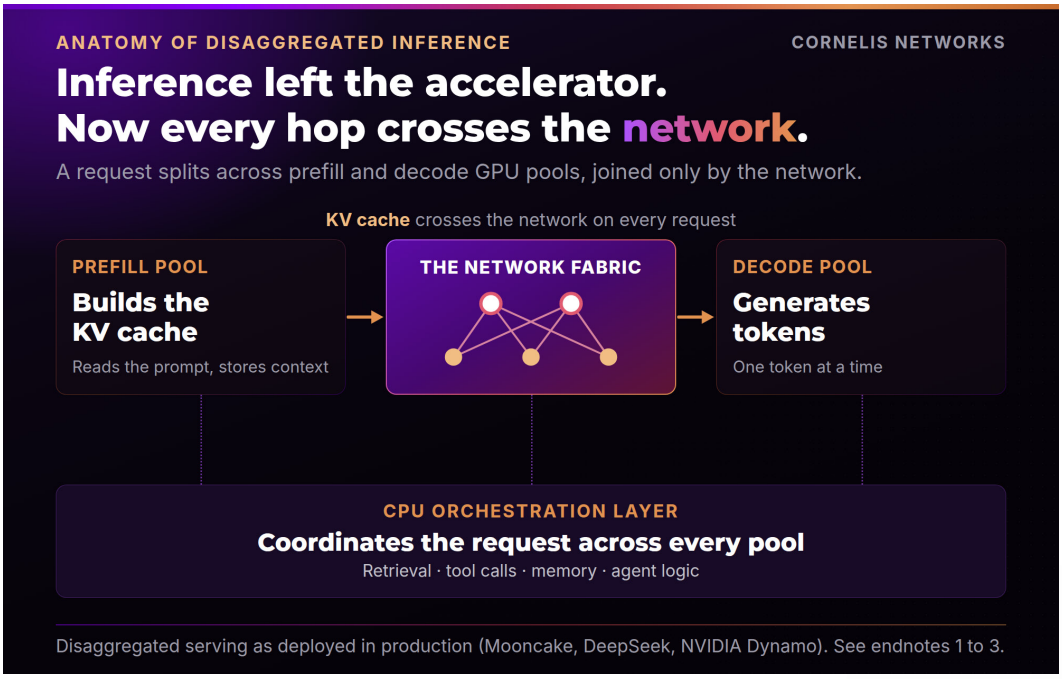
INFERENCE ECONOMICS ARE NETWORK ECONOMICS

Inference Is Where The Money Is Made

Hyperscalers, data center operators, and a fast-growing set of enterprises have spent the last several years building infrastructure to train AI models. They earn that investment back by running those models in production, the work the industry calls inference. In practice, that means turning each request into a response one token at a time, fast enough to hold the latency and service-level commitments a customer pays for. The prize is large and growing fast. Fortune Business Insights values the AI inference market at $104 billion in 2025 and projects it to roughly triple, to $313 billion, by 2034.[1]

The return on inference investment is measured in numbers every operator already tracks: cost per token, service uptime, and the latency a user feels, from the first token to the last. Inference behaves nothing like training, and the economics reflect the difference.

