MoE and the Network Cliff

MoE Efficiency Gains are Real, but Routing Overhead can become the Hidden Bottleneck

Author: Phil Marshall, PhD

Publish Date: June, 2026

Mixture-of-Experts architectures are often viewed as a more efficient alternative to conventional dense AI models. A useful analogy is a hospital with specialist departments. Rather than sending every patient through the entire hospital, a coordinator directs each case to the specialists best suited to handle it. MoE works in much the same way: a router sends each token to a small subset of expert networks rather than engaging the entire model.

This selective routing can increase model capacity without increasing computation in direct proportion. However, it also introduces a routing tax in both training and inference. Tokens must be dispatched to the selected experts, processed, and returned before the model can continue.

During training, this traffic is compounded by gradient synchronization and load-balancing requirements. During inference, particularly token-by-token decoding, routing latency can constrain response time and throughput. When experts are distributed across GPUs, servers, or racks, the resulting all-to-all communication can place substantial demands on the underlying infrastructure.

This trade-off is becoming more important as MoE gains traction among frontier AI developers. Google, Meta, DeepSeek, and xAI have publicly disclosed MoE-based models, while other developers are believed to be moving in a similar direction.

The key issue is that MoE does not eliminate infrastructure complexity. It shifts part of the burden from computation to communication.

© Copyright Tolaga Research, 2026

In a dense model, every token follows broadly the same computational path. In an MoE model, tokens must first be assigned to experts, transferred to the devices hosting those experts, processed, and then returned to the main model flow. When the experts are distributed across multiple GPUs, this creates an all-to-all communication pattern in which many devices exchange data before computation can proceed.

This distribution of experts across GPUs, servers, or racks is known as expert parallelism. It allows the model to scale to many more experts than would fit efficiently on a single device, but it also means that tokens may need to move across the system to reach the experts selected by the router.

As expert parallelism increases, the communication expands across more devices and more network tiers. Traffic that initially remains within a GPU server may begin to cross server and rack boundaries, where bandwidth is typically lower and latency is higher.

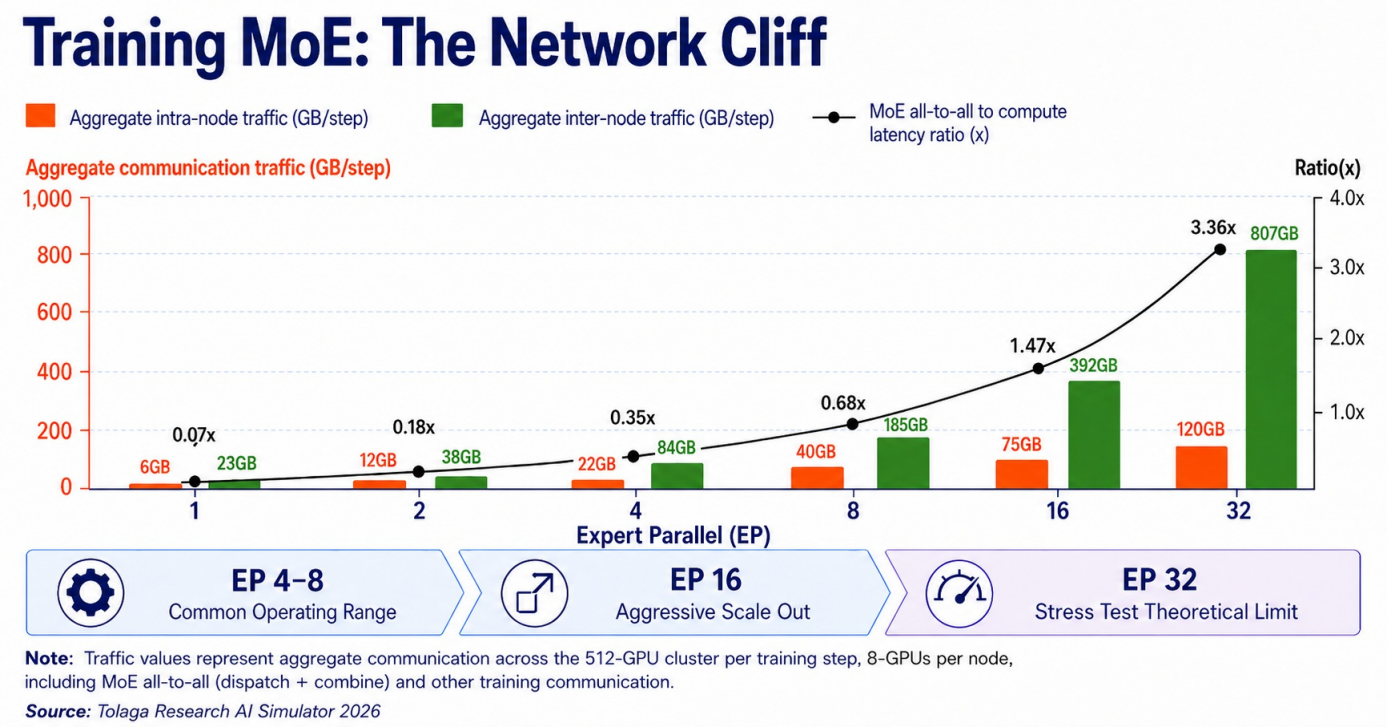
The challenge becomes greater when MoE is combined with other forms of distributed parallelism. Tensor parallelism introduces synchronization collectives. Data parallelism requires gradient synchronization. Pipeline parallelism adds coordination between stages. MoE routing adds dispatch-and-combine traffic on top of these existing flows.

