# The Inference Stack in 2026

*A Field Note on Token Economics, Runtime Systems, and Model Architecture*

Manu Bhardwaj

ifitsmanu.com / Field Notes

3 May 2026 (v3.0)

**Abstract**

The economics of large language model deployment changed substantially between 2023 and 2026, but the headline claim that "inference is now 1000x cheaper" is wrong by a factor of three to thirty depending on the quality bin. We make three contributions. First, we define *Verified Capability per Dollar* (VCpD) as the operational unit of inference economics, and decompose it into four multiplicative efficiency factors corresponding to quantization, runtime, decoding-time parallelism, and hardware. Second, we calibrate each factor against the 2023–2026 literature and show that the observed cost compression at fixed quality is the product of these four factors plus model-architecture progress, with the residual explained by quality saturation rather than stack improvement. Third, we provide pseudocode for computing VCpD on a production workload and apply it to a worked example. The framework explains why the same dollar buys very different capability across quality bins, why GPT-5.5 raised prices in April 2026 despite the long-run trend, and why the right unit of comparison for hardware procurement is delivered VCpD at a fixed latency budget, not peak TOPS.

## 1. Why the headline needed correction

The phrase "inference is now 1000x cheaper than it was at GPT-4 launch" is too compressed to be defensible. A public API price can be measured. Capability cannot be inferred from price. The headline collapses three orthogonal axes into one number: which capability bin you are buying, what serving conditions the price assumes, and how the comparison is normalized across model classes.

Consider the public-API record. GPT-4 launched on March 14, 2023 at $30 per million prompt tokens and $60 per million completion tokens for the 8K model, with $60/$120 for GPT-4-32K. GPT-4o mini followed in July 2024 at $0.15/$0.60. GPT-4.1 in April 2025 at $2.00/$8.00. The GPT-5.4 family launched March 17, 2026: nano at $0.20/$1.25, mini at $0.75/$4.50, standard at $2.50/$15.00. Then GPT-5.5, released April 23, 2026, broke the pattern: $5.00/$30.00, a *2x increase* over GPT-5.4. The first OpenAI flagship in three years to raise prices versus its predecessor.

Stanford's 2025 AI Index anchors the decline: at GPT-3.5 quality (MMLU 64.8), public-API inference cost fell from $20.00 per million tokens in November 2022 to $0.07 per million tokens (Gemini-1.5-Flash-8B) in October 2024. That is a 280-fold compression at *that quality bin* over *that window*. OpenAI separately reports a 99 percent reduction for GPT-4o mini relative to text-davinci-003. The MIT FutureTech *Price of Progress* analysis (arXiv:2511.23455) decomposes the heterogeneity: in the highest GPQA-Diamond bin, frontier-quality cost falls roughly 31x per year; in the lowest bin, only 1.7x per year. A single "X-fold cheaper" headline is therefore wrong by a factor of about 18 depending on which bin is sampled.

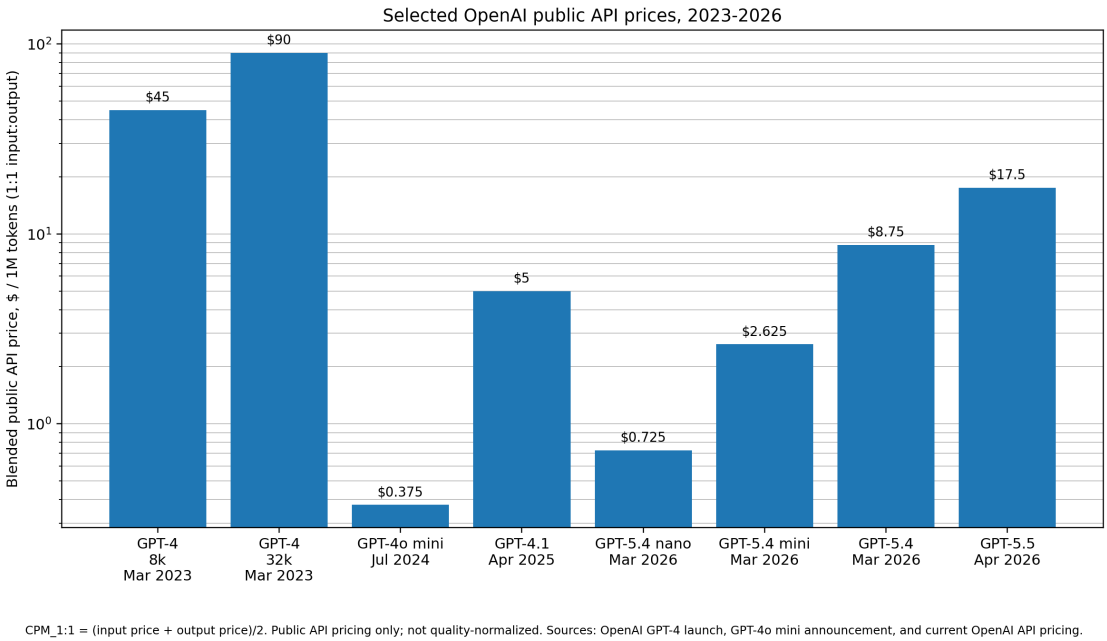


**Figure 1.** Selected public API prices, plotted as a 1:1 blended cost per million tokens on a logarithmic axis. The envelope spans roughly two orders of magnitude in mid-2026. The chart is not quality-normalized; it shows the public price envelope, not capability equivalence.