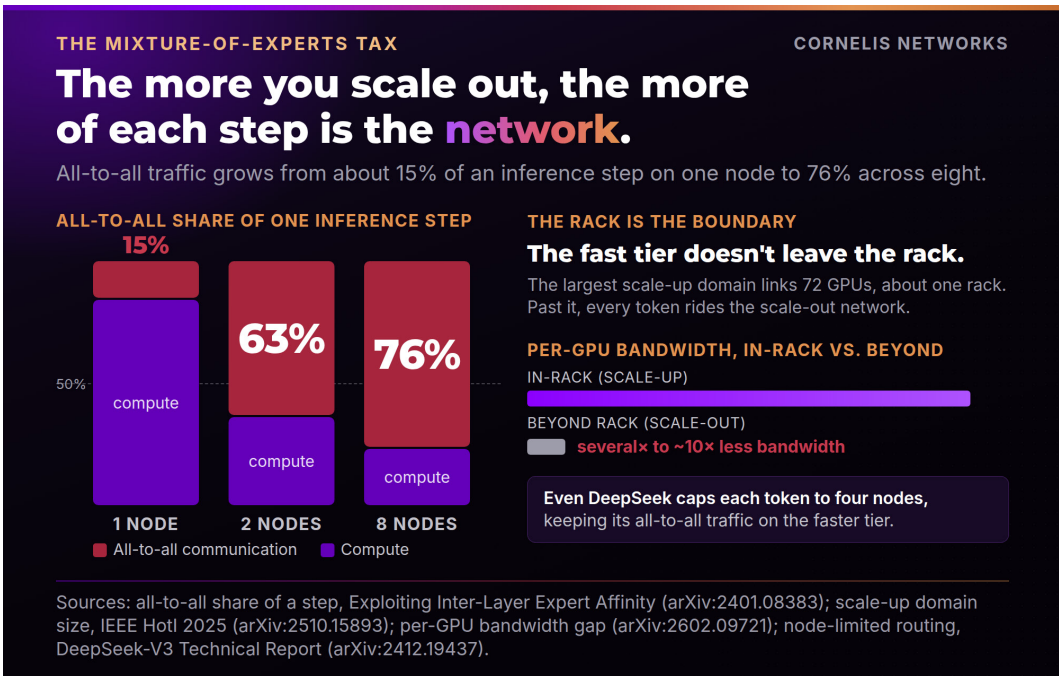


Consider how the work has evolved. A production inference system no longer runs one model on one accelerator. It splits the job into stages and runs each on the hardware best suited to it. A prefill pool reads the prompt and builds the context, the key-value (KV) cache the model draws on. A decode pool generates the answer one token at a time. A tier of CPUs orchestrates the rest: retrieval, tool calls, memory, and the logic that connects an agent's steps.

Moonshot AI runs its Kimi service on exactly this split and reports handling more than twice the requests of the system it replaced.[2] DeepSeek serves its flagship model from separate prefill and decode pools spanning dozens of nodes on each side.[3] Even the largest accelerator vendors now ship software dedicated solely to coordinating this disaggregated handoff.[4]

Every one of those stages hands data to the next across the network. The fabric is no longer passive infrastructure sitting beneath the compute. It has become part of the compute path, and its behavior appears directly in the cost of service.

The traffic pattern increases the challenge for the network. Inference is small-message-heavy, latency-sensitive, and bursty, the opposite of training's dense and predictable flow. Mixture-of-experts (MoE) models intensify it. Every forward pass routes tokens to a different subset of experts scattered across the cluster, which forces an all-to-all exchange on the network at each step. In GPT MoE inference, that all-to-all traffic runs near 15% of total time on a single node and climbs to 76% across eight nodes, at which point serving is almost entirely communication-bound.[5] The exchange is synchronous, so the fastest accelerator in the pool waits on the slowest transfer before it can proceed.



The fast tier does not stretch far. Passive copper interconnects reach only about a meter, which holds the scale-up domain to a single rack, and the largest such domains in volume deployment today link 72 GPUs.[6] Reach past that rack and you land on the scale-out network, where per-GPU bandwidth runs several times lower, as much as seven times on the latest accelerators.[7] Frontier MoE models and disaggregated serving both outgrow a single rack, so the traffic that decides cost and latency rides the slower fabric. Teams already build around this boundary. DeepSeek caps each token to four nodes to keep its all-to-all traffic on the faster path.[8] The scale-out network, not the accelerator, sets the ceiling on what an operator can build and how cheaply it can serve. When the fabric stalls, the token stream stalls with it, the latency budget breaks, and the operator absorbs the result in higher cost per token.

Optimizing The Model Only Goes So Far

As inference costs climb, most teams turn first to the model layer. They quantize weights to shrink them, batch requests more efficiently, and apply techniques such as speculative decoding to extract more tokens from each pass. When those gains are exhausted, they add accelerators. Each of these tactics helps, and all of them are worth doing.

Unfortunately, these measures miss where a growing share of the cost now resides. The expensive problem is no longer the computation inside a single accelerator. It is the movement between the pools that disaggregation created. A decode step cannot begin until the prefill pool's KV cache arrives, and that cache crosses the network on every request. When the transfer lags, time to first token climbs, users wait, and no amount of accelerator speed recovers the lost time.

The interconnects most teams already own make this harder still. They were selected for training, where traffic is steady and predictable. The dynamic, sparse, bursty pattern of MoE and agentic inference is fundamentally different, and a fabric optimized for one is not necessarily suited to the other.

The spending is inverted from the leverage. According to a study by Epoch AI, the fabric represents 13% of the cluster's cost, yet it governs the yield on the other 87%.[9] A team that adds accelerators to a congested fabric is buying more capacity to sit idle. Improving the network is the rare optimization where such a small share of the budget moves the performance of the entire system.

