



Fine-Tuning Small Language Models for the Edge

Building Specialized Intelligence for Scale

AUTHOR

Brian Martin

AI Data Center Performance | Signal65

IN PARTNERSHIP WITH

DELLTechnologies

OCTOBER 2026

Executive Summary

AI at the edge must deliver useful answers where work happens. A technician servicing remote equipment, an operator responding to a plant alarm, or a pilot receiving an onboard advisory needs relevant information within the constraints of that environment. Models must perform reliably despite intermittent connectivity and limited power, cooling, and memory.

Specialized small language models offer a practical approach. Fine-tuning adapts an existing model to domain terminology and task requirements. Authoritative local retrieval supplies current evidence, while application controls limit the model's authority. The opportunity is to develop and validate specialization centrally, then reuse a compact model across many edge deployments.

550 Aero's experiments illustrate the potential and the need for validation. Fine-tuning improved two 14B models by 5.2 and 6.0 percentage points on a 906-question aviation maintenance assessment in a four-bit deployment format. Other experiments regressed, and strong domain scores did not always translate into reliable application behavior. Model selection must consider quality, resource use, and the complete edge workflow together.

The infrastructure results show how organizations can expand this development capacity. Independent-job throughput reached 3.68 jobs per hour across eight RTX PRO 4500 Blackwell Server Edition GPUs in two Dell PowerEdge R770 servers, and 20.12 jobs per hour across 16 RTX PRO 6000 Blackwell Server Edition GPUs in two PowerEdge XE7745 servers. These represent 7.67x and 15.0x their respective single-GPU baselines. The workloads differ, so the absolute job rates are not a direct platform comparison.

Broadcom BCM57608 adapters configured as 1x400GbE, link the servers, connected through Dell PowerSwitch Z9864F-ON switches with Enterprise SONiC Distribution by Dell Technologies. The network carries shared data and coordinates distributed training across servers. Adding a second server increased distributed training throughput by 1.53x on R770 and 1.58x on XE7745 in the reported tests.

For business and technology leaders, the objective is to select the smallest model that reliably meets the operating requirement and build the infrastructure that keeps it qualified. Return depends on productive development capacity, efficient local operation, and dependable completion of useful work.

Key Highlights



Domain specialization: 5.2–6.0 percentage point gains for two compact models



Productive scaling: 96% (R770, 8 GPUs) and 94% (XE7745, 16 GPUs) concurrency efficiency



Cluster scaling: distributed training delivered **1.53x throughput on R770 and 1.58x on XE7745**

Why the Edge Changes the Model Decision

In a centralized service, infrastructure can absorb some of a larger model's cost. At the edge, every additional requirement competes with a fixed device budget. Model weights share memory with the operating system, retrieval index, conversation state, speech services, and other applications. Longer answers consume time and energy. A network dependency can turn a service interruption into an unavailable capability.

Many valuable edge tasks are also narrower than general-purpose assistant tasks. An industrial service assistant may need to explain an approved procedure for one product family. An energy operator may need a concise explanation of an alarm with the relevant local operating limits. A field-service application may need to extract observations into a maintenance record.

Narrowing the task creates room for specialization. The model needs the right vocabulary, an understanding of the expected interaction, and the ability to use supplied evidence. It does not necessarily need to solve unrelated problems or generate extensive open-ended reasoning on every request.

The resulting efficiency matters across an entire fleet. Lower memory requirements permit smaller deployment devices, potentially leaving capacity for additional services. Faster useful responses improve workflow completion. Local execution maintains availability during network loss and reduces the need to transmit operational context.

"Small" is therefore a deployment-relative term. A model suitable for a site server may be too large for an embedded device. A mixture-of-experts model may activate relatively few parameters per token while still requiring storage for a much larger total parameter set. The selection criterion is the application's measured resource envelope and task performance.



Specialization Combines Fine-Tuning, Retrieval, and Controls

Fine-tuning adjusts a pretrained model using examples of the work it should perform. For edge applications, those examples can teach domain language, concise explanations, structured responses, appropriate abstention, and the use of equipment-specific context. The goal is a model that needs less effort to perform the intended task reliably.

