PURETENSOR
INTELLIGENCE DISTILLED

Distributed Inference on Workstation Blackwell, Part 4

Cross-node tensor parallelism over 200 GbE

Paper: PT-R-2026-017 v1.0

Author: PureTensor Ltd

Date: 4 September 2026

Status: Published

Abstract

Cross-node tensor parallelism distributes a single model's layers across GPUs on separate hosts, requiring an all-reduce collective over the network at every layer of every token's forward pass. For models too large to fit on one machine, this is the only viable geometry; the cost is that single-stream decode becomes bounded by collective latency rather than arithmetic throughput. We report measurements from 21 August to 2 September 2026 on a two-node cluster of four RTX PRO 6000 Blackwell GPUs (384 GB VRAM total) joined by a 200 GbE RoCE fabric with GPUDirect RDMA, serving two models that exceed single-node capacity: NVIDIA Nemotron 3 Ultra 550B and GLM-5.3-Flash.

Nemotron 3 Ultra 550B (NVFP4, roughly 55B active parameters) loaded at 79.74 GiB per rank and decoded at 16.3 tokens per second single-stream, scaling to 64.5 tokens per second aggregate at four concurrent streams with zero per-stream degradation. A 90-minute soak completed 912 of 912 requests (350,208 tokens) with no errors, no stalls, and no GPU faults. The first boot attempt deadlocked before any weight loaded: the serving engine's custom all-reduce prober entered a symmetric-memory rendezvous that SM120 (Blackwell's compute capability) does not support. Disabling the prober with an explicit flag was the sole fix required.

GLM-5.3-Flash, a mixture-of-experts model with multi-head latent attention and no rotary position embedding in the attention path, required nine boot attempts in one day before producing correct output. Five distinct blockers emerged: the same symmetric-memory deadlock; an SM90 MLA backend that needed gating to capability 12; an auto-derived attention block size incompatible with the kernel's alignment; NaN logits from a FlashInfer bug fixed in a 0.6.18 nightly; and an MoE backend mismatch. With all five addressed, the model decoded at 95.2 tokens per second single-stream and 175.5 aggregate at 16 concurrent, later scoring 91.4 flat on our internal frontier benchmark. A community-qualified image with SM120-specific kernels reached 100.1 tokens per second single-stream (167 to 171 with multi-token prediction) and 691.3 aggregate at 16 concurrent, 2.8 times our best alternative seat for this model.

Two structural failures recurred across engines and models: the custom all-reduce deadlock on SM120, and speculative decoding across a network, where data-dependent accepted draft lengths cause ranks to diverge in the number of collectives they issue, stalling the sampling step. These are properties of multi-node tensor parallelism on this hardware generation, not of any single engine. Our throughput numbers were taken under power caps (250 W and 325 W against 600 W parts) and across a network rather than an intra-chassis bus; they are therefore lower bounds. The software path for SM120 is young, but cross-node tensor parallelism on

workstation Blackwell over commodity Ethernet works today for models that do not fit one node.

1. Hardware and Fabric

The cluster comprises two compute nodes, each carrying two RTX PRO 6000 Blackwell GPUs with 96 GB of VRAM per card (compute capability 12.0, designated SM120 in kernel toolchains). Node A uses the Max-Q variant, power-capped by us to 250 W during these runs; Node B uses the workstation variant, capped to 325 W after earlier bus-drop incidents on one card. The four GPUs together provide 384 GB of VRAM.

The nodes are joined by a 200 GbE RoCE fabric with GPUDirect RDMA. Preflight measurement of RDMA host-memory bandwidth returned 195.87 Gb/s, 98% of line rate. Every NCCL channel logged as GDRDMA, confirming that collective traffic bypassed host memory. Weight staging between nodes with our transfer tool achieved 9.97 to 11.33 GB/s depending on the checkpoint and the destination storage ceiling; parity was verified on every copy.

Cross-node tensor parallelism at TP4 shards every layer's weights across all four GPUs. Each token's forward pass therefore requires all-reduce collectives across the network at every layer. Single-stream decode is bounded by collective latency, not by arithmetic throughput. Aggregate throughput scales with concurrency because the fabric can carry multiple streams' collectives in parallel; the penalty is paid per stream, not per cluster.

2. Nemotron 3 Ultra 550B

NVIDIA's Nemotron 3 Ultra is a 550-billion-parameter hybrid model (Mamba state-space layers, 12 attention layers, and a mixture-of-experts block) with roughly 55 billion parameters active per forward pass. The NVFP4 quantisation we used had set our single-shot benchmark record (92.7 flat) through a hosted lane earlier on 21 August. We are not aware of any public report of this model served on four RTX PRO 6000 GPUs.

The serving engine was a development build of vLLM with a multiprocessing backend across two nodes, the Marlin MoE backend, an FP8 KV cache, eager execution, a 16,384-token context window, and a maximum of eight sequences. The engine skips the model's 20.9 GiB multi-token-prediction block at load; predicted memory was 76.8 GiB per rank, actual was 79.74 GiB per rank.

Metric	Value
Cold boot to serving	3 min 13 s (2 min 59 s on second boot)
Weight load (page cache hot)	79.74 GiB per rank in 99.4 s
Engine init (profile and warm-up)	51.7 s
Single-stream decode	16.3 tok/s steady (10.95 on first request)
Four-stream aggregate	64.5 tok/s (16.1 per stream)
10-question reasoning ladder	10/10 correct
KV pool at 90% memory utilisation	0.6 GiB, 68,266 tokens

The first attempt deadlocked before any weight loaded. All four ranks stalled in a symmetric-memory rendezvous inside the engine's custom all-reduce multicast prober. SM120 has no symmetric-memory support; the cross-node rendezvous never completes. We diagnosed this with a Python stack sampler on both nodes. The fabric itself was already live and was never the problem.