The all-to-all exchange in an MoE forward pass is a collective operation. So is the KV cache handoff between prefill and decode. Both are work. A passive fabric can only move that work faster. A fabric with compute in the data path can perform it. Reductions and gathers execute at the switch instead of round-tripping to an accelerator. The transfer stops being a wait state and becomes a step of the computation.

That is the difference between a faster pipe and an active fabric. Bandwidth improves the first. Only the second changes what the network contributes.

Tail Latency Sets the Price

The cost of a slow fabric does not stay small. It compounds with every token served. Consider how inference is priced today. The same model can carry a 6x spread in cost per million tokens, and that spread tracks one factor above all others: the latency guarantee. Lower-priced tiers carry tail-latency caveats. Premium tiers charge a clear markup for holding their service-level commitment under load.[10] The provider that keeps latency steady under heavy, unpredictable traffic sells the more valuable product.

Tail latency is where that economic erosion becomes visible. The figure every provider advertises is the median, the comfortable case. The figure that determines profitability is the 99th percentile, the response under a concurrency spike or an unusually long input. High tail latency breaks service-level agreements, triggers timeouts, and forces teams to overprovision hardware to cover the worst case.[11] Every coping mechanism is a network cost surfacing elsewhere: the extra decode capacity, the capped context window, the buffer against tail spikes. Each one is premium-token revenue the operator never gets to earn.



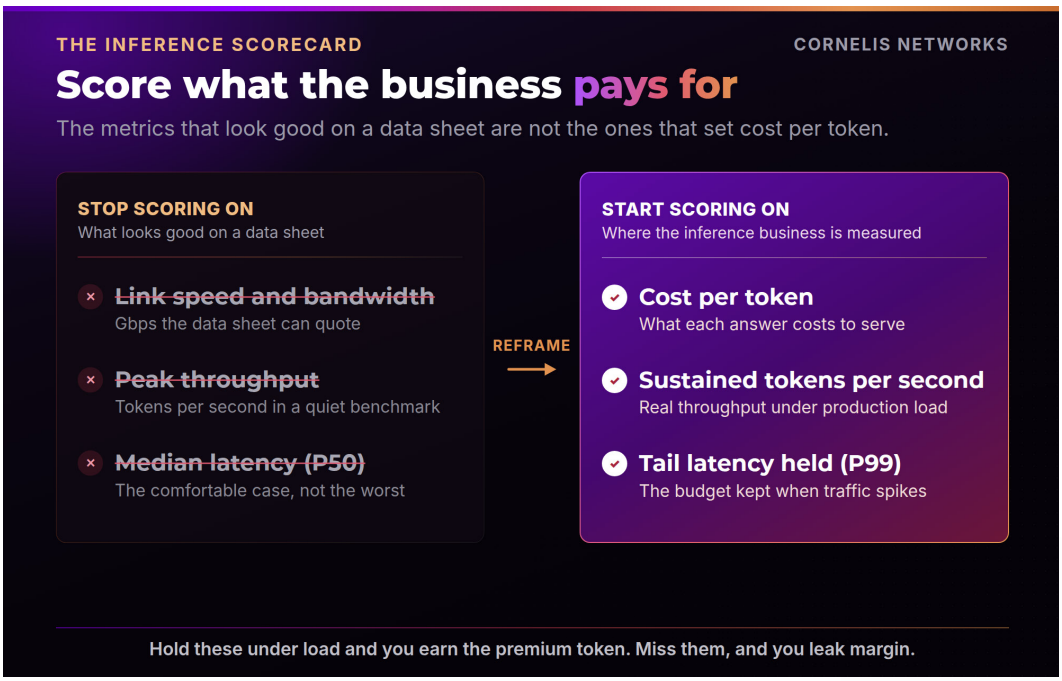
What To Expect From Your Infrastructure

If the network sets the economics of inference, buyers should evaluate suppliers on those terms. The list is short:

- Predictable tail latency under bursty, small-message load, the condition that actually stresses the network.
- Congestion-free movement of the KV cache between prefill and decode pools, so generation never waits on the fabric.
- A fabric that adapts to the shifting traffic of MoE and agentic serving, rather than one tuned for the steady collectives of training.
- Open standards, so accelerators can be matched to the economics of each workload instead of locked to one vendor for the life of the deployment.

These requirements share a thread. Raw bandwidth is rarely the limit. Modern links provide it in abundance. What matters is how the fabric behaves under pressure, whether it delivers small, urgent transfers on time when thousands of requests arrive at once.

This shift also demands a change to the scorecard. The industry can stop grading inference infrastructure on raw speed and begin grading it where business value is measured: tokens per second, cost per token, and whether the network holds its latency budget under heavy load. A network that moves those numbers positively is earning its place.