Retrieval supplies current evidence: maintenance instructions, service bulletins, asset configurations, and operating procedures change over time. Those facts should remain accessible through a versioned knowledge source rather than depending entirely on what the model absorbed during training. Fine-tuning and retrieval serve complementary functions: learned behavior helps the model use the evidence; retrieval makes the relevant evidence available.

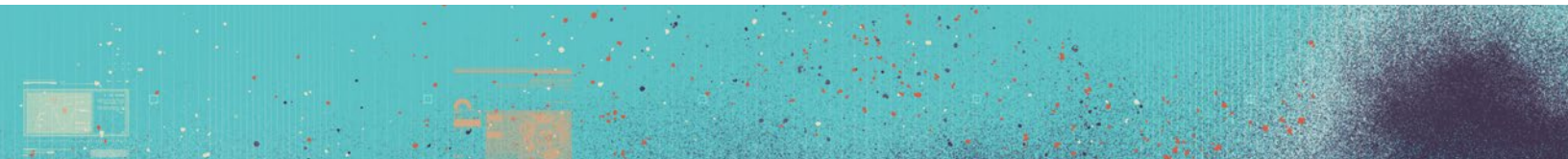
Application controls limit the model's authority. Software can restrict allowed actions, validate structured outputs, and provide a fallback when an answer is missing or invalid. A language model that explains an alarm need not also determine the alarm's severity. This separation makes the task more bounded and gives developers concrete behaviors to evaluate. It does not eliminate system errors or establish a safety approval.

An Efficient Path to Adaptation

Low-Rank Adaptation, or LoRA, trains a relatively small set of additional parameters while leaving the base model's weights frozen. QLoRA combines that approach with a quantized base model to reduce training memory requirements. These techniques can make domain adaptation practical on fewer GPUs than full-parameter training would require.

Memory planning still matters. Four-bit storage for 14 billion weights is approximately 7 GB before metadata and runtime overhead; that arithmetic is not a training memory estimate. Activations, adapter gradients, optimizer state, sequence length, and implementation choices add to the footprint. A candidate must complete a representative training run before it can be assigned to a single-GPU production queue.

The training format and deployment format also need separate evaluation. Four-bit NF4 used in QLoRA is not the same format as Q4_K_M used in a GGUF deployment artifact. Successful training does not establish the accuracy or speed of the converted model.



What the Aviation Case Study Establishes

550 Aero’s onboard advisory pipeline uses anomaly detection and a rules engine to establish an event and response. A local language model explains the supplied result and answers follow-up questions using retrieved maintenance information and aircraft context. 550 Aero reports operation on an NVIDIA Jetson AGX Orin using recorded flight data.

The development corpus contains 133 documents, approximately 6,600 extracted paragraphs, and 5,907 generated training question–answer pairs. The assessment contains 906 multiple-choice questions spanning general maintenance, Inspection Authorization, fuel systems, and ignition systems. These details make the work more informative than an isolated demonstration: they connect domain data to measured model behavior.

Domain adaptation can improve compact models

Model and evaluation format	Base accuracy	Fine-tuned accuracy	Change
Qwen3 14B, Q4_K_M	71.5%	76.7%	+5.2 percentage points
Phi-4 14B, Q4_K_M	71.4%	77.4%	+6.0 percentage points

Pre-sweep test accuracy compared with post training results

These results support a targeted proposition: a compact pretrained model can gain useful domain capability from a focused dataset. They do not show that fine-tuning always helps or that these models are sufficient for every aviation task. The experiments also include regressions, including an experiment using completion-style training rather than question–answer training. Differences in recipe, architecture, and execution prevent a clean causal ranking across the historical runs.

Knowledge and Application Behavior Need Separate Tests

The advisory evaluation checks structured output, permitted values, and consistency with the application's supplied context. Some models with strong domain scores failed these checks. The reported Qwen3 fine-tune passed 27 of 30 prompt cases, while several other candidates passed 30 of 30.

