxiv.org).

Quantization is the primary lever for shrinking hardware requirements: dropping from FP16 (16-bit) to FP8 halves weight memory, while 4-bit formats such as AWQ and GPTQ cut memory by roughly 75% with a larger, though often still acceptable, accuracy trade-off (see Table 2 below). For context, a frontier mixture-of-experts (MoE) model like DeepSeek V3 (671B total parameters) needs 671 GB of VRAM at FP8, fitting on a single eight-GPU H200 node (1,128 GB total) but not comfortably on an eight-GPU H100 node (640 GB total) without aggressive quantization (see the On-Premise Hardware Requirements section below).

On the economics side, cloud GPU rental rates vary enormously by provider: as of July 2026, on-demand H200 pricing ranges from $2.40 per GPU-hour on specialist marketplaces to $10.60 per GPU-hour on Azure, a more than fourfold spread for identical silicon (Source: thundercompute.com). Enterprises weighing on-premises deployment against cloud consumption have reported inference cost reductions of up to 62% versus public cloud LLM endpoints after migrating to owned infrastructure, alongside hardware payback periods as short as under four months for mid-size deployments (documented in the Case Studies section below). At the macro level, NVIDIA reported record Data Center revenue of $75.2 billion for the first quarter of fiscal 2027 (ended April 26, 2026), up 92% year over year, underscoring the scale of enterprise capital flowing into inference and training hardware (Source: nvidianews.nvidia.com), while IDC projects worldwide AI infrastructure spending will reach $487 billion in 2026 and surpass $1 trillion by 2029 (Source: idc.com). This report walks through the sizing formulas, the current GPU landscape, cluster networking, serving-engine mechanics, real-world deployment costs, and the benchmarks needed to make a defensible hardware decision for production LLM inference.

Introduction and Background

Enterprises evaluating a self-hosted large language model (LLM) deployment face a deceptively simple question with a genuinely complex answer: how much hardware does this actually require? The stakes are high. Undersizing a GPU cluster produces out-of-memory (OOM) crashes, throttled context windows, and unusable concurrency; oversizing wastes capital that, at data-center GPU prices that reach into the tens of thousands of dollars per accelerator (detailed in the Enterprise GPU Sizing Guide section below), can run into the millions across a full production cluster. Unlike training, where hardware requirements are somewhat forgiving of batch-size and checkpoint adjustments, inference sizing must account for unpredictable, bursty concurrent user traffic against a fixed memory budget that is consumed simultaneously by model weights and by a KV cache that grows and shrinks with every request (Source: bentoml.com).

This guide addresses **LLM inference hardware sizing** end to end: how to calculate VRAM requirements, what GPU options exist for on-premises and cloud deployment, how memory bandwidth (not just capacity) governs real-world throughput, how serving engines such as **vLLM** and **NVIDIA TensorRT-LLM** change the sizing math, and what enterprises have actually paid and saved by building their own inference clusters. As of July 2026, the hardware landscape includes the **NVIDIA Hopper** generation (H100, H200), the newer **Blackwell** generation (B200, GB200 NVL72), AMD's **Instinct MI300X**, and Intel's **Gaudi 3**, positioned as a lower-cost alternative to NVIDIA's data-center lineup. Model architectures have also shifted: many current-generation frontier models, including DeepSeek V3/R1, Kimi K2.5, GLM-5, and Llama 4, use mixture-of-experts (MoE) designs where the total parameter count (which determines memory footprint) is substantially larger than the active parameter count used per token (which governs compute cost), fundamentally changing how sizing decisions should be made (discussed further in the Implications section below).

The core sizing challenge has three interacting variables: model weight memory (a fixed cost set by parameter count and numeric precision), KV cache memory (a variable cost that scales with context length, batch size, and concurrent users), and memory bandwidth (which sets the achievable tokens-per-second ceiling for autoregressive decoding) (Source: pkhamdee.blog). Getting these three variables right, and understanding how quantization, tensor parallelism, and serving-engine choice interact with each, is the difference between a right-sized cluster and an expensive guess. The sections that follow work through each variable systematically, drawing on official vendor specifications, peer-reviewed and preprint research, benchmark consortium data, and named enterprise deployments to ground every recommendation in verifiable figures rather than rules of thumb alone.

A further complication specific to 2026-era sizing is that "the model" is rarely a single fixed artifact. Enterprises increasingly run several models concurrently on shared infrastructure: a large reasoning model for complex tasks, a smaller distilled model for routine queries, an embedding model for retrieval-augmented generation (RAG), and often a speech-to-text model such as Whisper for transcription workloads, as documented in several of the enterprise deployments examined later in this report. Each of these workloads has a distinct memory and bandwidth profile, and sizing decisions that only account for the largest model in isolation routinely underestimate the aggregate hardware footprint once multi-model serving, model swapping, and burst concurrency are taken into account. This report treats sizing as a portfolio exercise across the full inference workload, not a single-model calculation, reflecting how production deployments actually operate.

How to Calculate VRAM for LLM Inference

The starting point for any hardware sizing exercise is the memory footprint of the model weights themselves. The generally accepted formula, used consistently across infrastructure vendors and independent calculators, is:

Memory (GB) = Parameters (billions) x (Bits per parameter / 8) x (1 + Overhead percentage)

As one independent engineering writeup summarizes the weight term specifically: "Bytes per parameter is set by precision," so an 8B model at FP16 (2 bytes per parameter) requires 8 x 2 = 16 GB purely for weights, before any KV cache or overhead is added (Source: pkhamdee.blog).

Where the "Overhead percentage" (commonly 10 to 30%, though some guides recommend 20 to 50% for production workloads) captures the KV cache, activation buffers, workspace memory, and framework/runtime reservations that sit on top of the raw weights (Source: bentoml.com). Modal's engineering team frames the same relationship as a quick rule of thumb: approximately **2 GB of GPU memory per 1 billion parameters** for models loaded in 16-bit half precision, which for a 70B-parameter model like Llama 3.1 70B works out to 70 x 2 GB = 140 GB, exceeding a single 80 GB H100 and requiring at least two GPUs (Source: modal.com (Source: modal.com).