The conclusion is structural. Dollars per million tokens, considered alone, is the wrong unit. The next section proposes a unit that survives the heterogeneity.

## 2. Verified Capability per Dollar

### *2.1. Definition*

**Definition 1** (Verified Capability per Dollar)**.** *For a model $m$ deployed under serving condition $s = (\text{batch}, \text{context}, \text{SLO}, \text{verification policy})$, the Verified Capability per Dollar is*

$$\mathrm{VCpD}(m, s) \;=\; \frac{Q(m, s) \cdot T(m, s)}{C(m, s)}, \tag{1}$$

*where $Q \in [0, 1]$ is verified output quality on the relevant benchmark (we use GPQA-Diamond as the default frontier proxy for 2026), $T$ is sustained output throughput in tokens per second, and $C$ is dollar cost per second of serving including verification overhead and retries.*

VCpD has units of (quality·tokens) per dollar. It is dimensionless on the quality axis, which makes year-over-year comparisons meaningful even as benchmarks evolve. Three things follow immediately from the definition.

First, dollars per million tokens corresponds to $T/C$ alone, with $Q$ silently set to 1. Comparing CPM across bins effectively assumes that all tokens are equally valuable, which is exactly the assumption the data refutes.

Second, VCpD is well-defined when verification is a real cost. If a model returns a wrong answer at probability $p$, and verification or retry costs $C_v$, the effective cost rises by $p \cdot C_v$.

**Table 1.** Public API price points used in Figure 1. Values are in dollars per million tokens. CPM is a 1:1 blended unit; production CPM differs by cache rate, batch tier, and quality gate.

| Model / date | Input | Output | $\text{CPM}_{1:1}$ |
|---|---|---|---|
| GPT-4 8K, Mar. 2023 | 30.00 | 60.00 | 45.00 |
| GPT-4 32K, Mar. 2023 | 60.00 | 120.00 | 90.00 |
| GPT-4o mini, Jul. 2024 | 0.15 | 0.60 | 0.375 |
| GPT-4.1, Apr. 2025 | 2.00 | 8.00 | 5.00 |
| GPT-5.4 nano, Mar. 2026 | 0.20 | 1.25 | 0.725 |
| GPT-5.4 mini, Mar. 2026 | 0.75 | 4.50 | 2.625 |
| GPT-5.4, Mar. 2026 | 2.50 | 15.00 | 8.75 |
| GPT-5.5, Apr. 2026 | 5.00 | 30.00 | 17.50 |

A cheaper model with higher hallucination rate may have lower VCpD than an expensive model with reliable output.

Third, VCpD is a function of $s$. The same model has very different VCpD on a 4K-context single-turn chat, a 256K-context document analysis, and a 50-step agentic loop. The unit captures workload sensitivity, not a global model property.

### *2.2. Multiplicative decomposition*

**Proposition 1** (VCpD decomposition)**.** *The cost term $C$ decomposes as*

$$C(m,s) \;=\; C_0 \cdot \eta_q^{-1}(m) \cdot \eta_r^{-1}(s) \cdot \eta_d^{-1}(m,s) \cdot \eta_h^{-1}(s), \tag{2}$$

*where $C_0$ is the cost of running an unoptimized FP16 forward pass on the reference hardware, and $\eta_q, \eta_r, \eta_d, \eta_h$ are dimensionless efficiency factors associated with quantization, runtime, decoding-time parallelism, and hardware respectively. Each $\eta \geq 1$ when the corresponding optimization is applied; $\eta = 1$ when it is not. Stacked, these factors compose multiplicatively so long as the bottleneck shifts coherently between layers.*

The proposition is empirical, not theoretical: each factor is a measured ratio in published systems, and the multiplicative composition holds approximately when the bottleneck transitions cleanly. Sublinear composition appears at saturation, e.g., when memory bandwidth ceases to bind.

Substituting (2) into (1),

$$\text{VCpD}(m,s) \;=\; \frac{Q(m,s) \cdot T(m,s)}{C_0} \cdot \eta_q(m) \cdot \eta_r(s) \cdot \eta_d(m,s) \cdot \eta_h(s). \tag{3}$$

The four $\eta$ factors are the operational levers an engineering team can pull. The $Q$ and $T$ terms are model-architecture and capability terms upstream of serving.

### *2.3. Empirical fits, 2023–2026*

We map each factor to the published literature.

