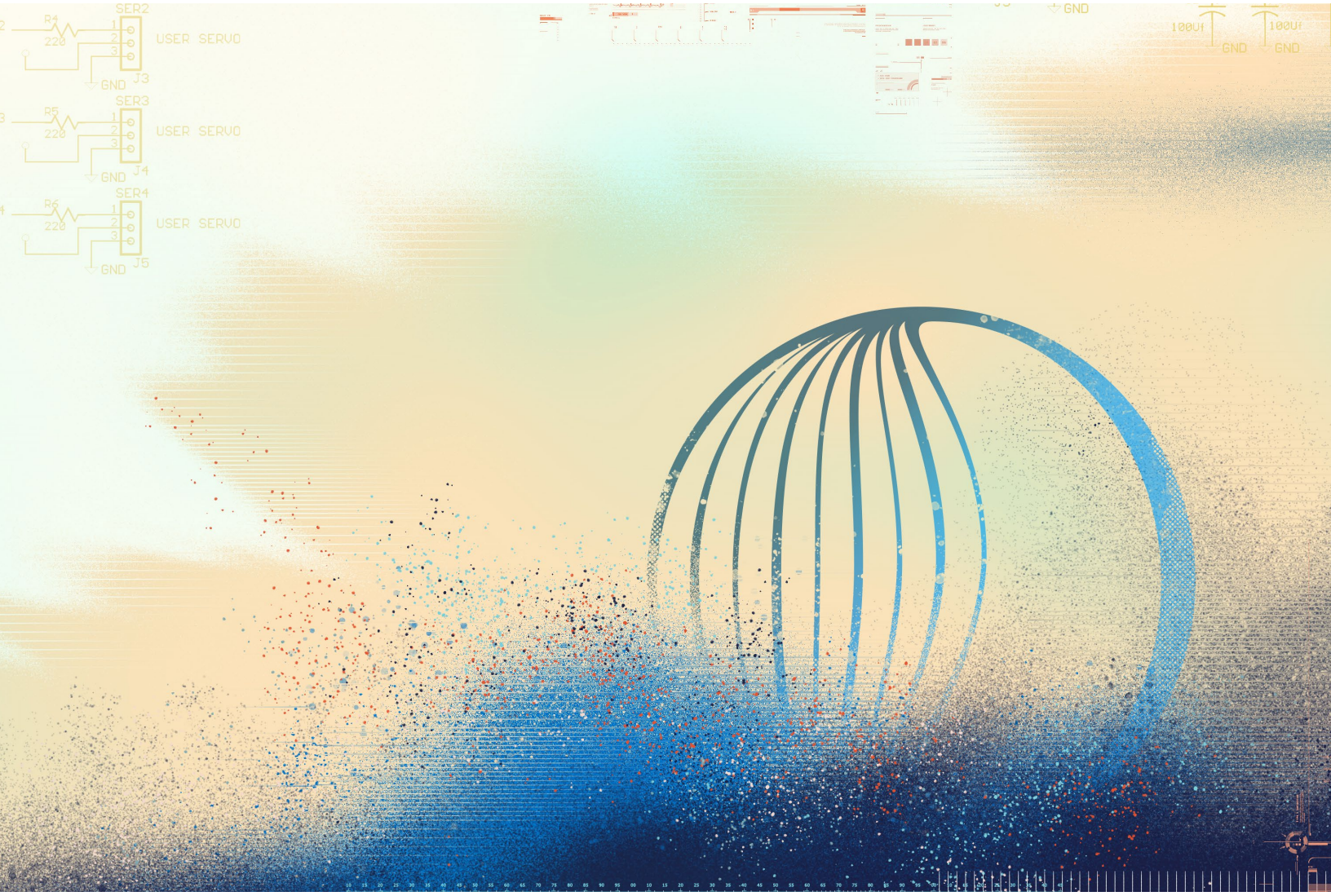
signal65



Network Topology Analysis: Scaling Considerations for Training and Inference

AUTHOR

Brian Martin
AI and Data Center Lead | Signal65

IN PARTNERSHIP WITH

DELL Technologies

AUGUST 2025

Executive Summary

Infrastructure Challenges of AI Training and Inference at Scale

The explosive growth of large language models (LLMs) and agentic AI has fundamentally transformed data center networking requirements, pushing traditional architectures beyond their breaking point. Modern AI training workloads, particularly those leveraging the latest NVIDIA GPUs with ever increasing high bandwidth memory, create an insatiable demand for network bandwidth that makes the fabric architecture a key determinant of both system performance and economic viability. The shift from compute-bound to communication-bound workloads has elevated network design to a critical enabler of AI infrastructure success.