The KV cache term deserves separate attention because it is the component most likely to be underestimated. VMware's private-AI engineering team derives a per-token KV cache size from a model's layer count, hidden dimension, and number of key-value heads (accounting for grouped-query attention, or GQA, which shares KV heads across multiple query heads to shrink the cache), then multiplies that per-token cost by average context window length and number of concurrent requests. For Llama 3-8B with an 8,192-token average context and 10 concurrent requests, the total memory footprint works out to roughly **26 GB**, computed as 8B params x 2 bytes (weights) plus 0.000122 GiB/token x 8,192 tokens x 10 requests (KV cache) (Source: blogs.vmware.com). Because all state-of-the-art LLMs are memory-bound during decode, batching more prompts into the KV cache is a primary lever for improving throughput, and the VMware team's worked example shows a Llama-3-8B model on a single 80 GB A100 or H100 can theoretically hold up to **262,144 tokens** in its KV cache once model weights are subtracted from total capacity (Source: blogs.vmware.com).

A useful single-line simplification comes from independent engineering writeups: Total VRAM = (Model Weights + KV Cache) / 0.9, where dividing by 0.9 accounts for the roughly 10% of card capacity that serving engines like vLLM reserve for CUDA kernels, the scheduler, and memory fragmentation headroom (Source: pkhamdee.blog). For enterprises sizing frontier mixture-of-experts (MoE) models, the practical rule from infrastructure specialists is to estimate VRAM as total model parameters (not active parameters) multiplied by bytes-per-parameter, then add 30 to 50% overhead: a 70B dense model comfortably fits a single eight-GPU H100 or H200 node, while a 671B-parameter MoE model like DeepSeek V3 requires the full memory of an eight-GPU H200 node once overhead is included (see Table 3 below). This distinction between total and active parameters is critical: MoE models are far more deployable than their headline parameter counts suggest because only a fraction of the network activates per forward pass, but the entire parameter set must still be resident in VRAM, a point developed further in the Implications section.

GPU Requirements for LLM Inference: Hardware Landscape

Selecting the right accelerator requires comparing memory capacity, memory bandwidth, and compute throughput simultaneously, since inference workloads can be bottlenecked by any of the three depending on batch size and context length. *Table 1* below summarizes the specifications of the accelerators most commonly deployed for enterprise LLM inference as of mid-2026.

Table 1: Data Center GPU Specifications for LLM Inference (as of July 2026)

GPU	MEMORY (VRAM)	MEMORY BANDWIDTH	FP8 COMPUTE	APPROX. PRICE
NVIDIA H100 SXM	80 GB HBM3	3.35 TB/s	3,958 TFLOPS	~$25,000 to $30,000/GPU
NVIDIA H100 NVL	94 GB HBM3	3.9 TB/s	3,341 TFLOPS	Varies by OEM
NVIDIA H200 SXM	141 GB HBM3e	4.8 TB/s	3,958 TFLOPS	$30,000 to $40,000/GPU (Source: jarvislabs.ai)
AMD Instinct MI300X	192 GB HBM3	5.3 TB/s	2,615 TFLOPS	$10,000 to $15,000/GPU (Source: spheron.network)
NVIDIA B200 (HGX 8-GPU)	192 GB HBM3e/GPU	Not separately disclosed by NVIDIA	Up to 2x H200 FP8 throughput	$68.80/hr for 8-GPU cloud node (Source: coreweave.com)
NVIDIA GB200 NVL72 (per GPU)	186 GB HBM3e/GPU (13.4 TB total, 72 GPUs)	576 TB/s aggregate	N/A (rack-scale)	$42.00/hr cloud rack (Source: coreweave.com)

Sources: NVIDIA H100 and H200 official product pages (Source: nvidia.com (Source: nvidia.com), AMD MI300X official specification page (Source: amd.com), NVIDIA GB200 NVL72 product page (Source: nvidia.com), and CoreWeave's public pricing page for cloud rental figures.

The table illustrates a clear pattern: newer generations trade higher absolute price for disproportionately higher memory capacity and bandwidth per dollar of compute. The **H200** offers 76% more memory and 43% more bandwidth than the H100 at a comparable list price band (Source: nvidia.com), a capacity and bandwidth advantage that translates directly into higher achievable throughput for memory-bound decode workloads. The **AMD MI300X** stands out on raw specification: 2.4 times the memory capacity and 1.6 times the peak theoretical memory bandwidth of "competitive accelerators," which AMD's own materials benchmark against the H100 (Source: amd.com). However, the CUDA software ecosystem's maturity advantage means MI300X deployments frequently carry higher integration costs that partly offset the hardware price gap, a trade-off enterprises should model explicitly before committing to an AMD-based cluster (Source: spheron.network). Independent hardware testing corroborates the memory specifications: the MI300X packs eight compute dies, each with 38 CDNA 3 compute units, atop a 256 MB Infinity Cache, fed by eight HBM3 stacks that together provide 5.3 TB/s of bandwidth and 192 GB of capacity (Source: chipsandcheese.com). The same reviewers frame the generational shift plainly: AMD's prior-generation MI210 "generally fell short of NVIDIA's H100 PCIe," while with the MI300X, for the first time, AMD built a compute-focused GPU physically larger and more capable than its NVIDIA counterpart (Source: chipsandcheese.com). The same independent testing found AMD's ROCm software stack, while much improved from the prior CDNA 2 generation, still lacks full support for common inference kernels such as flash-attention on bfloat16, underscoring the software-maturity gap enterprises should weigh alongside the price advantage (Source: chipsandcheese.com).

Below the flagship data-center tier, a set of lower-cost workstation and inference-optimized cards remains common for smaller deployments and entry-level self-hosting. VMware's private-AI engineering documentation lists the **NVIDIA A10** at 24 GB of memory and 600 GB/s of bandwidth, the **NVIDIA A30** at 24 GB and 933 GB/s, and the **NVIDIA L40** at 48 GB and 864 GB/s, positioning these cards well below the H100/H200 tier on both capacity and bandwidth but at a correspondingly lower price point (Source: blogs.vmware.com (Source: blogs.vmware.com). These cards typically serve small dense models (up to roughly 13B parameters) or act as inference nodes for embedding and transcription models alongside a larger primary serving GPU, a pattern visible in the German engineering firm case study discussed later in this report, which used four L40S cards (a 48 GB, 864 GB/s variant of the L40 optimized for inference) rather than H100s specifically to control cost for a serving-only workload.