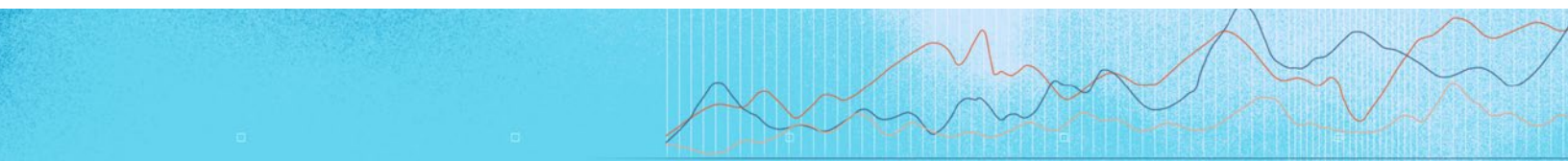
For an enterprise buyer, this distinction is consequential. A correct answer on a domain exam does not establish reliable behavior in a production workflow. A release must meet both requirements, and a 30-case test is an initial screening tool, not a reliability guarantee.

Compression and Size Can Impact Model Selection

Gemma 4 26B-A4B mixture-of-experts fine-tune scored 79.6% at Q4_K_M and 71.3% at Q3_K_S. That 8.3-point decline illustrates how an apparent model advantage can disappear when the model is compressed to fit a smaller device.

Decode rates may also differ; testing on an AGX Orin 64GB showed approximately 14 tokens per second for Qwen3 14B and 36 tokens per second for the Nemotron 3 Nano 30B-A3B MoE model. Faster candidates still need to meet the same domain and application requirements and success criteria. Mixture-of-experts models are especially appealing for edge AI as they can run faster on lower-power hardware.

The practical conclusion is to test multiple model types across several quantization levels and select the deployed artifact that best serves the complete application. Full-precision scores, server benchmarks, and parameter counts are useful inputs, but the final decision belongs on the target device.



Dell and Broadcom Infrastructure for Model Development

The central infrastructure requirement is a repeatable model-development service. Teams need to compare candidates, prepare domain variants, test revisions, and refresh deployed models as requirements change. This creates two distinct reasons to add GPUs: running more independent jobs and allocating more resources to a single job.

The R770 with RTX PRO 4500 path emphasized incremental capacity for jobs that fit on 32 GB GPUs. The XE7745 path provides 96 GB per RTX PRO 6000 GPU and greater GPU density per server, allowing more memory headroom per job and more concurrent jobs in one system.

Test Methodology

The training configuration used checkpoints with 4-bit quantized weights and BF16 compute. Maximum sequence length was 2,048 tokens with sequence packing enabled. A per-device batch size of four and four gradient-accumulation steps provided an effective batch size of 16 for a single-device job.

The software stack included unsloth[[cu130](#)], TRL, PEFT, Accelerate, bitsandbytes, and xFormers, with NVIDIA driver 595.71.05 and Linux kernel 6.8.0-137.

R770: Up to Eight GPUs Across Two Servers

Dell documents RTX PRO 4500 configurations of one, two, or four GPUs for the R770. The reported scaling tests use one and four active GPUs in a single server and eight across two servers.

Reported R770 configuration	Active GPUs	Sum of active GPU memory	Independent 14B QLoRA jobs/hour
1 R770, single-GPU allocation	1	32 GB	0.48
1 R770 × 4 RTX PRO 4500	4	128 GB	1.87
2 R770 × 4 RTX PRO 4500	8	256 GB	3.68

Results for Nemotron 3.5 30B-A3B NVFP4 QLoRA workloads across single-GPU jobs

The operating pattern is one independent fine-tuning job per GPU. Across two fully configured R770 systems, the pool provides eight concurrent job slots. Reported throughput rises approximately 7.67x from the single-GPU baseline, while individual job duration increases from 125.0 to 130.4 minutes. This expands the capacity to compare candidates, maintain domain variants, and repeat validation without requiring every job to coordinate across multiple GPUs.

XE7745: Up to Eight GPUs per Server, 16 Across Two Servers

The XE7745 supports up to eight RTX PRO 6000 Blackwell Server Edition GPUs. The reported one-, two-, four-, and eight-GPU tests refer to active GPUs assigned to workloads.