Training workloads are characterized by massive, synchronized collective communication patterns, particularly all-reduce and all-gather operations, where hundreds or thousands of GPUs exchange gradient updates and model parameters, creating predictable but enormous bandwidth demands that can overwhelm conventional network fabrics. Inference workloads prioritize low latency and high concurrent throughput for independent request processing, where network delay directly impact user experience and service quality.

The rise of Mixture-of-Experts (MoE) models, especially in conjunction with Agentic AI systems, has additionally altered the network equation, delivering increased token performance while simultaneously reducing collective network communication traffic to levels comparable to much smaller models. These evolutions continue to impact network topology design for both workload types.

AI Workloads

The evolution of AI network topologies must consider the different requirements of training and inference:

Training Workloads	Inferencing Workloads
Patterns: Synchronized collective operations (reduce, gather, broadcast)	**Patterns:** Independent distributed request processing
Requirements: High bandwidth	**Requirements:** Low latency, high throughput, high concurrency

Networks may need to support one or the other, or both, workload patterns.

Network Architecture Evolution

The evolution from CLOS to rail-based network topologies reflects a fundamental shift from general-purpose to AI-optimized designs. CLOS networks provide universal any-to-any connectivity through leaf-spine architectures that direct most communications through three-hop paths, regardless of AI workloads' predictable patterns where GPUs of the same local rank (0-7) across nodes communicate most frequently. RAIL topologies revolutionized this by grouping same-rank GPUs into dedicated "rails" with single-hop connectivity, dramatically improving efficiency for collective operations like all-reduce. RAIL-Optimized architectures retain a limited spine layer for dramatically larger low-hop

clusters for cross-rail flexibility while RAIL only designs eliminate the spine entirely, using internal connectivity for cross-rail forwarding, achieving substantial cost reduction with minimal performance impact, especially with MoE models that drastically reduce inter-rail communication requirements.

CLOS Leaf-Spine Networks

CLOS networks implement a hierarchical leaf-spine architecture designed to provide universal any-to-any connectivity, ensuring every GPU in the cluster has a path to communicate with every other GPU through the fabric. In this design, each server node connects to a leaf switch, which in turn connects to multiple spine switches, creating a structured mesh where the spine layer acts as the central interconnect for all inter-leaf communications. In AI environments where predictable bandwidth is essential, a "fat-tree" implementation of leaf-spine is used, meaning every leaf switch has as uplink and downlink bandwidth. While this topology guarantees full bisectional bandwidth and multiple redundant paths between any two endpoints, the path lengths are not uniform. Communications between nodes on the same leaf switch may traverse shorter paths, while inter-leaf communications must traverse the spine layer, creating the characteristic three-hop (leaf-spine-leaf) path that the document references. This universal connectivity comes at a significant cost, as every leaf switch requires high-bandwidth uplinks to multiple spine switches. For AI workloads with their predictable communication patterns, this over-provisioned approach becomes economically inefficient, as the network treats a critical same-rank GPU communication (like GPU0 to GPU0 between nodes) with the same resource-intensive spine traversal as less frequent cross-rank communications, despite the dramatically different traffic volumes and importance of these patterns.