At the rack-scale end, NVIDIA's **GB200 NVL72** links 72 Blackwell GPUs and 36 Grace CPUs into a single NVLink domain delivering 130 TB/s of NVLink bandwidth and 13.4 TB of aggregate HBM3e memory at 576 TB/s bandwidth (Source: nvidia.com (Source: nvidia.com). This architecture is purpose-built for trillion-parameter model inference where a single node's memory would otherwise be insufficient. NVIDIA's own marketing for the H200 frames the memory upgrade explicitly around total cost of ownership (TCO), stating that the card's larger and faster memory "accelerates generative AI and LLMs, while advancing scientific computing for HPC workloads with better energy efficiency and lower total cost of ownership"

(Source: nvidia.com), a framing worth treating as a vendor claim rather than an independently verified TCO figure. Intel's **Gaudi 3** occupies a lower-cost niche in the market, positioned as a value alternative to NVIDIA's Hopper and Blackwell lineup, though its software ecosystem remains less mature than NVIDIA's CUDA and TensorRT-LLM stack, a maturity gap enterprises should weigh against its price advantage before standardizing on it for production serving.

Beyond raw specification sheets, form factor matters for procurement planning. Both the H100 and H200 ship in **SXM** (a proprietary NVIDIA socket used in HGX server boards, offering full NVLink bandwidth) and **PCIe** (a standard slot-based form factor with no dedicated NVLink fabric) variants. Thunder Compute's pricing analysis notes that the H200 SXM and H200 NVL both carry 141 GB of HBM3e at 4.8 TB/s, but the NVL PCIe variant runs at 600 W with only two-to-four-way NVLink bridges, yielding roughly 18% lower multi-GPU throughput than the 700 W SXM form factor with full NVLink 4.0 at 900 GB/s (Source: thundercompute.com). For single-GPU or cost-sensitive deployments where multi-GPU tensor parallelism is not required, PCIe cards remain a reasonable choice; for any deployment requiring tensor parallelism across more than two GPUs, the SXM/HGX form factor is generally the better investment despite its higher power draw and server-platform requirements.

LLM Inference Memory Bandwidth and the Roofline Model

Independent technical writeups on inference performance describe the underlying picture succinctly: "People quote 'tokens per second' as if an engine has one speed. It has two, and they're governed by opposite limits," depending on whether the workload sits in prefill or decode (Source: rfriedmann.de). Memory capacity determines whether a model fits on a GPU at all; memory bandwidth determines how fast it runs once it fits. This distinction is formalized by the roofline model, a performance analysis framework in which a kernel's achievable speed is bounded by either the hardware's peak compute throughput (measured in floating-point operations per second, or FLOPS) or its peak memory bandwidth (measured in bytes per second), whichever is hit first for a given ratio of computation to data movement, known as arithmetic intensity (Source: zeroentropy.dev).

LLM inference splits cleanly into two phases that sit on opposite sides of this roofline. **Prefill**, which processes the full input prompt in parallel, reuses each loaded weight across many tokens simultaneously, pushing the workload toward the compute-bound region of the roofline. **Decode**, which generates output tokens one at a time, touches each weight exactly once per token, placing it firmly in the memory-bandwidth-bound region (Source: rfriedmann.de). Because most user-facing chat and agent applications spend the majority of wall-clock time in decode, memory bandwidth, not FLOPS, is usually the binding constraint on user-perceived latency and aggregate throughput.

Mechanically, each decode step requires the GPU to read the resident KV cache for the current request, perform a small matrix multiplication between the new query and the cached keys, append the new key and value to the cache, and then stream the layer's full feed-forward weight matrices from HBM before repeating the process for every layer in the network and every subsequent token (Source: itkservices3.com). A commonly cited engineering approximation makes this concrete: tokens-per-second is approximately equal to memory bandwidth divided by bytes read per token. A GPU with 8 TB/s of bandwidth running a 70B-parameter model at FP8 precision (roughly 70 GB resident in memory) caps out at approximately 115 tokens per second per single-stream GPU, with FP4 quantization roughly doubling that ceiling by halving the bytes that must be streamed per token (Source: itkservices3.com). This relationship is precisely why the industry's hardware roadmap over the past several generations has prioritized HBM bandwidth growth (H100 to H200 to B200) over raw FLOPS growth: a University of California, Berkeley research team documented that peak server hardware FLOPS scaled at 3.0 times every two years over the past two decades, while DRAM and interconnect bandwidth scaled at only 1.6 and 1.4 times every two years respectively, a widening gap the researchers term the "memory wall" (Source: arxiv.org). The same researchers frame this shift as affecting AI applications broadly but note it is particularly acute for serving workloads specifically, since serving is dominated by the memory-bound decode phase rather than the more compute-friendly training or prefill phases, reinforcing why inference hardware sizing has to weight bandwidth more heavily than a training-hardware sizing exercise would (Source: arxiv.org).

Interestingly, real-world measurements complicate the simplest version of the bandwidth story. A 2026 preprint examining batch-1 decode across four NVIDIA GPU generations found that the achieved fraction of peak HBM bandwidth actually falls as peak bandwidth rises: an NVIDIA L4 reached roughly 81% of its analytic memory floor on a headline test, while an H100 reached only 27% under the same conditions, indicating that software scheduling overhead (not just raw hardware bandwidth) materially affects realized throughput (Source: arxiv.org). Practically, this means enterprises should treat vendor-quoted peak bandwidth figures as an upper bound on achievable performance rather than a reliable predictor, and should validate throughput on representative workloads before finalizing hardware purchase decisions, echoing the guidance from independent engineering blogs to "always benchmark on your target hardware" rather than trusting a calculator's output alone (Source: bentoml.com).