Reported XE7745 configuration	Active GPUs	Sum of active GPU memory	Independent ~30B MoE FP8 adapter-tuning jobs/hour
Single-GPU allocation in 1 XE7745	1	96 GB	1.34
Partial allocation in 1 XE7745 with 1 RTX PRO 6000	2	192 GB	2.65
Partial allocation in 1 XE7745 with 1 RTX PRO 6000	4	384 GB	5.24
Full allocation in 1 XE7745 with 1 RTX PRO 6000	8	768 GB	10.32
Full allocation across 2 XE7745	16	1,536 GB	20.12

Nemotron 3 Nano 30B-A3B FP8 QLoRA fine tuning. Listed memory is aggregate GPU memory.

The larger per-GPU memory budget can accommodate more or larger models, depending on the training method. It also provides room to test longer sequences or less aggressive training quantization. These are capacity opportunities; not every workload will fit or run faster.

An XE7745 can also run small independent jobs. Its role need not be restricted to a single large training run. A shared service may divide its GPUs among fine-tuning, evaluation, and more demanding reference-model workloads as long as resource allocation and isolation are validated.

Two Forms of Scaling, Two Measures of Success

Scaling approach	How it works	Measure that matters
Independent-job concurrency	Separate jobs run on separate GPUs or nodes	Qualified jobs completed per day; queue-to-release time
Distributed training	Several GPUs cooperate on one job through data parallelism or sharding	Time to the target quality; communication overhead; GPU-hours

Independent jobs do not need gradient synchronization with one another. Their shared constraints are more likely to include model loading, CPU processing, storage, checkpoint writes, or downstream evaluation. Distributed jobs add communication to the training path. Data parallelism replicates the model, while sharded approaches distribute training state; summing memory across GPUs does not make that total available to a single model.

For distributed tests, validate GPU placement, PCIe paths, peer-to-peer behavior, and the selected collective-communication stack. Across nodes, the reported tests use RoCEv2 and Broadcom BCM57608 adapters configured as 1x400GbE on both platforms. The network uses Dell PowerSwitch Z9864F-ON running Dell SONiC.

The R770 and XE7745 results characterize separate workload paths. Their independent jobs use different models and recipes, so jobs/hour does not establish a direct platform speed comparison. Procurement comparisons require matched task quality, workload demand, and cost accounting.

Broadcom Networking Across Servers

Adding GPUs increases available compute. Keeping those GPUs productive requires the rest of the system to deliver data and complete communication at the pace the workload demands. At the two-server stage on either platform, Broadcom networking becomes a core architectural component connecting development capacity across servers.

Both platforms use Broadcom BCM57608 adapters. Each adapter is configured as one 1x400GbE link, with Dell PowerSwitch Z9864F"-ON switches running Dell SONiC connecting the servers. The adapters support PCIe 5.0 and RoCEv2, which provides remote direct memory access over Ethernet for compatible software. Dell identifies Broadcom Tomahawk 5 as the Z9864F-ON switch silicon.

The network serves two workload patterns:

Workload	Broadcom network role	Operational value to measure
Independent fine-tuning and evaluation jobs	Move datasets and base models from shared services; write checkpoints and publish approved artifacts	Shorter job startup and staging times; fewer shared-data bottlenecks as concurrency increases
Distributed fine-tuning across servers	Carry gradient synchronization or transfers of sharded training state through the configured communication stack	Less exposed communication time; better GPU utilization and time to target quality

For independent jobs, locally cached data can substantially reduce network traffic during training. For distributed jobs, communication increases within and across training steps, making bandwidth, congestion, and tail latency directly relevant to completion time. Testing must therefore measure the traffic generated by the selected training recipes and model sizes.

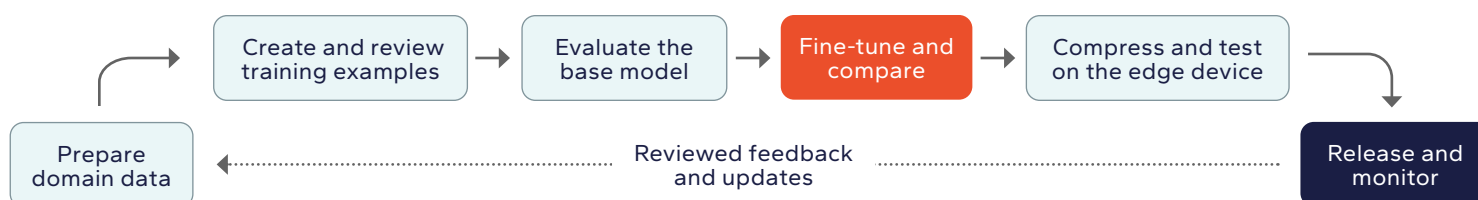
Size the network alongside the GPUs. Select adapter counts and link allocation for each server after checking PCIe slot availability, GPU-to-NIC locality, shared-storage demand, and the distributed workload’s communication volume. It was also important to validate switch uplink capacity, oversubscription, congestion-control, and traffic-priority settings under simultaneous training and checkpoint traffic.