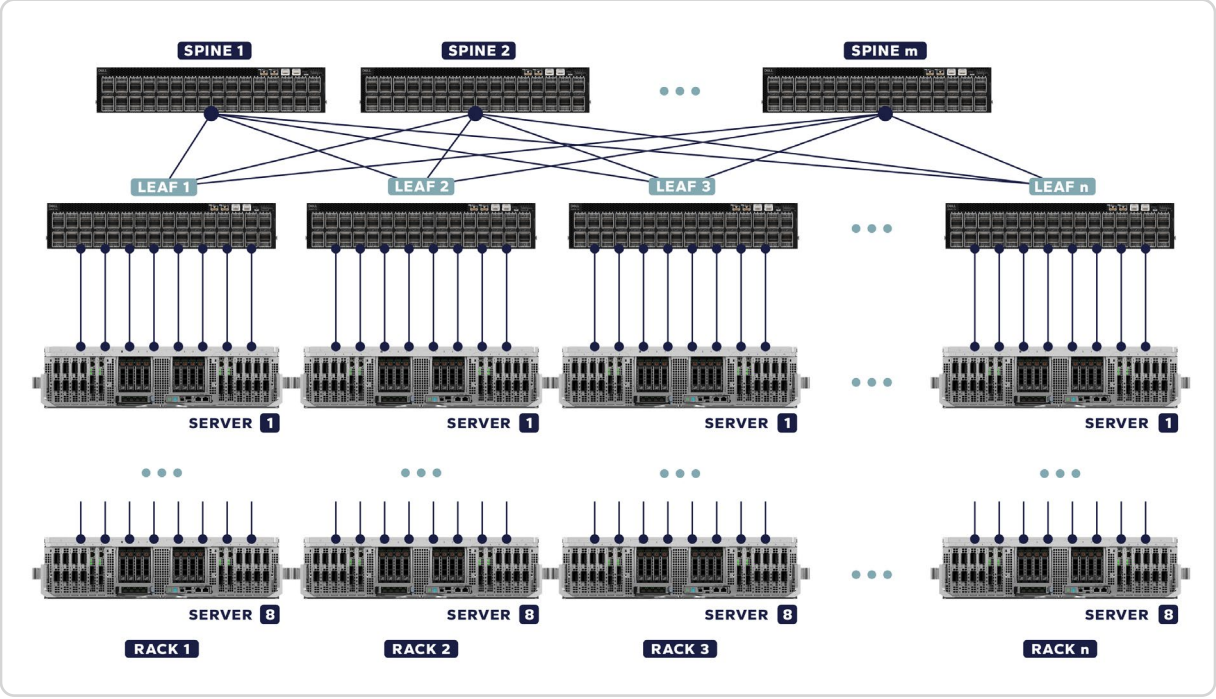


Figure 1: *A typical CLOS Network*

A CLOS network comprised of Dell Z9864F-ON switches with 64 ports at 800gb each can support one set of 8 servers (64 GPUs) per leaf switch. As an extreme scaling example, if 400gb uplinks are used, to match the 400gb connections to each GPU, then a total of 64 leaf switches can be combined into a single fabric for a total of 64 leaf switches x 8 servers per leaf switch = 512 servers or 4096 GPUs in a single, two layer fabric.

RAIL-Optimized Networks

RAIL network topology emerged from the high-performance computing (HPC) community's recognition that AI workloads exhibit highly predictable, rank-based communication patterns. The rail concept organizes GPUs by their local rank within nodes, creating logical groupings or "rails" where all GPU1s across the cluster belong to RAIL 1, all GPU2s to RAIL 2, etc. This enables single-hop communication for the dominant same-rank traffic patterns that characterize collective operations like all-reduce. RAIL-Optimized designs maintain a fewer number of spine switches to handle cross-rail traffic while enabling single-hop intra-rail communications for same-rank patterns, effectively providing the best of both worlds. The advantages include near-optimal performance for collective operations, and massive scalability to tens of thousands of GPUs for both training and inference workloads. Research has demonstrated near-linear scaling efficiency for transformer models in RAIL-Optimized configurations, making them suitable for frontier model development where maximum performance and architectural flexibility are paramount. While this architecture delivers superior performance compared to pure rail designs for workloads with significant cross-rail communication, the cost premium may become increasingly difficult to justify as MoE models and other sparse architectures reduce the actual need for spine connectivity unless extreme scalability is required.