GPU Cluster Architecture for LLM Serving

When a single GPU's memory is insufficient to hold a model, or when throughput requirements exceed what one accelerator can deliver, enterprises must distribute the model or the workload across multiple GPUs, and increasingly across multiple physical servers. Two distinct interconnect technologies govern this scaling, operating at very different bandwidth tiers and physical scopes.

NVLink (and the associated NVSwitch fabric) is NVIDIA's proprietary interconnect for GPU-to-GPU communication within a single server node, engineered for the highest possible bandwidth over short physical distances. **InfiniBand** (or increasingly RDMA over Converged Ethernet, RoCE) connects GPUs across multiple server nodes in a cluster, prioritizing scalability over raw per-link bandwidth (Source: jarvislabs.ai). NVIDIA's own H100 documentation illustrates the same tiering at the chip-to-chip level: the Grace Hopper Superchip pairs a Grace CPU with a Hopper GPU over a dedicated chip-to-chip interconnect delivering 900 GB/s of bandwidth, seven times faster than a PCIe Gen5 link, while a standalone H100 accelerating data analytics workloads draws on 3 TB/s of GPU memory bandwidth per card (Source: nvidia.com) (Source: nvidia.com). NVIDIA's own guidance on distributed training and inference underscores the magnitude of this gap: GPU connections inside a node using NVLink are "orders of magnitude faster" than interhost networking options such as InfiniBand, the AWS Elastic Fabric Adapter (EFA), or standard Ethernet (Source: developer.nvidia.com). The GB200 NVL72's fifth-generation NVLink fabric scales intra-rack bandwidth to 130 TB/s across 72 GPUs, a leap over prior-generation eight-GPU NVLink domains (Source: nvidia.com).

This bandwidth hierarchy directly informs how enterprises should split a model across hardware. **Tensor parallelism**, which splits individual weight matrices across GPUs and requires frequent all-reduce communication after nearly every layer, should be confined within a single NVLink-connected node because it is highly communication-sensitive, incurring only modest overhead when kept inside a node's high-bandwidth NVLink domain; the same technique spread across InfiniBand-connected nodes would be prohibitively slow given the order-of-magnitude bandwidth gap noted above. Larger deployments that must span multiple nodes, whether to fit an extremely large MoE model or to scale aggregate throughput, instead rely on **pipeline parallelism** (splitting the model by layer across nodes) or data parallelism (replicating the full model across nodes to serve independent request streams), both of which tolerate InfiniBand's comparatively higher latency and lower per-link bandwidth. Deployment guidance from VMware for GPUDirect RDMA over InfiniBand specifically targets this multi-node inference use case, coordinating GPU resources across a Kubernetes-based supervisor cluster and a fabric manager service (Source: blogs.vmware.com).

For most enterprise deployments serving models up to roughly 70B dense parameters or MoE models with total parameters under 700B, a single eight-GPU HGX node with full intra-node NVLink is sufficient, and multi-node InfiniBand clusters only become necessary for the largest frontier models or for horizontally scaling throughput across many replica nodes to serve high concurrent user counts.

On-Premise LLM Hardware Requirements and vLLM/TensorRT-LLM Sizing

Beyond raw GPU selection, the choice of serving engine materially changes both the achievable throughput per GPU and the practical VRAM budget available for the KV cache. The two dominant open ecosystem options are **vLLM**, developed originally at UC Berkeley, and **NVIDIA TensorRT-LLM**, NVIDIA's proprietary but open-source inference optimization library.

vLLM is built around PagedAttention, an attention algorithm inspired by operating-system virtual memory paging that stores the KV cache in non-contiguous, fixed-size blocks rather than requiring each request's cache to occupy one contiguous memory region. The original 2023 paper from the vLLM team reports that this design achieves near-zero waste in KV cache memory and improves serving throughput of popular LLMs by 2 to 4 times compared to prior state-of-the-art systems such as FasterTransformer and Orca, at the same level of latency (Source: arxiv.org (Source: arxiv.org). vLLM's own project blog reports an even larger headline figure for a specific benchmark configuration: up to 24 times higher throughput than the Hugging Face Transformers library on LLaMA-7B and LLaMA-13B workloads, without any model architecture changes (Source: vllm.ai). For sizing purposes, vLLM's `gpu_memory_utilization` parameter directly controls what fraction of a card's VRAM the engine reserves for its own operation versus leaving free, and vLLM's documentation recommends **tensor parallelism** (splitting a model across multiple GPUs via the `tensor_parallel_size` setting) when a single GPU's memory is insufficient for the target model (Source: github.com).

vLLM's documentation also cautions that with tensor parallelism enabled, each worker process reads the entire model checkpoint before splitting it into shards, which makes disk-loading time longer in proportion to the tensor-parallel degree unless the checkpoint is pre-sharded (Source: github.com), a practical operational detail that affects cold-start time for large multi-GPU deployments.

TensorRT-LLM, NVIDIA's optimization library, targets a narrower but deeper hardware support matrix: officially supported architectures include the GB200 NVL72, the broader Blackwell architecture, Grace Hopper, Hopper (H100/H200), Ada Lovelace, and Ampere, running exclusively on Linux x86_64 or aarch64 (Source: nvidia.github.io). NVIDIA's support-matrix documentation notes that the library "optimizes the performance of a range of well-known models on NVIDIA GPUs" and warns that if a GPU architecture is not listed, the TensorRT-LLM team does not develop or test the software on it, limiting support to the community (Source: nvidia.github.io). NVIDIA reports TensorRT-LLM running Llama 4 at over 40,000 tokens per second on B200 GPUs, and Blackwell submissions using TensorRT-LLM with FP4 Tensor Cores enabled 3.4 times higher per-GPU performance on the Llama 3.1 405B MLPerf benchmark compared to an eight-GPU H200 system (Source: developer.nvidia.com). For enterprises already standardized on NVIDIA hardware and seeking maximum single-vendor optimization, TensorRT-LLM's tighter hardware-software co-design typically yields higher raw

throughput; for organizations wanting broader hardware portability (including AMD and future accelerators) or simpler deployment, vLLM's ecosystem is more common. In practice, many production deployments documented publicly, including the case studies below, standardize on vLLM specifically for its combination of throughput, hardware breadth, and operational simplicity.