Together, these workloads compete for shared capacity across NVLink, NVSwitch, InfiniBand, Ethernet, and custom interconnect fabrics. Expert load imbalance can add further delay because the most heavily loaded expert may determine how quickly a training step, inference batch, or decode iteration completes. At modest scale, the overhead may remain manageable. As expert parallelism increases, however, routing latency, synchronization, network contention, and uneven expert loading can begin to offset the compute savings.

© Copyright Tolaga Research, 2026

How Expert Parallelism Creates the Network Cliff

The simulation results below show how communication pressure changes as expert parallelism increases from EP1 to EP32 for a large-scale MoE training workload distributed across 512 H100-class GPUs, configured as 64 nodes of 8 GPUs each. The simulated model uses 64 experts, with two experts activated for each token, allowing the analysis to isolate how wider expert distribution affects all-to-all traffic and training latency. Because each node contains 8 GPUs connected by fast intra-node interconnects, expert-parallel groups up to EP8 can largely remain within a single node; beyond that, communication increasingly crosses to slower inter-node fabric, which is the primary driver of the inter-node traffic growth shown below. Tensor-parallel traffic, which is typically kept intra-node due to its latency sensitivity, is a separate contributor to the intra-node totals shown.



Note: Traffic values represent aggregate communication across the 512-GPU cluster per training step, 8-GPUs per node, including MoE all-to-all (dispatch + combine) and other training communication.

Source: *Tolaga Research AI Simulator 2026*

At low levels of expert parallelism, communication remains relatively modest. Intra-node and inter-node traffic are present, but they do not dominate the training step.

Between EP 4 and EP 8, the system remains within a plausible operating range, although the ratio of MoE all-to-all latency to expert compute latency is already increasing. The architecture continues to deliver compute savings, but the supporting infrastructure is carrying a growing communication burden.

© Copyright Tolaga Research, 2026

The pressure becomes more visible at EP 16. Inter-node traffic approaches 400 GB per step, while the all-to-all-to-compute latency ratio reaches 1.47x. At this point, communication is no longer a secondary implementation detail. It has become a major component of the training step.

At EP 32, the simulation enters a theoretical stress-test regime. Inter-node traffic exceeds 800 GB per step, and the all-to-all-to-compute latency ratio rises to 3.36x. In this regime, the system can spend substantially more time moving tokens between experts than performing useful expert computation.

This is the MoE network cliff.