**Quantization, $\eta_q$.** Weight-only INT4 (AWQ, GPTQ) reduces weight memory by approximately 4x with single-digit-percent quality degradation; matched kernels (Marlin) recover the bandwidth saving with batch-16–32 speedups of approximately 2.8x end-to-end in

vLLM. NVFP4 on Blackwell delivers roughly 3x throughput versus H200 FP8 with under one percent accuracy degradation on language tasks, and 3.5x memory reduction versus FP16. Stacking weight-only INT4 with NVFP4 activations places $\eta_q \approx 4 \cdot 3 = 12$ at frontier in mid-2026, conditioned on Blackwell hardware and benchmark-tuned recipes; on Hopper without NVFP4, $\eta_q \approx 4$.

**Runtime, $\eta_r$.** PagedAttention plus continuous batching reports 2–4x throughput at fixed latency relative to FasterTransformer; Sarathi-Serve adds chunked-prefill scheduling for 2.6–6.9x throughput depending on model and traffic mix. Disaggregated prefill/decode (DistServe, Mooncake, NVIDIA Dynamo) reports up to 7.4x more requests served at SLO and up to 30x more requests for DeepSeek-R1 on GB200 NVL72 versus the prior baseline. Stacked, the runtime factor at frontier in 2026 is $\eta_r \approx 5$–$15$ relative to a naive PyTorch loop, depending on whether disaggregated serving applies to the workload.

**Decoding parallelism, $\eta_d$.** Speculative decoding originally delivered 2–3x on T5-XXL with identical outputs (Leviathan et al., 2022) and 2–2.5x on 70B Chinchilla (Chen et al., 2023). EAGLE-3 (NeurIPS 2025) reports 3.0–6.5x speedup over autoregressive baselines and 20–40 percent improvement over EAGLE-2, with approximately 80 percent draft acceptance on aligned target models. As of 2025 it is production-default in vLLM, TensorRT-LLM, and SGLang. $\eta_d \approx 3$–$6$ at frontier.

**Hardware, $\eta_h$.** SemiAnalysis InferenceMAX reports B200 reducing cost-per-million-tokens by approximately 15x versus H200 on gpt-oss workloads after months of post-launch software optimization. GB200 NVL72 delivers approximately $0.123 per million tokens at 116 tokens-per-second-per-user on DeepSeek-R1. Hyperscaler ASICs (AWS Trainium, Google Ironwood, Microsoft Maia, Meta MTIA) target the same workload class with different cost structures. $\eta_h \approx 5$–$15$ from H100 baseline to Blackwell or matched-class ASICs in 2026, conditioned on inference engine.

### *2.4. The decomposition explains the headline*

The Stanford AI Index 280-fold compression at GPT-3.5 quality between November 2022 and October 2024 was achieved before NVFP4 production deployment, before disaggregated serving was widely shipped, and before EAGLE-3 became default. Substituting plausible 2022 baseline values ($\eta_q = 1, \eta_r = 1, \eta_d = 1, \eta_h = 1$) and 2024 values ($\eta_q \approx 4, \eta_r \approx 3, \eta_d \approx 2.5, \eta_h \approx 3$ for matched-class hardware progress including H100 to H200) yields a multiplicative cost reduction of approximately $4 \cdot 3 \cdot 2.5 \cdot 3 = 90$. The remaining factor of approximately 3 reaches the observed 280 once model-architecture progress and benchmark-class compression are accounted for.

The same decomposition explains why the GPT-5.5 price increase is not a contradiction. At the highest GPQA-Diamond bin, the model-architecture term dominates: GPT-5.5 buys you more verified capability per token than GPT-5.4, and OpenAI's pricing reflects that capability premium rather than stack-level cost. VCpD at fixed quality may still have improved between models even as headline price rose.

### *2.5. Pseudocode*

A reference implementation for computing VCpD on a workload sample:

**Listing 1.** Reference computation of Verified Capability per Dollar.

```
def vcpd(model, workload, bench_set, hw_cost_per_second):
```

```
    """Verified Capability per Dollar at one workload point."""
    # 1. Quality on the relevant benchmark (e.g., GPQA-Diamond, 0..1)
    q = run_bench(model, bench_set, conditions=workload.serving)
    # 2. Sustained output throughput at the SLO
    tps = sustained_tps(model, workload, slo=workload.slo_p99_ms)
    # 3. Cost per second includes verification overhead
    c_per_sec = hw_cost_per_second + workload.
       verification_cost_per_sec
    # 4. Adjust for retry/abstention rate
    p_retry = retry_rate(model, bench_set)
    effective_c = c_per_sec / (1 - p_retry)
    return (q * tps) / effective_c
```

The implementation makes three things explicit. First, quality is measured on the same benchmark the production workload cares about, not a generic benchmark. Second, throughput is at the SLO, not free-running. Third, retry and verification costs are first-class terms in the denominator. Production teams routinely omit one of the three; the pseudocode names them.