On-premise hardware requirements ultimately compress into three purchasing tiers. Entry-level self-hosting for models up to roughly 30B dense parameters can run on a single workstation-class card or a small multi-GPU server; mid-size production serving of 70B-class dense models or MoE models with sub-200B total parameters typically requires a single eight-GPU HGX H100 or H200 node; and frontier MoE models in the 600B-plus range require either a full eight-GPU H200 node (1,128 GB aggregate VRAM) or multi-node clusters when using H100-class hardware, whose eight-GPU aggregate of 640 GB cannot hold such models without aggressive INT4 quantization (Source: iternal.ai).

Quantization and Its Impact on Hardware Sizing

Quantization, the practice of representing model weights (and sometimes activations and the KV cache) in fewer bits than the original training precision, is the single most effective lever for reducing hardware footprint without buying additional GPUs. *Table 2* below summarizes the trade-offs across the precision levels most commonly deployed in production.

Table 2: Quantization Precision, Memory Savings, and Accuracy Trade-offs

PRECISION	BITS	MEMORY SAVINGS VS. FP16	THROUGHPUT GAIN	TYPICAL ACCURACY RETENTION	RECOMMENDED USE
FP16/BF16	16	Baseline	Baseline	100%	Maximum quality, training-parity
FP8	8	50%	~1.5 to 2.2x	~99.9%	H100/H200 production default
INT8 (W8A8)	8	50%	~1.5 to 2x	~99.96%	General production
GPTQ-INT4	4	75%	~2.7x	~98.1%	Memory-constrained deployments

Source: On-Prem AI Hardware Sizing Guide quantization comparison table (Source: iternal.ai).

Beyond weight quantization alone, engines that also quantize the KV cache to FP8 report a further reduction in cache memory with negligible measured quality impact, compounding the near-zero waste already achieved by vLLM's PagedAttention design discussed earlier (Source: arxiv.org). Architectural techniques native to specific model families compound these savings further: DeepSeek and Kimi's Multi-head Latent Attention (MLA) mechanism substantially reduces KV cache size as an inherent property of the architecture, independent of any post-training quantization applied on top, which is a major reason those model families remain deployable on single-node hardware despite very large total parameter counts.

Among the specific quantization algorithms, **AWQ** (Activation-aware Weight Quantization), developed by researchers at MIT, was originally motivated by on-device deployment, where the authors note that "the astronomical model size and the limited hardware resource pose significant deployment challenges" for running LLMs locally rather than in a data center (Source: arxiv.org), a motivation that applies equally to enterprise on-premise sizing where hardware budgets, not just data-center capacity, constrain what can be deployed. AWQ protects the small fraction of "salient" weight channels that disproportionately affect model accuracy by identifying them through activation statistics rather than weight magnitude alone; the technique's authors report that protecting just 1% of salient weights can greatly reduce quantization error, and their accompanying TinyChat inference framework delivers more than 3 times speedup over the Hugging Face FP16 implementation (Source: arxiv.org (Source: arxiv.org). **GPTQ**, the other widely deployed post-training quantization method, is frequently paired with the ExLlamaV2 inference runtime and, according to developer community testing on Reddit's r/LocalLLaMA forum, can inference faster than an equivalent-bitrate EXL2 model under some configurations, though such community benchmarks should be treated as directional rather than authoritative given the lack of standardized methodology (Source: reddit.com). The same community discussion also found EXL2 to be the fastest of the tested formats overall, followed by GPTQ run through the original ExLlama v1 runtime, illustrating that runtime implementation, not just bit width, materially affects achieved throughput for a given quantization format (Source: reddit.com). One commenter on the same thread summarized the practical takeaway for practitioners weighing these formats: the surprising result was that the choice of runtime matters as much as the choice of quantization algorithm itself, since a properly optimized runtime for an older format can outperform a newer format on a less-optimized runtime (Source: reddit.com).

The practical sizing consequence of quantization choice is dramatic at the model-family level. Using the FP16/FP8/INT4 weight-memory comparison for current-generation open models, a 671B-parameter DeepSeek V3 requires 1,342 GB at FP16, 671 GB at FP8, and approximately 336 GB at INT4, meaning the choice of precision alone determines whether the model requires two eight-GPU H200 nodes, one eight-GPU H200 node, or a single four-GPU H200 configuration (Source: iternal.ai).

Enterprise GPU Sizing Guide: Cost and Cloud Comparison

Once technical requirements are established, enterprises must decide between renting cloud GPU capacity and purchasing on-premises hardware, a decision that hinges heavily on utilization patterns, data residency requirements, and total cost of ownership over a multi-year horizon. Cloud GPU pricing for the same underlying silicon varies by a striking margin across providers. As of July 2026, on-demand H200 pricing ranged from **$2.40 per GPU-hour** on the Hyperbolic marketplace to **$10.60 per GPU-hour** on Azure's ND96isr H200 v5 instances, a more than fourfold spread, with AWS's p5e.48xlarge instance sitting at $7.91 per GPU-hour and CoreWeave at $6.16 per GPU-hour (Source: thundercompute.com (Source: thundercompute.com). For H100 capacity, CoreWeave's eight-GPU HGX H100 node lists at $49.24 per hour, which normalizes to $6.16 per GPU-hour (Source: coreweave.com).

At the newest hardware tier, CoreWeave prices an eight-GPU HGX B200 node at $68.80 per hour on-demand, and a full GB200 NVL72 rack at $42.00 per hour on-demand (Source: coreweave.com (Source: coreweave.com). Outright purchase pricing tells a similar story of steep premiums for newer memory-dense silicon: a single H200 GPU costs $30,000 to $40,000, with a full eight-GPU HGX H200 server running $300,000 to $500,000 depending on original equipment manufacturer (OEM) configuration (Source: thundercompute.com), while the AMD MI300X carries an estimated unit price of $10,000 to $15,000 (Source: spheron.network).