Why the Cliff Occurs

The cliff results from four interacting effects.

All-to-All Routing

Tokens are not necessarily assigned to experts located on the same GPU or within the same server. They must be dispatched to the appropriate experts and returned after processing, generating traffic in both directions.

Expanding Communication Boundaries

Higher expert parallelism spreads experts across more devices. As the expert pool expands, a larger share of routing traffic may cross GPU, server, and rack boundaries, where bandwidth is lower and latency is higher.

Competition with Other Training Traffic

MoE routing shares the interconnect with tensor-parallel collectives, pipeline transfers, and data-parallel gradient synchronization. These flows can contend for the same underlying network capacity.

Uneven Expert Loading

Tokens are not always distributed evenly across experts. If certain experts receive disproportionately large workloads, the most heavily loaded expert can determine the duration of the entire step. Average utilization may appear acceptable even as stragglers degrade overall performance.

These effects mean that MoE communication does not necessarily scale smoothly. As fabric utilization approaches saturation, contention, queueing, and synchronization delays can increase nonlinearly.

© Copyright Tolaga Research, 2026

A Hospital Analogy

MoE resembles a network of specialist departments.

The model works efficiently when cases can be directed quickly to nearby specialists, processed without delay, and returned to the broader workflow. In that environment, specialization improves both capability and efficiency.

The advantage begins to erode when referrals require excessive coordination, transport, waiting, or handoffs. As the specialist network grows, the referral process itself can become the bottleneck. A poorly coordinated specialist system may ultimately operate more slowly than a well-run general hospital, even when its individual specialists are more capable.

MoE has the same infrastructure dependency. Expert specialization improves theoretical compute efficiency only when routing, scheduling, load balancing, and networking are efficient enough to support it.

Implications for AI Infrastructure

MoE changes the core infrastructure question.

It is not enough to ask how many parameters are activated for each token. The more important question is whether the system can route tokens, synchronize devices, and balance expert workloads quickly enough to preserve the theoretical efficiency gain.

Network Performance Comes to the Fore

MoE deployments require more than high-performance accelerators. The bandwidth, latency, and topology of NVLink, NVSwitch, InfiniBand, Ethernet, and custom fabrics directly affect realized performance.

Topology Matters

The number of GPUs per node is a direct lever here: in our simulation, which included contention from tensor parallelism as well as MoE routing, an 8-GPU node meant expert-parallel groups up to EP8 could largely remain on fast intra-node interconnects, with the cliff emerging as EP scaled beyond that point. A cluster with fewer GPUs per node, or with more parallelism strategies competing for the same fabric, would likely hit the cliff earlier; one with more GPUs per node could push the usable range further out.

Routing and Scheduling Matter

Token routing, batching, capacity allocation, congestion management, and expert load balancing determine whether the available hardware is used efficiently. Small orchestration weaknesses can become significant at scale.

© Copyright Tolaga Research, 2026

Full-Stack Integration Matters

The strongest systems will treat model architecture, accelerator design, memory, networking, and scheduling as an integrated stack rather than as separate optimization problems.