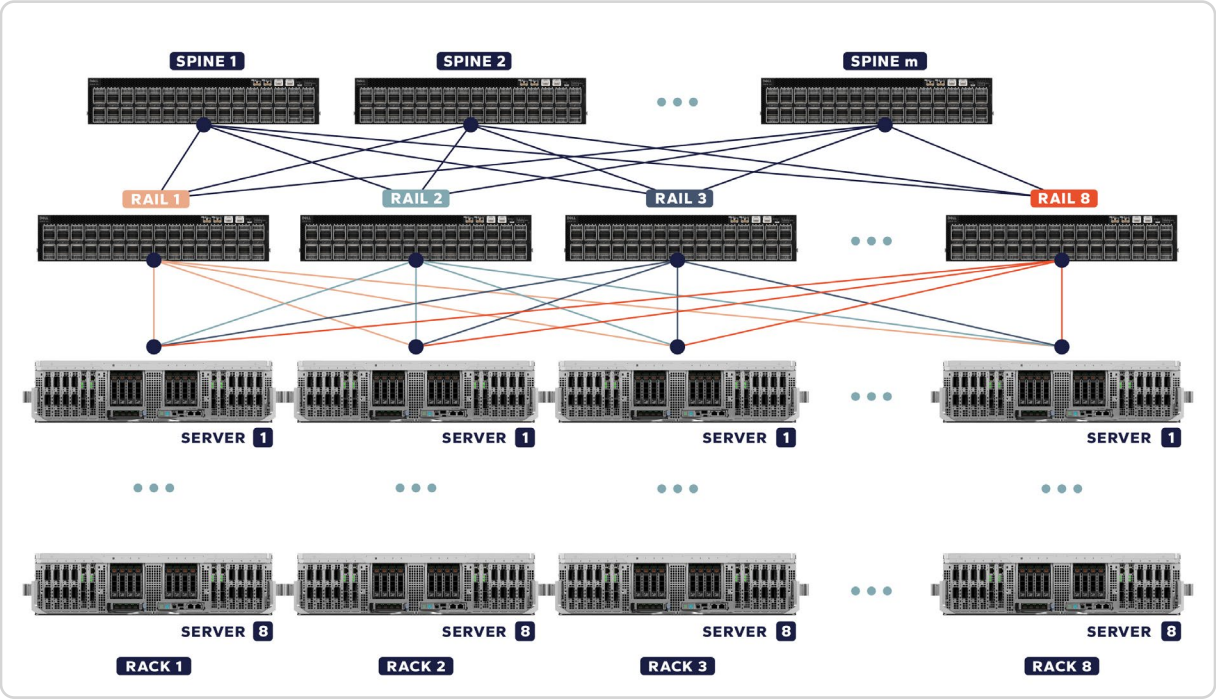


Figure 2: *A typical RAIL-Optimized Network*

A RAIL-Optimized network comprised of Dell Z9864F-ON switches with 64 ports at 800gb each can support 64 GPUs at 400gb each. As each GPU in a server connects to a different RAIL, this equates to 64 servers per switch. At scale, this represents 8 RAIL switches at 64 servers per leaf switch = 512 servers or 4096 GPUs. Connecting each of these RAILs to spine switches with 400gb uplinks allows for a total of 16 groups per RAIL-Optimized network fabric - 128 ports at 400gb per spine translates to 128 total leaf switches; at 8 RAIL switches per group that yields 128/8=16 groups per fabric. The fully scaled network fabric comprises 192 Z9864F-ON switches (8 RAILs x 16 groups = 128 RAIL switches, plus 64 spine switches). At this scale, the RAIL-optimized network fabric supports 16 groups x 512 servers per group = 8192 servers or 65536 GPUs. This represents 8x the servers of a RAIL network and 16x the servers of the CLOS network.

RAIL Only Networks

More recently, the concept of a "RAIL only" network was introduced as a more streamlined and cost-effective alternative, specifically tailored for training large language models (LLMs). Researchers observed that the communication patterns in LLM training are often sparse, meaning not all GPUs need to communicate with every other GPU at all times. The "RAIL only" architecture capitalizes on this by eliminating the spine layer of switches found in the fully RAIL-optimized design. It maintains the dedicated rails for same-index GPU communication but forgoes the comprehensive any-to-any connectivity at the spine level. This approach significantly reduces network cost, complexity, and power consumption while still providing the high-performance communication necessary for many LLM training workloads. The growing trends of MoE models and Agentic AI workflows with smaller, more focused models, have reduced the demand for cross RAIL traffic, allowing for additional streamlining of AI cluster networks.