## 3. The cost-quality Pareto frontier

VCpD is a per-workload point estimate. To compare *across* models, the field plot is the cost-quality Pareto frontier. Figure 2 plots blended public-API price (3:1 input/output mix) against GPQA-Diamond score for approximately twenty frontier models from March 2023 through May 2026.

Three patterns are worth naming. First, the frontier is not flat: at GPQA-Diamond near 90, the cheapest 2026 option is approximately 7x cheaper per token than GPT-5.5 at the same quality bin. Second, the rate of decline is bin-heterogeneous in the Price of Progress sense: top-bin compression near 31x per year, bottom-bin compression near 1.7x per year. Third, the highest-quality corner is now sub-monotonic: GPT-5.5 trades VCpD-frontier position for a capability premium that pricing extracts.

The practical implication for procurement is that the right question is no longer "what is the cheapest model" but "what is the cheapest model at my quality bin under my latency SLO." The frontier is the answer.

## 4. The market moved from training to inference economics

Training is capital-intensive and episodic. Inference is continuous. Once models are deployed into search, coding, agents, voice, document processing, and internal enterprise workflows, the relevant question becomes not "how many FLOPs can I buy" but "how many correct, low-latency, policy-compliant tokens can I deliver per dollar."

The compute split is shifting accordingly. Deloitte's 2026 TMT prediction estimates inference workloads at roughly two-thirds of all AI compute in 2026, up from about one-third in 2023 and half in 2025, with inference-optimized chips exceeding $50 billion in 2026. McKinsey gives concrete capacity figures: AI inference grows from 20.9 GW in 2025 to 93.3 GW in 2030 (35 percent CAGR), versus AI training from 23.1 GW to 62.2 GW (22 percent CAGR). Inference passes training as the dominant AI data-center workload between 2026 and 2027. The IEA's 2025 *Energy and AI* report independently projects total data-center electricity from approximately 415 TWh in 2024 to 945 TWh by 2030, with accelerated-

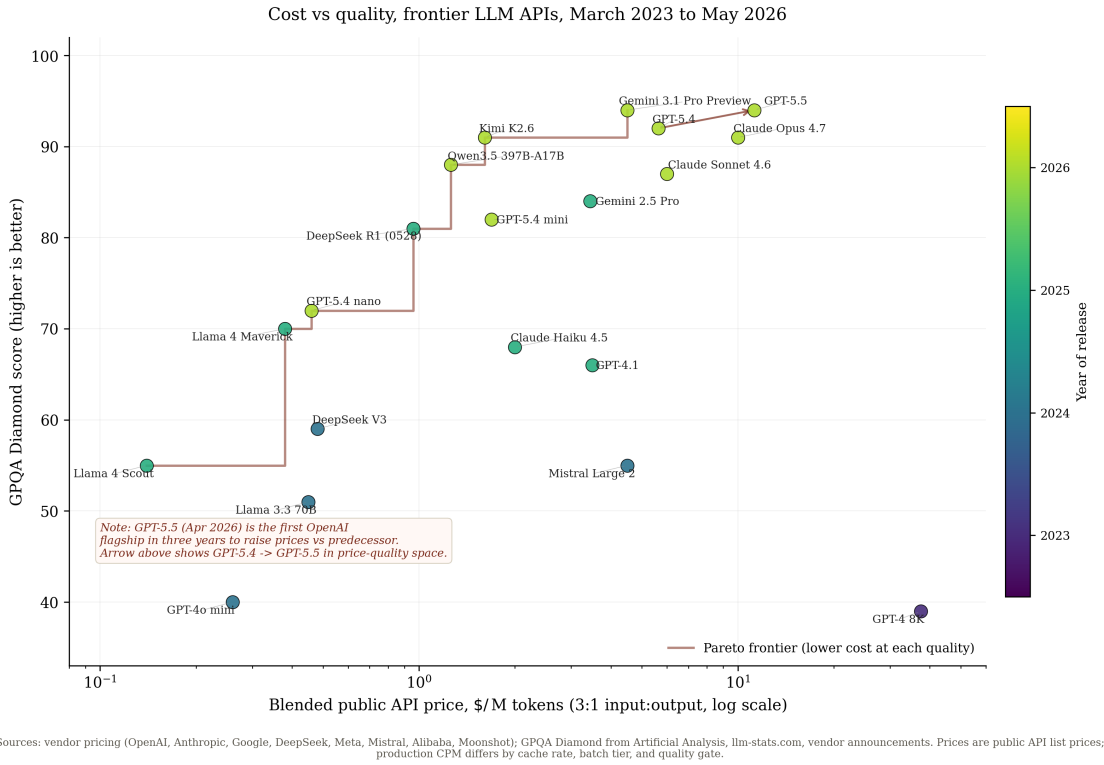


**Figure 2.** Cost-quality Pareto frontier across frontier LLM APIs, March 2023 to May 2026. Each dot is one model. X-axis is blended public-API price (3:1 input/output mix, Epoch AI convention) on a logarithmic scale. Y-axis is GPQA-Diamond score. Color encodes year of release. The Pareto frontier curve marks the cheapest model available at or above each quality bin. Two structural facts emerge: the frontier moves down and to the right over time, but unevenly; and at the top-right corner GPT-5.5 sits above GPT-5.4 in price, breaking monotonic decline. Sources: Artificial Analysis, llm-stats.com, vendor pricing pages.

server (AI inference) electricity growing roughly 30 percent per year.