Investor and Practitioner Takeaways

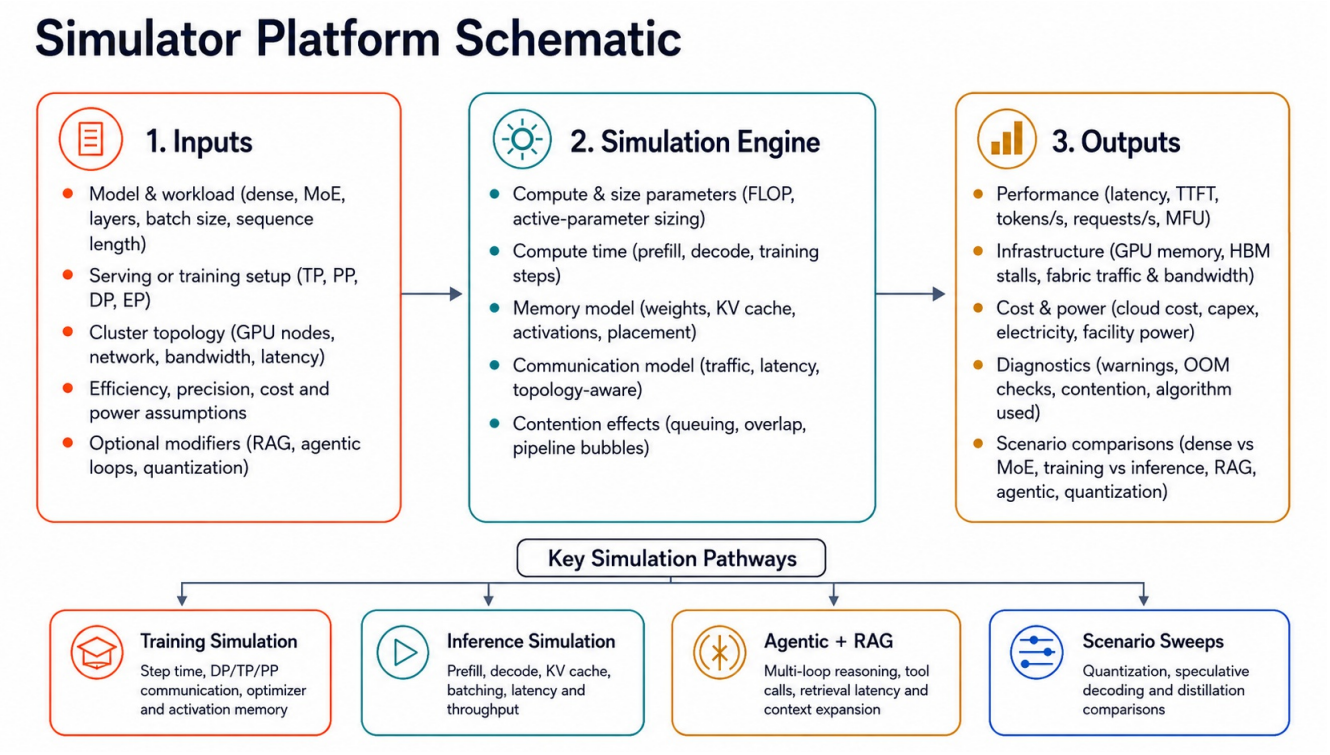
MoE efficiency gains are real, but they are not automatic.

A poorly optimized MoE deployment can underperform a dense model despite having superior theoretical compute efficiency. The risk increases when expert parallelism expands faster than the networking, topology, and orchestration systems supporting it.

Frontier AI infrastructure therefore cannot be evaluated using GPU count, model size, or peak FLOPS alone. MoE shifts part of the competitive advantage toward high-performance networking, topology-aware scheduling, load balancing, memory systems, and custom infrastructure integration.

The likely winners will not simply be those with the largest models or the most accelerators. They will be those that can coordinate distributed computation most efficiently under real workload, latency, and networking constraints.

Tolaga AI Infrastructure Simulator



LEARN MORE

© Copyright Tolaga Research, 2026

About the Author: Phil Marshall, PhD

Phil Marshall's work in AI began more than 30 years ago with Kohonen maps, discrete neural networks, machine learning, and fuzzy logic. More recently, his work has targeted AI-enabled solutions for the automotive, healthcare, manufacturing, and smart city sectors. Over the past decade, he has applied natural language processing and large language models to analyze online content and generate distinctive, data-driven industry research insights.

Marshall also brings deep experience in telecom, wireless, mobile, and networking technologies, fields he has worked in since the early 1990s. More recently, he has combined this infrastructure background with advanced software and modeling capabilities to develop Tolaga's AI Infrastructure Simulator, which is used to analyze the performance, cost, and scalability of emerging AI architectures.

Marshall is a dual citizen of the United States and New Zealand and currently resides in Auckland, New Zealand.

Contact: philip.marshall@tolaga.com

© Copyright Tolaga Research, 2026