The impact is on capital productivity. GPUs waiting on data or synchronization consume time and resources without advancing the release. Networking is a crucial part of the investment, enabling clusters to deliver useful capacity.



A Repeatable Path from Domain Data to Edge Release

An effective development service starts with the deployment contract. Define the task, authoritative sources, permitted outputs, acceptable failure behavior, latency target, and device resource ceiling before choosing the model. That turns evaluation into an acceptance decision rather than a search for the highest generic score.



The workflow then follows six stages:

- 1. Prepare authoritative data.** Version source documents and remove duplicate or conflicting content. Keep asset-specific applicability and revision dates attached to retrieved material.
- 2. Create and review training examples.** Cover representative requests, concise responses, missing information, and the boundaries of the model’s role. Review generated examples for unsupported claims and incorrect procedures.

3. **Measure the untuned baseline.** Test the base model with the intended prompt and retrieval pipeline. Fine-tune only where it offers a measurable benefit over that baseline.
4. **Fine-tune and compare.** Use a common starting recipe across candidates and record required exceptions. Preserve a held-out test set; tune recipes on a separate development set.
5. **Convert, compress, and qualify.** Reevaluate the exact deployment artifact for task accuracy and application behavior, then measure it on the target hardware with other services active.
6. **Release and maintain.** Version the model, retrieval content, prompt, and runtime together. Use controlled deployment, rollback, and regression evaluation when any component changes.

The aviation-based experiments use a deterministic paragraph-level training/evaluation split. Further validation should check near-duplicate content across paragraphs and documents, and include a document- or revision-level holdout. A paragraph split alone cannot rule out exposure to substantially similar source material.

Updates should follow the nature of the change. A revised service bulletin may require a retrieval refresh and regression tests. A new interaction pattern may justify adapter training. Fine-tuning every time any document changes adds work without necessarily adding value.

Qualify the Complete Edge Application

Acceptance dimension	Required evidence
Task quality	Held-out domain accuracy, grounded answers, and appropriate abstention
Application behavior	Valid outputs, permitted values, and no unauthorized changes to supplied decisions
User experience	Time to first useful response and complete response, including retrieval and speech where applicable
Resource fit	Peak application memory, realistic context use, and steady-state thermal behavior
Operating resilience	Behavior during lost connectivity, unavailable evidence, restart, and rollback

A model file that fits in device memory may still fail during a long interaction. Runtime memory includes the key-value cache used to retain token state as well as the rest of the application. Similarly, a high decode rate may coexist with a slow initial response. Report median and tail latency for the complete workflow, plus sustained behavior under the intended power mode.

Measuring Efficiency in Qualified Work

The economic advantage of specialization spans development and operation. Development consumes infrastructure time, data preparation, engineering, and evaluation effort. Deployment repeats across devices and interactions, potentially multiplying small per-interaction savings. The balance depends on workload volume, update frequency, and how much specialization you can share.

For budgeting, define:

Cost per accepted task = attributable lifecycle cost over a defined period ÷ tasks completed within the agreed quality and latency requirements during that period.

Lifecycle cost includes allocated server and edge-device costs, Broadcom adapters and Dell switching, optics, energy, software, data curation, validation, deployment management, and support. Costs incurred by failed responses, retries, and unsuccessful model candidates remain in the numerator. Define an accepted task independently of the candidate model so that a system cannot appear efficient merely by declining difficult requests.