The constraint that decides who deploys frontier inference is increasingly grid power and water per square mile, not chips per dollar. GB200 NVL72 racks dissipate roughly 130 kW each and require liquid cooling; the next-generation Rubin platform pushes per-rack dissipation higher still. Hyperscaler capex for 2026 is forecast at over $700 billion across the four largest U.S. operators, more than double 2024's level.

## 5. Quantization: lower memory traffic, not free accuracy

The first operational lever is quantization. The important production pattern is weight-only quantization: store weights in a lower-bit format while keeping activations in higher precision. In W4A16, weights are 4-bit integers and activations remain 16-bit. Moving from FP16/BF16 weights to 4-bit weights cuts weight storage by roughly 75 percent in the idealized case. End-to-end memory reduction is smaller because the KV cache, activations, framework overhead, and batching policy still bind.

AWQ, or Activation-aware Weight Quantization, exploits the observation that only a small fraction of weights are sensitive. The AWQ paper reports that protecting roughly 1

percent of salient weights substantially reduces quantization error, identifying these channels via activation statistics rather than weight magnitude. GPTQ takes a different path: one-shot post-training quantization with approximate second-order information, achieving 3-bit and 4-bit quantization of large GPT-family models with negligible degradation and end-to-end speedup over FP16 on A100/A6000-class GPUs.

Quantized weights produce production gains only when the serving path uses kernels that avoid surrendering the saving back to dequantization overhead. This is what Marlin contributes. Marlin reports up to 4x at batch 16–32 (W4A16) and approximately 2.8x end-to-end with vLLM. The RTX 3090 actually outpaces A100 here because A100's bandwidth advantage shrinks the memory-bound win. On NVIDIA Ampere, Ada, Hopper, and Blackwell, vLLM's quantization documentation lists AWQ, GPTQ, and Marlin paths among supported formats.

NVFP4 is the most consequential quantization development of 2025–2026 and warrants specific mention. NVFP4 is NVIDIA's 4-bit floating-point format introduced with Blackwell. On GB200 and B200 with TensorRT-LLM, NVFP4 inference of DeepSeek-R1 reaches over 3x the throughput of H200 FP8 at 2.5x lower per-user latency, with a 3.5x reduction in memory footprint versus FP16 (1.8x versus FP8) and accuracy degradation typically below one percent on language-modeling benchmarks. SemiAnalysis's open-source InferenceMAX reports B200 reducing cost-per-million-tokens by approximately 15x relative to H200 on gpt-oss workloads at matched quality, conditioned on configuration and post-launch software optimization.

The engineering rule is not "always use AWQ." It is: benchmark W4A16/AWQ/GPTQ/FP8/NVFP4 on the exact hardware, model, batch regime, and quality suite you will serve. A quantized-kernel path is a reasonable default candidate to test before shipping a full-precision server.

## 6. Serving runtime: memory management, scheduling, and disaggregation

The second operational lever is the runtime. Autoregressive inference stresses systems in a specific way: prefill is compute-bound, decode is often memory-bandwidth-bound, request lengths vary, and the KV cache grows and shrinks dynamically.

PagedAttention addressed a central memory-management problem. In vLLM, KV-cache memory is managed in blocks analogous to virtual memory pages. This reduces fragmentation and permits sharing of key-value blocks across requests. The vLLM paper reports near-zero KV-cache waste and 2–4x throughput improvement at the same latency compared with prior serving systems such as FasterTransformer and Orca.

Continuous batching, also called iteration-level scheduling, attacks a different bottleneck. Static batching forces the whole batch to wait for the slowest request. Orca instead schedules at the granularity of generation iterations, so completed requests can leave and new requests can enter without waiting. Orca's OSDI paper reports a 36.9x throughput improvement over FasterTransformer at the same latency on a GPT-3 175B serving setup, against a specific baseline. Sarathi-Serve adds chunked-prefill scheduling and reports throughput gains of 2.6x to 6.9x depending on model and traffic mix.