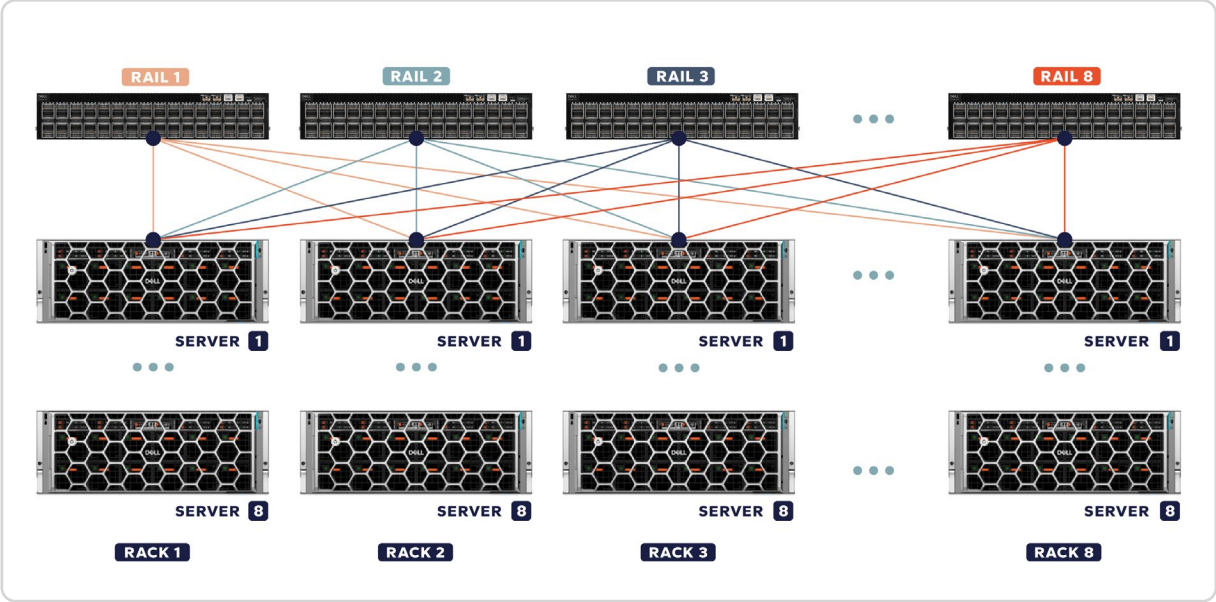


Figure 3: *A typical RAIL Network*

A RAIL network comprised of Dell Z9864F-ON switches with 64 ports at 800gb each can support 128 GPUs at 400gb each. As each GPU in a server connects to a different RAIL, this equates to 128 servers per switch. At full scale, 8 RAIL switches x 128 servers per leaf switch = 1024 servers or 8192 GPUs in a single fabric. This represents twice as many servers (1024 vs 512) in the fabric with a fraction of the total switches (8 vs 128) compares to a CLOS network.

CLOS Leaf-Spine Network	RAIL Network	RAIL-Optimized Network
Connectivity: Full	**Connectivity:** RAIL	**Connectivity:** Full
Cost/Scale: High/Low	**Cost/Scale:** Low/High	**Cost/Scale:** High/Very high
Usage: Frequent all-to-all or unknown traffic patterns	**Usage:** Aligned all-reduce patterns or MoE inference	**Usage:** Very large training and diverse model inference

Benchmarking

LLM Training Performance with Standard Models

For the training of standard, dense transformer-based LLMs like the GPT family, the dominant distributed training strategy is data parallelism. In this paradigm, the primary communication bottleneck is the all-reduce collective operation, which is used to aggregate and distribute parameter gradients across all workers after each backward pass. This operation consists almost entirely of traffic between GPUs of the same rank, making it an intra-rail communication pattern.

Because both the RAIL-Optimized and RAIL-Only topologies are explicitly designed to optimize this path with a single-hop connection at the rail switch, their performance for these workloads is virtually identical. Analytical models and empirical studies confirm that for standard LLM training, the RAIL-Only network achieves the same training performance and throughput as the RAIL-Optimized network. Both rail-based designs will outperform a generic CLOS topology, which forces this critical traffic to take a slower, three-hop path through the spine.

The introduction of the H200 GPU further reinforces this conclusion. Its larger 141 GB memory allows for the use of much larger batch sizes. For example, when training a model like Llama 2 70B, the batch size can be increased from 8 on an H100 to 32 on an H200, which can improve throughput by up to 4x. A larger batch size increases the amount of computation performed per training step relative to the amount of communication. This higher computation-to-communication ratio makes the overall training job even more resilient to minor variations in network latency, further strengthening the case that the low-cost RAIL-Only design is often sufficient, challenging a long-held industry definition of an "optimal" network as one providing full, non-blocking, any-to-any connectivity.