Independent GPU cloud aggregators corroborate the wide spread in per-GPU rental pricing across the broader accelerator lineup. Tracking more than 60 providers, one such aggregator lists H100 rentals spanning $0.61 to $14.90 per GPU-hour, A100 rentals spanning $0.13 to $5.04 per GPU-hour, L40S rentals spanning $0.50 to $7.58 per GPU-hour, and the newest B200 rentals spanning $2.69 to $16.11 per GPU-hour as of mid-2026 (Source: getdeploying.com) (Source: getdeploying.com). This spread, wider than the H200-specific range cited above, reflects the fact that older, more widely available GPUs such as the A100 have far more competing suppliers than the newest Blackwell-generation cards, pushing their low end down as capacity commoditizes over time. The same aggregator lists the newer NVIDIA B300 accelerator, with up to 288 GB of HBM3e memory, at $2.50 to $18.00 per GPU-hour, a range that overlaps substantially with B200 pricing despite the memory capacity increase, indicating that early-generation pricing for the newest cards has not yet settled (Source: getdeploying.com). Jarvislabs' pricing guide independently corroborates the specifications and purchase figures cited above, stating plainly that "the NVIDIA H200 GPU costs $30K-$40K to buy outright and $3.72-$10.60 per GPU hour to rent (January 2026)" and that "the NVIDIA H200 offers 141GB of HBM3e VRAM" (Source: jarvislabs.ai), independently reproducing the figures reported by other sources. The same guide also poses the open question of whether "NVIDIA H200 prices drop when Blackwell GPUs ship" in greater volume, a dynamic worth factoring into any multi-year hardware procurement plan rather than assuming current price levels are fixed (Source: jarvislabs.ai).

Table 3 below summarizes representative eight-GPU node configurations and their investment levels, useful as a starting reference point for enterprise budget planning.

Table 3: Representative Multi-GPU Node Configurations and Investment (2026)

CONFIGURATION	MODEL SERVED	THROUGHPUT	TOTAL MEMORY	APPROX. INVESTMENT
1x H100 SXM	Llama 4 Scout (INT4)	~120 to 150 tok/s	80 GB	$35,000 to $40,000
4x H100 SXM	Qwen 3 235B (FP8)	~1,400 tok/s	320 GB	$140,000 to $160,000
8x H100 SXM	DeepSeek V3 (AWQ INT4)	~3,000 tok/s	640 GB	~$300,000
4x H200 SXM	Qwen 3.5-397B (FP8)	~4,600 tok/s	564 GB	$140,000 to $175,000
8x H200 SXM	DeepSeek V3 (FP8)	~2,864 tok/s	1,128 GB	~$315,000
8x H200 SXM	Llama 4 Scout (FP8)	~12,432 tok/s	1,128 GB	~$315,000

Source: On-Prem AI Hardware Sizing Guide performance-scaling table (Source: iternal.ai).

The table shows that throughput does not scale linearly with GPU count or cost. The jump from a four-GPU to an eight-GPU H200 configuration on DeepSeek V3 barely doubles memory but only modestly increases throughput, because at this scale communication overhead between GPUs, not raw compute, increasingly dominates. This reinforces the earlier point that memory bandwidth and interconnect topology, not GPU count alone, are the deciding factors in cluster-level throughput. For decision-makers evaluating build-versus-rent, the case studies later in this report show that owned hardware can amortize faster than continued consumption-based cloud billing once monthly usage is large and predictable, with the documented examples below achieving payback in well under a year.

Case Studies and Real-World Examples

A German Engineering Consultancy: L40S-Based Private AI Platform

A 200-employee German structural engineering consultancy replaced approximately EUR 14,000 per month in cloud AI subscriptions and application programming interface (API) costs with a fully self-hosted platform built on a single Supermicro server housing four **NVIDIA L40S** GPUs (48 GB GDDR6 memory each, 192 GB total), serving Qwen3-32B and Qwen3-8B models through vLLM (Source: particula.tech). The firm chose the L40S specifically over the H100 for cost reasons: approximately EUR 7,500 per card versus EUR 28,000-plus for an H100, judging the L40S's inference-optimized Ada Lovelace architecture appropriate since the workload was serving, not training, models (Source: particula.tech). Total hardware cost, including server chassis, dual Xeon CPUs, 256 GB RAM, and 4 TB NVMe storage, came to EUR 48,000, and the deployment paid for itself in under four months, with the firm reporting first-year net savings after hardware cost of EUR 115,000 (Source: particula.tech). At peak usage, the system handled 30 to 50 concurrent requests from 200 employees, with sub-2-second responses for the 8B model and 5 to 15-second responses for complex 32B-model tasks (Source: particula.tech). Adoption climbed from 40% of employees in the first month to 94% by month three (Source: particula.tech).

A West African Tier-1 Bank: Sovereign, Air-Gapped LLM Platform

A Tier-1 universal bank headquartered in West Africa, operating across nine countries, consolidated a fragmented set of individually commissioned LLM pilots into a single on-premises, air-gapped inference platform running across five accelerators, delivered over a 36-week engagement (Source: mindmapdigital.ai). The platform now serves more than 40 production AI use cases, spanning contact-center conversation assistance, credit-memo drafting for SME lending, SWIFT message parsing, and internal policy question-answering, with zero external LLM calls leaving the air-gapped environment (Source: mindmapdigital.ai). The bank reported inference cost per use case approximately **62% lower** than its previous mix of public cloud LLM endpoints, driven by shared infrastructure, model right-sizing that routes simple requests to smaller models, and elimination of per-token markups (Source: mindmapdigital.ai). New use cases now reach production in 4 to 8 weeks, down from the 6 to 9 months comparable pilots previously required, and the bank's regulator explicitly cited the deployment as a reference architecture in a published data-residency directive following a compliance audit (Source: mindmapdigital.ai (Source: mindmapdigital.ai).