The most consequential 2025–2026 runtime advance is disaggregated prefill/decode serving. Prefill is compute-bound and decode is memory-bound; running both phases on the same homogeneous GPU pool wastes one or the other. Mooncake (FAST 2025 best paper) disaggregates prefill and decode onto separate GPU clusters with a KV-cache pool spanning CPU DRAM and SSD, reporting up to 5.25x throughput in simulation and 75 percent more

requests served in production for Kimi. DistServe (OSDI 2024) reports 7.4x more requests or 12.6x tighter SLO at 90 percent success versus prior systems. Microsoft Splitwise reports 1.4x throughput at 20 percent lower cost. NVIDIA Dynamo, released at GTC 2025, generalizes the pattern with disaggregated prefill/decode, KV-cache-aware request routing, and inflight KV migration on top of vLLM, TensorRT-LLM, and SGLang; on GB200 NVL72 serving DeepSeek-R1, Dynamo reports up to 30x more requests served versus the prior baseline.

The kernel layer evolved as well. FlashAttention-3, optimized for Hopper with asynchronous warp specialization and FP8 paths, reaches approximately 740 TFLOPS FP16 (75 percent of H100 peak, up from 35 percent for FlashAttention-2) and 1.2 PFLOPS FP8 with 2.6x lower numerical error than baseline FP8 attention. FlashAttention-4 targets Blackwell directly and reports up to 22 percent improvement over NVIDIA cuDNN attention on B200. On memory-bound serving paths these compound multiplicatively with the runtime wins above.

Speculative decoding adds a decoding-time parallelism lever. A small draft model proposes a short continuation. The large target model verifies multiple candidate tokens in one pass. The original speculative decoding work reports 2–3x acceleration on T5-XXL with identical outputs, and DeepMind's speculative sampling paper reports 2–2.5x speedup on 70B Chinchilla without quality compromise. The current production state of the art is EAGLE-3, which reports 3.0x to 6.5x speedup over autoregressive baselines and 20–40 percent improvement over EAGLE-2, with approximately 80 percent draft acceptance on aligned target models. Speculative decoding became production default in vLLM, TensorRT-LLM, and SGLang during 2025.

These techniques compound but not linearly. The bottleneck shifts as each lands. Quantization reduces weight movement. PagedAttention improves KV-cache packing. Continuous batching lifts occupancy under heterogeneous request lengths. Disaggregation prevents one phase from starving the other. Speculation reduces expensive target-model passes. FlashAttention-3 and 4 push the kernel layer closer to peak. A serving stack that gets all of these right can be dramatically cheaper than a naive PyTorch/Hugging Face loop. The only honest number is the one measured under the production traffic distribution.

## 7. Hardware: GPUs remain central, but inference is contested

The hardware story is not "NVIDIA lost inference" or "ASICs replaced GPUs." The accurate claim is that inference made specialization economically attractive. Once a workload stabilizes, buyers can optimize for tokens per watt, tokens per dollar, memory locality, networking, and software support.

Inference-optimized chips and accelerators have entered from Meta, Google, Amazon, Intel, AMD, Qualcomm, Groq, SambaNova, Cerebras, Graphcore, and others. This does not eliminate GPU clusters. It creates a heterogeneous procurement problem. GPUs retain advantages in flexibility, ecosystem maturity, training, post-training, and fast model churn. ASICs and inference specialists become attractive when workloads are predictable, batchable, and large enough to amortize integration cost.

For engineers, the decision variable is not peak TOPS. Peak TOPS usually ignores memory bandwidth, interconnect, KV-cache behavior, software support, and the cost of hitting latency SLOs. The correct benchmark is delivered VCpD at a fixed quality target, context length, concurrency distribution, and latency SLO. SemiAnalysis InferenceMAX,

the most credible open-source benchmark in this space as of mid-2026, reports GB200 NVL72 delivering approximately $0.123 per million tokens at 116 tokens-per-second-per-user on DeepSeek-R1; B200 NVFP4 reduces cost-per-token by roughly 15x versus H200 FP8 on the same workload. At hyperscaler scale: AWS reports more than 500,000 Trainium2 chips deployed in late 2025 with Trainium3 and a 144-chip UltraServer in development; Google's Ironwood TPU was announced at Cloud Next 2025 as the first TPU purpose-built for inference; Microsoft's Maia 200 and Meta's MTIA 300/400 lineup target the same workload class. The number to watch in 2027 is delivered VCpD at a fixed quality bar, not peak FP4 TFLOPS in vendor decks.

## 8. Architecture: long context pushed the stack beyond pure attention

Pure Transformer attention was the default production architecture from roughly 2017 through 2024. It remains powerful, but long-context serving exposes the cost of KV-cache growth. At 128K-plus contexts, KV cache can dominate memory and limit batch size.

State-space models such as Mamba offer a different scaling profile. The Mamba paper reports linear scaling in sequence length and 5x higher inference throughput than Transformers in its setting. Hybrid architectures combine attention with state-space layers: attention is retained where global mixing is valuable, while linear-complexity layers carry much of the sequence-processing burden.