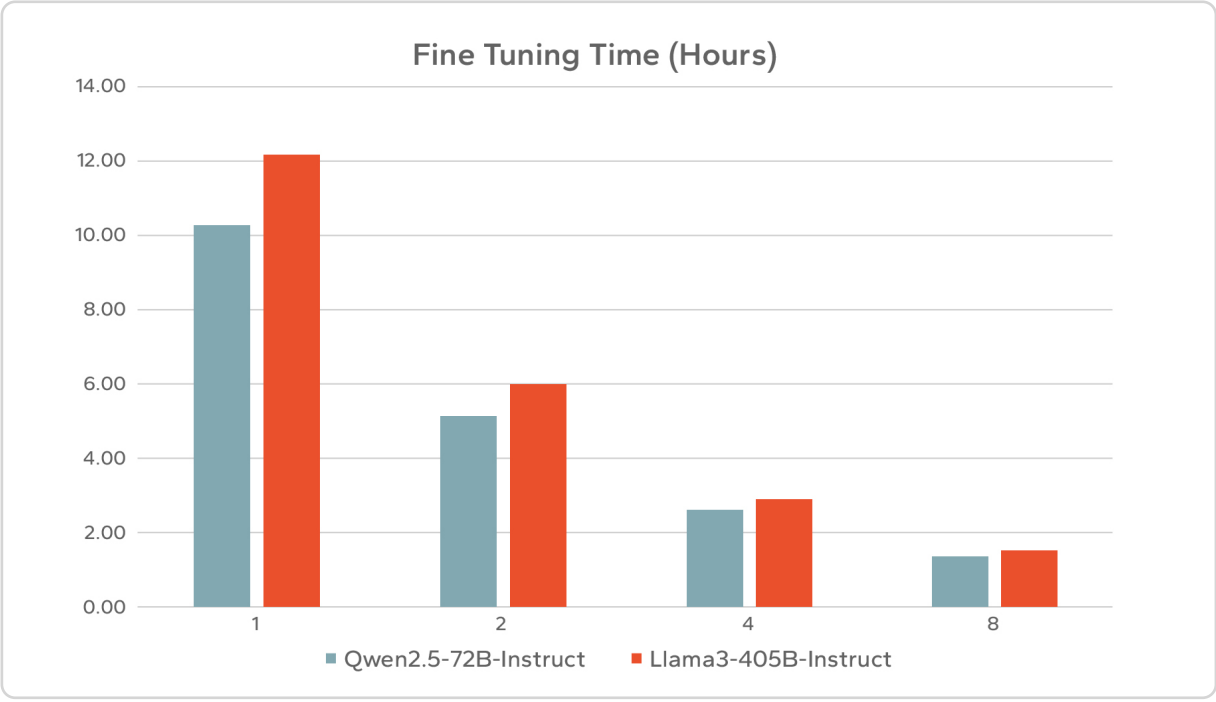


Figure 4: *Fine Tuning Time*

LLM Training Performance with Mixture of Experts

Mixture-of-Experts (MoE) models represent a different class of LLM architecture that introduces a more complex communication pattern. During the forward pass of an MoE model, each input token is dynamically routed to a small subset of "expert" sub-networks (which are typically feed-forward layers) distributed across the GPUs. This routing requires an all-to-all collective communication operation to exchange token embeddings before the expert computation and to gather the results afterward. Unlike all-reduce, an all-to-all operation is inherently cross-rail, as every GPU must communicate with every other GPU in the collective, not just its same-rank peers. This is the stress test scenario where the RAIL-Only topology's lack of a spine layer is most acutely felt. The heavy all-to-all traffic must be handled by the slower, multi-hop forwarding path that traverses the compute nodes' internal fabric.

This architectural difference results in a small but measurable performance trade-off in training. Research quantifies the throughput overhead for MoE models running on a RAIL-Only network to be in the range of 5-10% compared to a RAIL-Optimized network, which can handle this traffic efficiently via its spine. The RAIL-Only topology is therefore not a "free lunch." However, this performance degradation should not be viewed as a technical showstopper but rather as an economic decision variable.

LLM Inferencing

In the rapidly evolving landscape of large language models, two prominent architectural approaches stand out: the dense transformer models, exemplified by Meta's Llama 3 family, and the sparsely activated Mixture-of-Experts (MoE) models. While both aim to achieve state-of-the-art performance, they do so through fundamentally different strategies regarding how they scale and utilize their parameters. The core distinction lies in how they process information. Llama 3 models are "dense," meaning that for any given input, all of the model's parameters are activated to process the information. In contrast, MoE models are "sparse," utilizing a routing mechanism to select a small subset of specialized "experts" to handle a specific input. This results in only a fraction of the model's total parameters being used at any one time. Below is a comparison of the models by size, training, and requirements.

Llama Models	**Llama 2**			**Llama 3**			**Llama 4**	
	7b	**13b**	**70b**	**8b**	**70b**	**405b**	**17B-16E**	**17B-128E**
Size (GB)	13.5	26	138	15	131	1012	271	989
GPU Hours Trained(M)	.184	.369	1.72	1.3	6.4	30.84	5	2.4
Min GPUs (H100)	1	1	4	1	4	8	8	8

Figure 5: *Llama Models*

While Llama3-405b and Llama4-Maverick are approximately the same size, the dramatic difference in performance and system demand is apparent in multi-node testing:

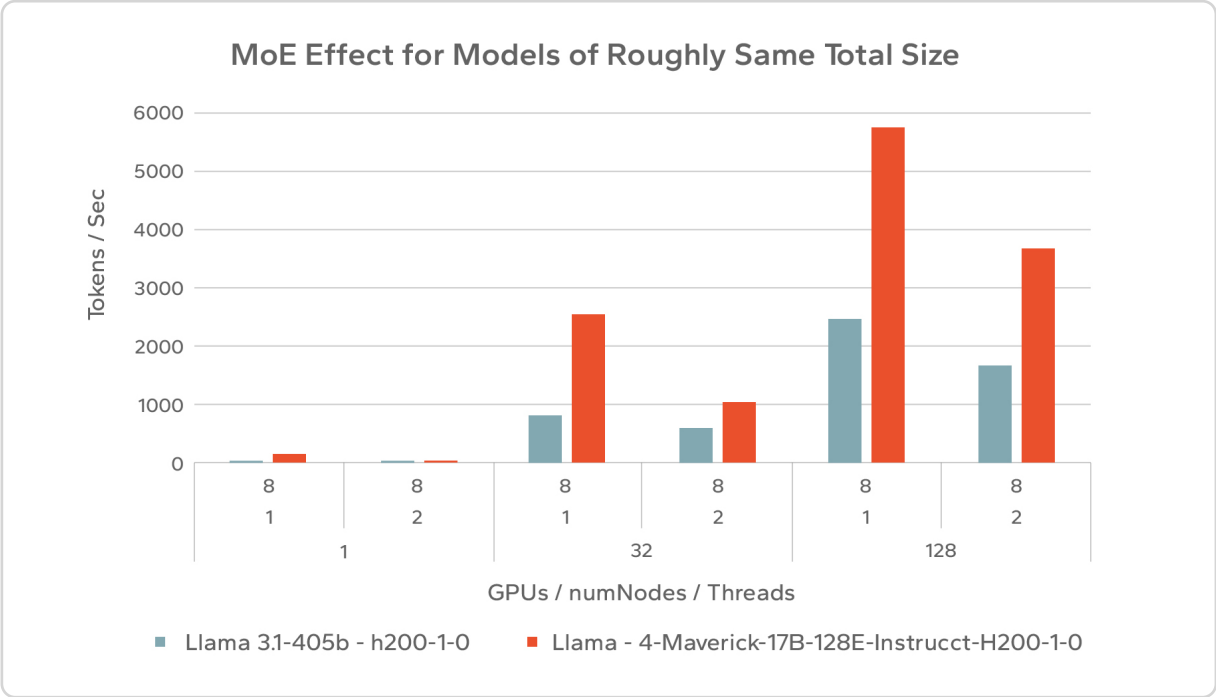


Figure 6: *MoE Effect for Models of Roughly Same Total Size*

Collective Traffic

Critically telling for this is the backend network traffic differences between models:

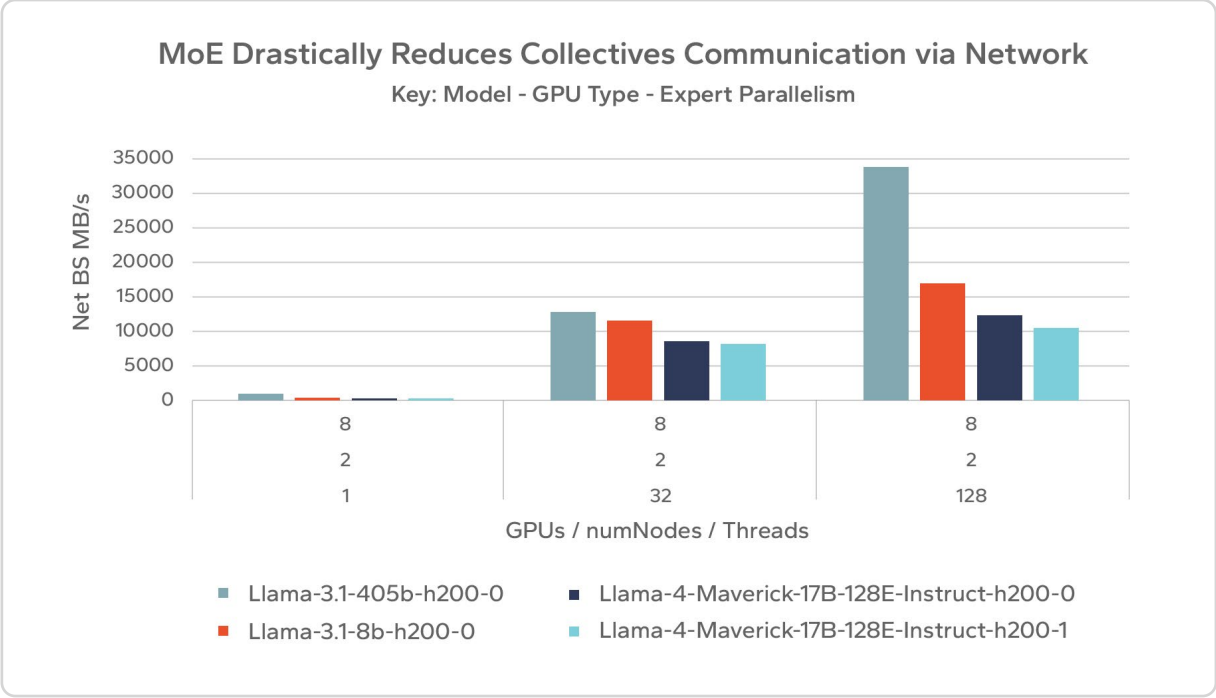


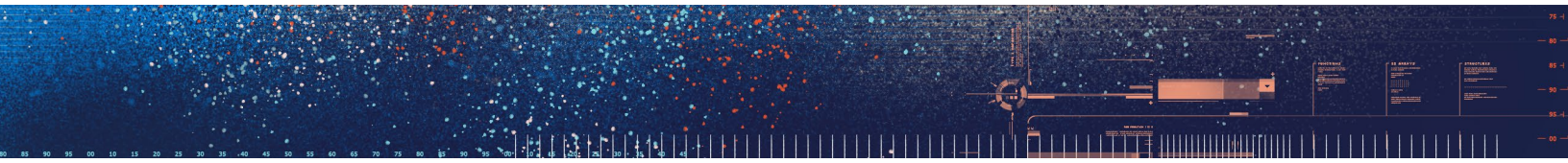
Figure 7: *MoE Drastically Reduces Collectives Communication via Network*