SS&C Technologies: Shared H100/H200 Private Cloud at Multi-Tenant Scale

Financial technology firm SS&C Technologies built a private-cloud AI-as-a-service platform that keeps all data on-premises while serving generative AI inference, embeddings, and retrieval-augmented generation (RAG) to internal business lines, publicly documented through an InfoQ technical presentation (Source: news.lavx.hu). The platform's hardware footprint totals 80 GPUs, a mix of H100 and H200 units, across 12 dual-socket AMD EPYC compute nodes with 100 Gigabit Ethernet (GbE) leaf-spine networking (Source: news.lavx.hu). The headline operational claim is that a single eight-chip H100 chassis in the firm's Kansas City facility simultaneously serves more than 250 production tenants and over 1,000 use cases without exceeding budgeted GPU spend, using Kubernetes 1.30, vLLM as the primary inference runtime, and Valkey (a Redis-compatible in-memory store) with Lua scripts for atomic rate-limiting and priority queuing (Source: news.lavx.hu (Source: news.lavx.hu). SS&C is exploring further optimization through NVIDIA KV-cache extensions targeted at cutting long-context token-generation latency by up to 15%, and 4-bit INT4 quantization kernels to increase the number of concurrent models served per GPU (Source: news.lavx.hu).

Data Analysis and Evidence

Quantifying the current state of the LLM inference hardware market requires triangulating vendor financial disclosures, independent research-firm forecasts, and standardized benchmark consortium data. On the supply side, **NVIDIA** reported record revenue of $81.6 billion for the first quarter of fiscal year 2027 (the quarter ended April 26, 2026), up 85% year over year, with record Data Center revenue of $75.2 billion, up 92% year over year (Source: nvidianews.nvidia.com (Source: nvidianews.nvidia.com). Within that figure, Data Center compute revenue specifically was $60.4 billion, up 77% year over year and up 18% sequentially, while Data Center networking revenue (spanning NVLink, InfiniBand, and related fabric products directly

relevant to cluster sizing) reached a record $14.8 billion, up 199% year over year (Source: nvidianews.nvidia.com). CEO Jensen Huang characterized the underlying trend as "the buildout of AI factories, the largest infrastructure expansion in human history," accelerating at extraordinary speed (Source: nvidianews.nvidia.com).

On the demand side, market research firm IDC recorded worldwide AI infrastructure spending of $89.9 billion in the fourth quarter of 2025 alone, a 62.2% year-over-year increase, bringing full-year 2025 spending to $318 billion, more than double the $153 billion recorded in 2024 (Source: idc.com). IDC projects the market will reach $487 billion in 2026, roughly 53% year-over-year growth, and exceed $1 trillion by 2029 at an approximately 31% five-year compound annual growth rate (CAGR) from 2025 (Source: idc.com). IDC's Juan Seminara, Research Director for Worldwide Infrastructure Trackers, characterized the trend as structural rather than cyclical, noting that "enterprises and hyperscalers are not building for today's workloads, but for AI architectures that are still being defined" (Source: idc.com).

On the standardized-benchmark side, MLCommons' MLPerf Inference v5.0 suite (published April 2025) introduced two new LLM benchmarks, Llama 3.1 405B Instruct and a lower-latency Llama 2 70B interactive variant, reflecting the industry's shift toward longer context windows (Llama 3.1 405B supports a 128,000-token context, versus Llama 2 70B's original 4,096 tokens) and stricter latency targets suited to reasoning and agentic use cases (Source: mlcommons.org). MLCommons continued using the OpenOrca dataset and ROUGE accuracy metrics for the Llama 2 70B benchmark specifically "due to their relevance and proximity to chatbot tasks," a methodological choice that keeps the benchmark's accuracy scoring aligned with real conversational workloads rather than narrow academic tasks (Source: mlcommons.org). The interactive Llama 2 70B benchmark tightened its 99th-percentile time-to-first-token requirement to 450 milliseconds and its 99th-percentile time-per-output-token requirement to 40 milliseconds, based on analysis of production platforms including ChatGPT and Perplexity AI, finding that a 50th-percentile token generation rate of 20 to 50 tokens per second is critical for a seamless user experience (Source: mlcommons.org (Source: mlcommons.org). On the resulting benchmark, NVIDIA's Blackwell-based eight-GPU B200 system achieved 98,443 (server) and 98,858 (offline) tokens per second on Llama 2 70B, compared to 33,072 and 34,988 tokens per second respectively for an eight-GPU H200 system, a 3x and 2.8x speedup (Source: developer.nvidia.com). On the larger Llama 3.1 405B benchmark, the rack-scale GB200 NVL72 delivered up to 3.4 times higher per-GPU performance than the eight-GPU H200 system, and an unverified NVIDIA-run result on GB200 NVL72 reached 869,203 tokens per second on the earlier Llama 2 70B benchmark (Source: developer.nvidia.com). Even the prior-generation Hopper architecture continued improving through pure software optimization, with NVIDIA reporting Llama 2 70B throughput on H100 GPUs increasing by up to 1.5 times over roughly one year purely through software enhancements, without any hardware change. Independent benchmark analysis firm Signal65 corroborates the pace of this year-over-year progress at the suite level: comparing MLPerf Inference v4.0 to v5.0, the median Llama 2 70B score across all submitted systems doubled, and the single top-performing system achieved a 3.3 times performance increase, gains the firm attributes to a combination of new quantization techniques, newer hardware accelerators, and continued inference software optimization (Source: signal65.com).

These figures collectively demonstrate that hardware sizing decisions made today should account for a documented pattern of continual throughput improvement on existing silicon through software alone, meaning that a hardware purchase should not be evaluated solely against today's benchmark numbers but against a trajectory of ongoing optimization from serving-engine and firmware updates over the hardware's useful life.

Implications and Future Directions

Several structural trends are likely to reshape LLM inference hardware sizing over the next two to three years. First, the widening gap between compute-throughput scaling and memory-bandwidth scaling documented in "AI and Memory Wall" research is unlikely to close on its own, meaning enterprises should expect memory bandwidth, not FLOPS, to remain the binding constraint on decode throughput for the foreseeable future, and should weight bandwidth-per-dollar more heavily than compute-per-dollar in procurement decisions (Source: arxiv.org).

Second, the accelerating shift toward mixture-of-experts (MoE) architectures across frontier open-weight models (Llama 4, DeepSeek V3/R1, Qwen 3.5, GLM-5, Kimi K2.5, Mistral Large 3), most of which use MoE designs that fundamentally change sizing math (Source: iternal.ai), means enterprise sizing methodology must evolve past simple "parameters times bytes" heuristics toward separately accounting for total resident parameters (which set the VRAM floor) and active parameters (which set the compute and latency profile). This trend favors GPUs with the highest memory-per-card, since it directly determines how many experts can be resident without resorting to expensive multi-node deployments, reinforcing the H200 and future Blackwell-generation cards' design emphasis on capacity alongside bandwidth.