Jamba-1.5-Large is the clean reference case. The Jamba-1.5 paper describes a hybrid Transformer-Mamba mixture-of-experts model with 398B total parameters, 94B active, and 256K-token context. It reports roughly an order-of-magnitude reduction in KV-cache memory at 256K context (4 GB for Jamba-1.5-Large versus 32 GB for Mixtral and 128 GB for Llama-2-70B at the same context length). DeepSeek-V3 takes a different path with Multi-Head Latent Attention (MLA), which compresses the KV cache by projecting keys and values into a smaller latent space; DeepSeek-V3.2 layers Sparse Attention on top to reduce attention complexity from $O(L^2)$ to $O(Lk)$ at long context.

The practical conclusion is not that every production model should be Mamba-like or MLA-based. It is that long-context architecture and serving architecture must be designed together. A retrieval-heavy 8K system, a 256K document-analysis system, and a real-time voice agent should not share the same default inference assumptions.

## 9. Hallucination as a verification cost in VCpD

The VCpD denominator includes verification cost. This is not a rhetorical move. Production reliability is part of delivered token quality, and a confidently-wrong token can be more expensive than no token.

Kalai et al. (2025), an OpenAI-led collaboration with Georgia Tech, argue in "Why Language Models Hallucinate" that hallucinations persist partly because standard training and benchmark procedures reward guessing over calibrated uncertainty. Stanford HAI's legal-domain work (Dahl et al., January 2024) found hallucination rates ranging from 69 to 88 percent on specific legal queries for GPT-3.5, Llama 2, and PaLM 2. A more recent Stanford follow-up on legal RAG tools using GPT-4-class models reports rates closer to 17 percent: still significant but a sharp improvement. These numbers should not be generalized to every model or every domain, but the pattern is clear: hallucination rates vary sharply by task, model, data availability, and verification surface.

The production mitigation is a system, not a prompt: abstention-aware benchmarking, retrieval with source constraints, span-level verification, uncertainty surfacing, domain-specific test sets, and a human escalation path for high-risk outputs. Each is a real cost. VCpD makes that cost legible.

## 10. Engineering implications

1. **Compute VCpD, not CPM, when comparing model options.** Quality and verification cost belong in the unit. A cheaper model with higher hallucination rate may have lower VCpD than a more expensive reliable model.
2. **Benchmark quantized serving before shipping full precision.** W4A16, AWQ, GPTQ, FP8, NVFP4 should be treated as candidates, not slogans. The best choice depends on model family, hardware, batch size, context length, and quality-test sensitivity.
3. **Profile prefill and decode separately.** The bottleneck during prefill is not necessarily the bottleneck during decode. Track TTFT, TPOT, queueing delay, KV-cache occupancy, accepted speculative tokens, and tokens per watt. Disaggregated serving gates open when these diverge.
4. **Long-context systems are not prompt-length extensions.** At 128K-plus contexts, architecture, retrieval, KV-cache layout, prefix caching, and verification become one design problem.
5. **Treat factuality as part of serving quality.** Production inference should measure not only latency and throughput, but also abstention, citation accuracy, retrieval coverage, and verified answer rate.

## 11. Conclusion

The inference stack in 2026 is not one breakthrough. It is a compound curve. Public API prices fell because models became smaller and better, quantized serving became practical, kernels improved (FlashAttention-3 to FlashAttention-4), KV-cache memory was managed more intelligently (PagedAttention, MLA, Sparse Attention), schedulers stopped wasting batches (Orca, Sarathi-Serve, disaggregated serving), speculation reduced serial decode cost (EAGLE-3), and hardware competition moved from peak FLOPs to delivered VCpD.

The next engineering regime will be defined less by whether inference becomes cheaper in the abstract and more by how precisely teams can trade off cost, latency, context, reliability, and verification. The framework we propose, VCpD, makes that tradeoff explicit and decomposable. The systems that win will not simply generate cheaper tokens. They will generate higher VCpD under production constraints.

## References

[1] OpenAI. "GPT-4." March 14, 2023. `https://openai.com/index/gpt-4-research/`

[2] OpenAI. "GPT-4o mini: advancing cost-efficient intelligence." July 18, 2024. `https://openai.com/index/gpt-4o-mini-advancing-cost-efficient-intelligence/`

[3] OpenAI. "API Pricing." Accessed May 3, 2026. `https://openai.com/api/pricing/`

[4] Stanford Institute for Human-Centered AI. "The 2025 AI Index Report." 2025. https://hai.stanford.edu/ai-index/2025-ai-index-report

[5] NVIDIA. "Rethinking AI TCO: Why Cost per Token Is the Only Metric That Matters." April 15, 2026. https://blogs.nvidia.com/blog/lowest-token-cost-ai-factories/

[6] Deloitte. "Why AI's next phase will likely demand more computational power, not less." November 18, 2025. https://www.deloitte.com/us/en/insights/industry/technology/technology-media-and-telecom-predictions/2026/compute-power-ai.html

[7] McKinsey. "The future of AI workloads." February 24, 2026. https://www.mckinsey.com/featured-insights/week-in-charts/the-future-of-ai-workloads