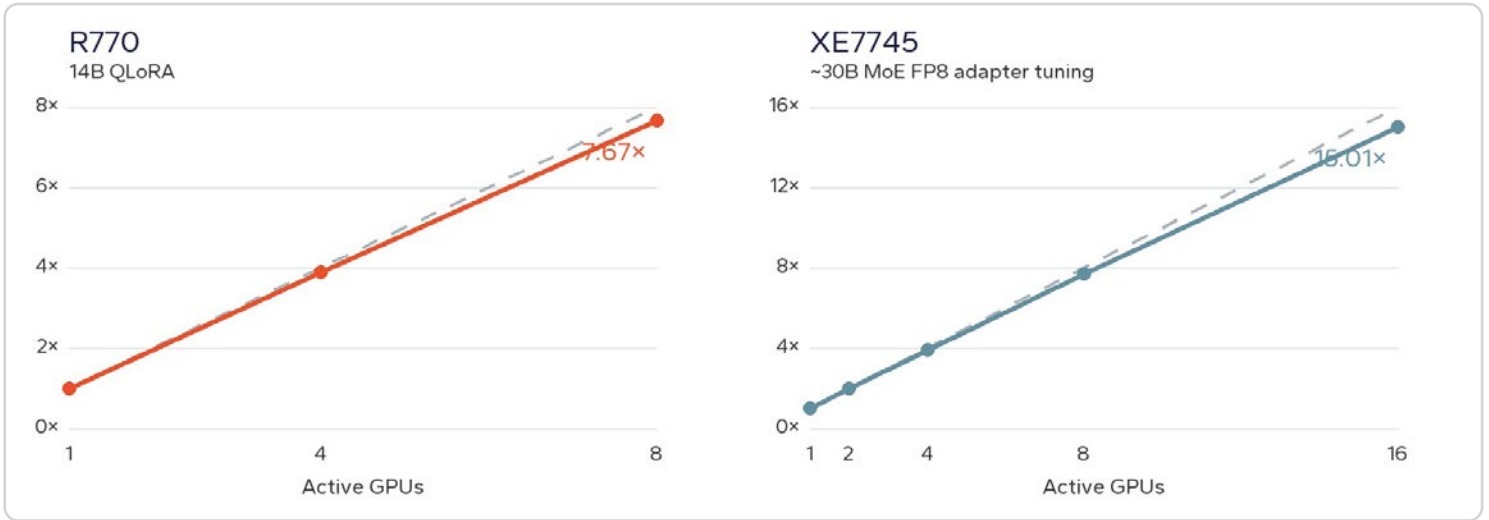
Token metrics remain useful operational measures, but a concise and correct response can be more valuable than a larger volume of generated text. Response time is also often critical in edge AI. Assess the economics at the level of completed work.

Three operating measures make the business case reviewable:

- **Qualified releases per development dollar.** Count model versions that pass all release gates, including the cost of failed experiments and evaluation.
- **Energy per accepted edge task.** Measure the complete device and application during representative operation, including their share of idle consumption.
- **Time from approved change to qualified release.** Include queue time, training, conversion, testing, and deployment preparation.

Scaling Results

The scaling results cover eight independent-concurrency configurations and four distributed-training configurations. They show two ways to scale specialized model development: increasing the number of independent jobs completed and shortening a training workload through coordinated GPU execution.



Independent jobs scale with modest per-job slowdown. Dashed lines show ideal linear scaling.

R770 throughput increases from 0.48 jobs/hour on one GPU to 3.68 on eight GPUs across two servers, approximately 7.67x, with reported concurrency efficiency of 95.9%. XE7745 throughput increases from 1.34 to 20.12 jobs/hour across one to 16 GPUs, approximately 15.0x, with reported efficiency of 93.9%. Individual job duration increases by approximately 4.3% on R770 and 6.5% on XE7745 between those endpoints.

These results support the model-development service described in this paper: additional GPUs can expand the number of domain variants, candidates, and repeat runs processed concurrently with relatively modest per-job slowdown. Completed jobs are not automatically qualified releases; domain and application tests remain release gates.