Implementation Considerations

Lossless AI Fabrics with RoCEv2

The physical topology defines the potential pathways for data, but it is the transport protocol and its supporting features that transform this physical infrastructure into the high-performance, reliable fabric required for distributed AI training. The protocol of choice for high-performance Ethernet-based AI clusters is RDMA over Converged Ethernet version 2 (RoCEv2), supported by advanced congestion control mechanisms.

Effective congestion control is not just a switch feature or a NIC feature; it is a tightly coupled system that requires seamless, low-latency cooperation between the two. Vertically integrated solutions, where the switch ASIC and NIC are co-designed, offer a distinct advantage. Broadcom's Thor 2 NICs and Tomahawk switch ASICs are engineered as such a system for AI/ML workloads. The Broadcom Thor 2 NIC features hardware-based congestion control. This means the logic to process incoming CNPs and adjust the sending rate is implemented directly in the NIC's silicon, not in a slower software driver or firmware layer. This hardware offload allows for microsecond-level reactions to congestion signals, enabling the fabric to stabilize much more quickly than systems relying on software-based control loops.



Total Cost of Ownership Considerations

While raw performance is a primary driver, the decision to build a multi-million-dollar AI factory hinges on a broader set of practical and economic considerations. This section provides a holistic assessment of the network topologies, moving beyond throughput benchmarks to analyze Total Cost of Ownership (TCO), fault tolerance, and the operational realities of deployment and management. TCO provides a comprehensive financial view of an infrastructure investment, encompassing not only the upfront Capital Expenditure (CapEx) for hardware but also the recurring Operational Expenditure (OpEx) for power, cooling, space, and management over the system's lifecycle. For AI network fabrics, the differences in TCO between topologies are stark and strategically significant. The primary CapEx components for a network fabric are the switches, NICs, and the optical transceivers and cables used for interconnects. Critically, research highlights that optical transceivers can account for most of the total network cost, especially as link speeds and distances increase.

Strategic Recommendations and Future Outlook

Decision Framework for Network Infrastructure Investment

Choose RAIL-Only Architecture when:

- Building dedicated AI infrastructure primarily for inferencing
- MoE models represent significant portion of workload mix
- TCO optimization is critical for competitive advantage

Consider RAIL-Optimized when:

- Requiring maximum flexibility for experimental AI architectures
- Supporting diverse legacy models with unknown communication patterns
- Extreme scalability is needed for AI training

Consider CLOS Networks when:

- Building heterogeneous infrastructure with limited AI workloads
- Simplified routing and management are preferred

Conclusion

The emergence of MoE models for agentic AI and larger GPU memory capacity are driving a shift in economics and architecture. Organizations that embrace these innovations can achieve significant competitive advantages through:

- Reduction in network infrastructure costs
- Improvement in inference throughput with MoE models
- Ability to deploy more computational resources within fixed budgets

Purpose-built architectures that precisely align with the unique demands of modern machine learning workloads will define the next generation of AI infrastructure. Early adopters of rail-based architectures optimized for MoE models will establish sustainable competitive advantages in the rapidly evolving AI landscape.

Important Information About this Report

CONTRIBUTORS

Brian Martin
AI and Data Center Lead | Signal65

PUBLISHER

Ryan Shrout
President and GM | Signal65

INQUIRIES

Contact us if you would like to discuss this report and Signal65 will respond promptly.

CITATIONS

This paper can be cited by accredited press and analysts, but must be cited in-context, displaying author's name, author's title, and "Signal65." Non-press and non-analysts must receive prior written permission by Signal65 for any citations.

LICENSING

DISCLOSURES

Signal65 provides research, analysis, advising, and consulting to many high-tech companies, including those mentioned in this paper. No employees at the firm hold any equity positions with any companies cited in this document.

IN PARTNERSHIP WITH

DELL Technologies

ABOUT SIGNAL65

Signal65 is an independent research, analysis, and advisory firm, focused on digital innovation and market-disrupting technologies and trends. Every day our analysts, researchers, and advisors help business leaders from around the world anticipate tectonic shifts in their industries and leverage disruptive innovation to either gain or maintain a competitive advantage in their markets.

signal65

signal65

CONTACT INFORMATION
Signal65 I signal65.com