[8] International Energy Agency. "Energy and AI." 2025. https://www.iea.org/reports/energy-and-ai

[9] Lin, J. et al. "AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration." arXiv:2306.00978, 2023. https://arxiv.org/abs/2306.00978

[10] Frantar, E., Ashkboos, S., Hoefler, T., and Alistarh, D. "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers." arXiv:2210.17323, 2022. https://arxiv.org/abs/2210.17323

[11] Frantar, E. et al. "MARLIN: Mixed-Precision Auto-Regressive Parallel Inference on GPUs." arXiv:2408.11743, 2024. https://arxiv.org/abs/2408.11743

[12] NVIDIA. "Introducing NVFP4 for Efficient and Accurate Low-Precision Inference." 2025. https://developer.nvidia.com/blog/introducing-nvfp4-for-efficient-and-accurate-low-precision-inference/

[13] Kwon, W. et al. "Efficient Memory Management for Large Language Model Serving with PagedAttention." SOSP 2023. arXiv:2309.06180. https://arxiv.org/abs/2309.06180

[14] Yu, G.-I. et al. "Orca: A Distributed Serving System for Transformer-Based Generative Models." OSDI 2022. https://www.usenix.org/conference/osdi22/presentation/yu

[15] Agrawal, A. et al. "Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve." OSDI 2024. arXiv:2403.02310. https://arxiv.org/abs/2403.02310

[16] Zhong, Y. et al. "DistServe: Disaggregating Prefill and Decoding for Goodput-optimized Large Language Model Serving." OSDI 2024. arXiv:2401.09670. https://arxiv.org/abs/2401.09670

[17] Patel, P., Choukse, E. et al. "Splitwise: Efficient Generative LLM Inference Using Phase Splitting." ISCA 2024. arXiv:2311.18677. https://arxiv.org/abs/2311.18677

[18] Qin, R. et al. "Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving." FAST 2025 Best Paper. arXiv:2407.00079, 2024. https://arxiv.org/abs/2407.00079

[19] NVIDIA. "Introducing NVIDIA Dynamo." GTC 2025. https://developer.nvidia.com/blog/introducing-nvidia-dynamo-a-low-latency-distributed-inference-framework-for-scaling-reason

[20] Shah, J., Bikshandi, G., Dao, T. et al. "FlashAttention-3: Fast and Accurate Attention with Asynchrony and Low-precision." arXiv:2407.08608, 2024. https://arxiv.org/abs/2407.08608

[21] Leviathan, Y., Kalman, M., and Matias, Y. "Fast Inference from Transformers via Speculative Decoding." arXiv:2211.17192, 2022. https://arxiv.org/abs/2211.17192

[22] Chen, C. et al. "Accelerating Large Language Model Decoding with Speculative Sampling." arXiv:2302.01318, 2023. https://arxiv.org/abs/2302.01318

[23] Li, Y. et al. "EAGLE-3: Scaling up Inference Acceleration of Large Language Models via Training-Time Test." NeurIPS 2025. arXiv:2503.01840. https://arxiv.org/abs/2503.01840

[24] Gu, A. and Dao, T. "Mamba: Linear-Time Sequence Modeling with Selective State Spaces." arXiv:2312.00752, 2023. https://arxiv.org/abs/2312.00752

[25] Lieber, O. et al. "Jamba-1.5: Hybrid Transformer-Mamba Models at Scale." arXiv:2408.12570, 2024. https://arxiv.org/abs/2408.12570

[26] DeepSeek-AI. "DeepSeek-V3 Technical Report." arXiv:2412.19437, December 2024. https://arxiv.org/abs/2412.19437

[27] DeepSeek-AI. "DeepSeek-V3.2 with Sparse Attention." arXiv:2512.02556, December 2025. https://arxiv.org/abs/2512.02556

[28] Kalai, A. T., Nachum, O., Vempala, S. S., and Zhang, E. "Why Language Models Hallucinate." arXiv:2509.04664, 2025. https://arxiv.org/abs/2509.04664

[29] Dahl, M. et al. (Stanford HAI). "Hallucinating Law: Legal Mistakes with Large Language Models are Pervasive." January 11, 2024. arXiv:2401.01301. https://hai.stanford.edu/news/hallucinating-law-legal-mistakes-large-language-models-are-pervasive

[30] MIT FutureTech. "The Price of Progress: Algorithmic Efficiency and the Falling Cost of AI Inference." arXiv:2511.23455, November 2025. https://arxiv.org/abs/2511.23455

[31] Erdil, E. (Epoch AI). "Inference Economics of Language Models." arXiv:2506.04645, June 2025. https://arxiv.org/abs/2506.04645

[32] SemiAnalysis. "InferenceMAX: Open-Source Inference Benchmarking." 2026. https://newsletter.semianalysis.com/p/inferencemax-open-source-inference