Third, quantization is likely to continue moving toward lower precision as a default rather than an optimization, with FP4 already delivering roughly twice the peak throughput of FP8 on Blackwell hardware while meeting MLPerf benchmark accuracy requirements, suggesting that INT4/FP4-class formats will increasingly become the production default for latency-sensitive serving rather than a fallback for memory-constrained deployments (Source: developer.nvidia.com).

Fourth, on the economics front, the fourfold cloud-pricing spread documented for identical H200 silicon across providers, alongside the 62% and 3.5-to-4-month payback figures reported by the West African bank and German engineering firm case studies discussed earlier, suggests that enterprises with predictable, sustained inference load are increasingly finding on-premises or dedicated private-cloud deployment more economical than continuous hyperscaler consumption once monthly usage becomes large and predictable enough to amortize fixed hardware cost. Given IDC's projection of AI infrastructure spending exceeding $1 trillion by 2029, capacity constraints and continued price volatility in the cloud GPU rental market are plausible risks enterprises should factor into multi-year sizing plans, favoring flexible architectures (tensor-parallel-ready, quantization-agnostic serving stacks) that can absorb hardware generation changes without a full re-architecture (Source: idc.com).

Frequently Asked Questions (FAQs)

How do I calculate VRAM for LLM inference? Use the formula Memory (GB) = Parameters (billions) x (Bits per parameter / 8) x (1 + Overhead percentage), where overhead of 20 to 50% covers the KV cache, activations, and framework buffers (Source: bentoml.com). A quick rule of thumb is approximately 2 GB per 1 billion parameters at 16-bit precision (Source: modal.com).

What GPU do I need for LLM inference? It depends on model size and precision. A 70B model in FP16 needs roughly 140 GB, requiring two 80 GB H100s or a single 141 GB H200 (see the VRAM calculation section above). Frontier MoE models over 600B parameters, such as DeepSeek V3, typically require the full memory of an eight-GPU H200 node (see the Quantization section above for the exact FP16/FP8/INT4 memory breakdown).

What are the on-premise LLM hardware requirements for a mid-size enterprise? Most production deployments of 70B-class dense models or sub-200B MoE models fit on a single eight-GPU HGX H100 or H200 node (see the On-Premise Hardware Requirements section above). Smaller deployments (up to 200 employees) have run successfully on four L40S GPUs (192 GB total) at roughly EUR 48,000 total hardware cost, as documented in the German engineering consultancy case study above.

What are the vLLM hardware requirements? vLLM runs on any CUDA-capable GPU with sufficient VRAM for weights plus KV cache, using its `tensor_parallel_size` setting to split models across multiple GPUs when a single card's memory is insufficient (Source: github.com). It achieves near-zero KV cache waste through PagedAttention, improving throughput 2 to 4 times over prior serving systems (Source: arxiv.org).

What are the TensorRT-LLM hardware sizing considerations? TensorRT-LLM officially supports GB200 NVL72, Blackwell, Grace Hopper, Hopper, Ada Lovelace, and Ampere architectures on Linux (Source: nvidia.github.io). It typically delivers higher raw throughput on NVIDIA hardware than vLLM through tighter hardware-software co-design, including FP4 precision support on Blackwell hardware (see the Implications section above for the throughput comparison versus FP8).

How does memory bandwidth affect LLM inference? Decode (token generation) is memory-bandwidth-bound because each token requires re-streaming the full weight set from HBM; tokens-per-second is approximately memory bandwidth divided by bytes read per token (Source: itkservices3.com). Prefill, by contrast, is compute-bound (Source: rfriedmann.de).

Should I build an enterprise GPU cluster or rent cloud capacity? Cloud H200 pricing spans $2.40 to $10.60 per GPU-hour depending on provider (see the Enterprise GPU Sizing Guide section above for the full provider comparison). Enterprises with predictable, sustained inference load have documented payback periods under four months and cost reductions up to 62% versus cloud after moving to owned infrastructure, as shown by the case studies earlier in this report.

Conclusion

Sizing hardware for LLM inference is fundamentally an exercise in balancing three interacting constraints: total memory capacity, memory bandwidth, and cost, layered against a workload that behaves differently in its compute-bound prefill phase versus its bandwidth-bound decode phase. The dominant formula, model weight memory plus KV cache plus a 20-to-50% overhead margin divided by available VRAM, provides a reliable starting estimate, but real deployments consistently show that quantization choice, serving-engine efficiency, and interconnect topology can each shift the final hardware bill by a factor of two or more. As of July 2026, enterprises have a genuinely wide menu of options, from four-GPU L40S workstations serving a 200-person firm for roughly EUR 48,000, through eight-GPU H100 or H200 nodes serving hundreds of concurrent tenants, up to rack-scale GB200 NVL72 systems built for trillion-parameter frontier models.

The evidence gathered across vendor specifications, peer-reviewed research, MLPerf benchmark results, and named enterprise deployments points toward a consistent set of practical recommendations: size for total (not active) parameters when working with mixture-of-experts models, budget generously for KV cache growth under realistic concurrency rather than best-case single-request scenarios, default to FP8 or lower precision unless accuracy requirements specifically preclude it, and validate vendor-quoted bandwidth and throughput figures against representative workloads before

finalizing a purchase, since achieved performance can diverge meaningfully from theoretical peaks. With global AI infrastructure spending on a trajectory toward $1 trillion by 2029 and hardware generations continuing to arrive on a roughly annual cadence, the sizing discipline outlined in this report is likely to remain a recurring, rather than one-time, exercise for any enterprise running production LLM workloads.

Tags: llm inference hardware sizing, gpu requirements for llm inference, vram calculator llm, on-premise llm hardware, gpu cluster llm serving, vllm hardware requirements, tensorrt-llm sizing, nvidia h100 h200, enterprise gpu sizing

---