That standard does more than hold down cost. A network that keeps latency flat under load makes faster, higher-value responses possible, which turns steady latency into a revenue lever.

The Network Is The Lever

Inference economics are network economics. The measures that decide whether AI pays for itself all run through the fabric connecting the pools. The operator that owns that fabric owns its cost per token and earns the right to price the premium tokens that steady, low-latency service makes possible.

Compute will continue to advance. It will not, on its own, resolve an economic problem that lives in the movement between processors.

That movement is not overhead. The all-to-all exchange in an MoE pass and the KV cache handoff between pools are both work the system has to do. A passive fabric can only carry that work faster. A fabric with programmable compute at every hop performs it. A fabric either computes or it does not, and only one of those answers changes the cost per token.

Cornelis calls the second one the Active Compute Fabric. It is open, it spans scale-up and scale-out, and it is built on a foundation proven in production today at national laboratory scale.

That is the problem Cornelis is built to address. As inference becomes the predominant workload, the network sets the price of everything an AI operator sells.

References

1. Fortune Business Insights, AI Inference Market Size, Share & Industry Analysis, By Hardware, Deployment, Application, End-user, and Regional Forecast, 2026–2034, last updated July 27, 2026. The global AI inference market was valued at $103.73 billion in 2025, is put at $117.80 billion in 2026, and is projected to reach $312.64 billion by 2034, a CAGR of 12.98%. fortunebusinessinsights.com
2. Moonshot AI, Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving. Kimi runs on separated prefill and decode clusters with a disaggregated KV-cache pool; reported roughly 115% and 107% more requests handled on A800 and H800 clusters versus the prior system. arXiv:2407.00079
3. DeepSeek-AI, DeepSeek-V3 Technical Report. Production serving splits prefill and decode into separate node pools, with wide expert parallelism on the decode side. arXiv:2412.19437; deployment detail in LMSYS, Deploying DeepSeek with PD Disaggregation...
4. NVIDIA, Introducing NVIDIA Dynamo, a distributed inference framework built specifically to serve disaggregated prefill and decode at scale (reported up to 7x higher throughput on DeepSeek-R1). Cited as evidence that the field is standardizing on disaggregated serving, not as an architecture to emulate. developer.nvidia.com
5. J. Yao et al., Exploiting Inter-Layer Expert Affinity for Accelerating Mixture-of-Experts Model Inference. All-to-all communication rises from roughly 15% of GPT MoE inference time on a single node to 76% across eight nodes, where inference becomes almost purely communication-bound. arXiv:2401.08383
6. M. Bernadskiy et al., Accelerating Frontier MoE Training with 3D Integrated Optics, 2025 IEEE Symposium on High-Performance Interconnects (HotI). Passive electrical interconnects reach roughly one meter, limiting the scale-up domain to a single rack; the Blackwell DGX pod linked 72 GPUs in 2024. arXiv:2510.15893
   NVIDIA, GB200 NVL72. The system "connects 36 Grace CPUs and 72 Blackwell GPUs in a rack-scale, liquid-cooled design," forming "a 72-GPU NVIDIA NVLink domain that acts as a single, massive GPU." nvidia.com
7. G. Liu et al., Revealing the Challenges of Attention-FFN Disaggregation for Modern MoE Models and Hardware Systems, Baidu. Scale-up bandwidth runs several times the amortized scale-out bandwidth per GPU (3.2× on H800, 7.2× on H20, higher on recent parts). arXiv:2602.09721
8. DeepSeek-AI, DeepSeek-V3 Technical Report. "We limit each token to be dispatched to at most 4 nodes." arXiv:2412.19437
9 Servers account for 60% of the total cost of ownership of a one-gigawatt AI data center. Epoch AI
10. Digital Applied, AI Inference Providers: Q2 2026 Pricing Matrix. Cost per million tokens varies up to roughly sixfold for the same model across service tiers, with the spread tied to latency and tail-latency service-level guarantees. digitalapplied.com
11. Spheron, AI Inference Cost Economics in 2026, and BentoML, Beyond Tokens-per-Second. P99 tail latency breaks SLAs, triggers timeouts, and forces GPU overprovisioning; median (P50) figures conceal it. spheron.network

Learn more about industry leading AI and HPC scale-out network at cornelis.com

CORNELIS™

Other names and brands may be claimed as the property of others. All information provided here is subject to change without notice. Contact your Cornelis Networks representative to obtain the latest Cornelis Networks product specifications and roadmaps. Cornelis Networks technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Copyright © 2026, Cornelis Networks. All rights reserved.