Distributed Training Reduces Completion Time

Platform	Active GPUs / servers	Training throughput (tokens/second)	Training completion time (hours)	Throughput gain over measured single-server baseline
R770	4 / 1	3,920	4.12	1.00×
R770	8 / 2	6,010	2.69	1.53×
XE7745	8 / 1	18,850	0.86	1.00×
XE7745	16 / 2	29,800	0.54	1.58×

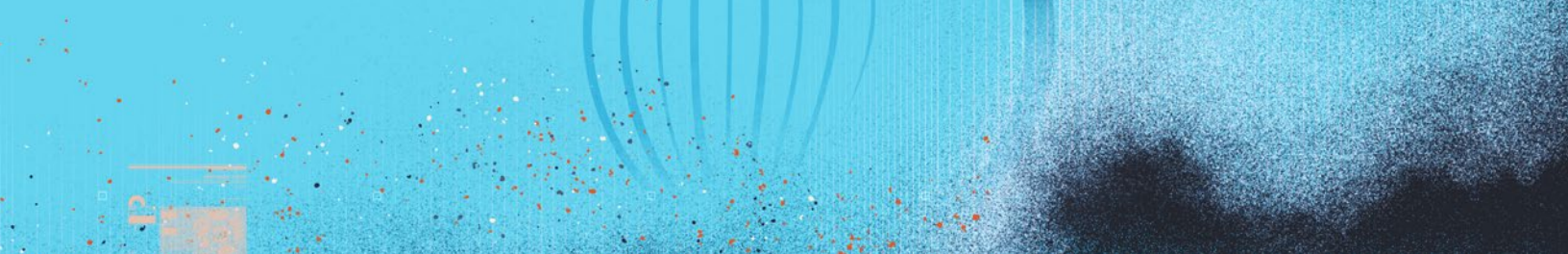
Reported ~30B MoE distributed runs. Completion times describe the training workload

Doubling GPU allocation across two servers delivers 76.7% scaling efficiency on R770 and 79.0% on XE7745, relative to their measured single-server baselines. The results show faster completion and sublinear throughput growth. The difference from ideal linear scaling can include communication, synchronization, imbalance, and other effects.

Networking Underpins the Distributed Configuration

Both server platforms use identical Broadcom BCM57608 adapters configured as 1x400GbE, with RoCEv2 carrying cross-node training traffic. The second server raises distributed throughput by 1.53x on R770 and 1.58x on XE7745.

The network is critical to making separate GPU servers function as a coordinated development resource. Adapter capability, configured link rate, GPU-to-NIC placement, and congestion behavior each affect cross-node performance.



Conclusion

Specialized small language models offer enterprises a practical path to useful AI within the power, memory, and connectivity constraints of the edge. The investment decision starts with a bounded operational task and a model that can meet its requirements. 550 Aero's results show why validation matters: focused fine-tuning improved domain accuracy for compact models, while other experiments exposed weaknesses in compression and application behavior.

Develop specialization centrally, where teams can share infrastructure, compare candidates, and repeat evaluation. Dell PowerEdge R770 and XE7745 systems provide complementary capacity for that work, scaling to eight RTX PRO 4500 GPUs or 16 RTX PRO 6000 GPUs across two servers, respectively. Broadcom BCM57608 adapters configured as 1x400GbE and Broadcom-powered Dell switching connect those resources for shared-data access and distributed training. Planning compute and networking together helps translate additional GPUs into productive development capacity.

Qualify the complete application on its target edge device before release. Domain accuracy, grounded responses, application behavior, latency, and memory use must meet requirements with the intended services running. Version the model, retrieval content, prompts, and runtime together so updates remain testable and reversible.

Measure the return through cost per accepted task and time from approved change to qualified release. Central development can spread the cost of specialization across a fleet; efficient local execution can extend that value across repeated use. Invest in the capacity to develop and maintain specialization, then deploy the smallest model that reliably meets the operating requirement.

Important Information About this Report

CONTRIBUTORS

Brian Martin

AI Data Center Performance | 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

This document, including any supporting materials, is owned by Signal65. This publication may not be reproduced, distributed, or shared in any form without the prior written permission of Signal65.

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.

ABOUT SIGNAL65

Signal65 is a leading research organization specializing in enterprise AI infrastructure optimization and deployment strategies. Our lab focuses on evaluating and optimizing AI hardware and software solutions for real-world enterprise applications, with particular expertise in large language models, retrieval-augmented generation systems, and distributed AI architectures.

For more information, visit signal65.com or contact research@signal65.com



IN PARTNERSHIP WITH



CONTACT INFORMATION

Signal65 | signal65.com