The fix was to pass the flag that disables the custom all-reduce explicitly, along with a one-hour distributed timeout. The engine's claimed automatic multi-node disable does not gate the prober in this build; without the flag every rank deadlocks silently.

A 90-minute soak at four concurrent streams completed 912 of 912 requests, generating 350,208 tokens at 64.8 tokens per second aggregate. Latency was 23.5 s at p50, 24.8 s at p95, and 25.9 s at worst; no degradation tail appeared. Zero GPU errors were logged on either node. Node A's Max-Q cards ran 80 to 84 °C and throttled to approximately 2.3 GHz, with no effect on throughput because decode is latency-bound. A publicly reported hang on two-node TP over RoCE on this GPU generation, said to occur 35 to 55 minutes in, did not manifest. We keep the caveat that we soaked 90 minutes at four streams; longer horizons and higher concurrency remain unproven.

Multi-token-prediction speculative decoding was deliberately not used. Elsewhere on the Marlin path we measured a 22% throughput loss with it enabled.

3. GLM-5.3-Flash: FP8 and the NoPE Geometry

GLM-5.3-Flash is a mixture-of-experts model with multi-head latent attention (MLA). Its attention path uses no rotary position embedding: the rotary head dimension is 0, a geometry sometimes called NoPE. This is the geometry that broke most kernels.

On 26 August we loaded the FP8 checkpoint (62 shards, 145 files, 328 GB) on all four GPUs at TP4 with expert parallelism 4. Each rank held 75.74 GB in 67.8 to 68.9 s; NCCL confirmed GPUDirect RDMA in both directions; 902,592 FP8 KV tokens per rank were exposed.

No tested native path supported the NoPE sparse-MLA geometry on SM120. The vLLM FP8 MLA cache requires a 64-dimension rotary component. The TensorRT-LLM generated kernels return "unsupported architecture" on SM120. The TileLang path requested 151,552 bytes of dynamic shared memory and could not launch. The new FlashInfer SM120 sparse-MLA kernel is hard-coded to a KV low-rank of 512, a rotary head dimension of 64, and a query dimension of 576; this model has rotary dimension 0 and query dimension 512.

A temporary pure-PyTorch fallback answered at about 7 tokens per second with repetitive non-terminating reasoning and missed tool calls. We stopped and published nothing: the two attempted benchmark artefacts are zero bytes; partial fallback output was not scored. This session is not evidence about the model's quality. We filed or referenced upstream issues in the three kernel projects.

The same day we tried a workaround: a self-converted 8-bit GGUF (direct FP8-to-Q8_0 repack, 341 GB in eight splits, 33 minutes, no BF16 intermediate) on a day-old llama.cpp branch, four GPUs across two hosts via its RPC transport over the fabric. This decoded at 46.1 tokens per second single-stream, 1.5 times a hosted reference's 31.3. But tool-call parsing was absent for this architecture, so four of ten benchmark dimensions were unscored and no leaderboard entry was made.

4. GLM-5.3-Flash: NVFP4 and the Five Fixes

The NVFP4 checkpoint quantises only the MLP experts to FP4 (171.23 GB of 190.20 GB total weights); attention, shared experts, embeddings, the visual encoder, and the dense MLP stay BF16 or FP8. On one two-GPU node alone the residual per card would be about 6.7 GB, below the roughly 10.2 GB of non-weight runtime measured earlier. Cross-node TP4 was the only on-premises shape.

Nine attempts on 28 August produced five distinct blockers.

Attempts 1 and 2 hung for 30 minutes with no error. Stack traces showed local-rank-0 workers blocked in the custom all-reduce symmetric-memory rendezvous while local-rank-1 workers had left the try/except and entered a message-queue broadcast: a rank-divergent deadlock. The fix was to disable custom all-reduce.

Attempt 3 loaded weights (44.69 GiB per rank in 19 to 54 s) then died in profiling: the FP8 MLA cache requires a 64-dimension rotary component. The image's only SM120 sparse-MLA backend hard-requires DeepSeek's layout.

Attempt 4 applied a patch gating the Hopper NoPE MLA backend to compute capability 12 and used a BF16 KV cache. An indexer kernel asserted on the attention block size: the engine auto-derived 1152 (for Mamba page alignment), which is not a multiple of 256. The fix was to set block size 2304.

Attempt 5 served and captured CUDA graphs (19 piecewise plus 11 full, 14 s), allocating 24.17 GiB of KV per rank (2,063,051 tokens). Output was degenerate: 400 tokens of the word "lock" repeated.

Attempt 6 used eager mode. Same garbage at 5 tokens per second, ruling out CUDA graphs as the cause.

Attempt 7 used the Marlin MoE backend. Same garbage. Requesting prompt log-probabilities returned HTTP 400 citing out-of-range float values, confirming the logits were NaN at prefill.

Attempt 8 used an image with FlashInfer upgraded from 0.6.17 to a 0.6.18 nightly (which carries a fix for an FA2 MLA NaN on SM12x), plus NaN hooks. Output was correct ("2317 = 2310 + 23*7 = 230 + 161 = 391") with reasoning separated by the parser, 57 completion tokens. NaN hooks across 1,171 modules on four ranks reported nothing. The FlashInfer upgrade was the decisive fix.

Attempt 9 used the production shape (CUDA graphs on, hooks off). Output was correct. Engine init took 126 s (weights 16 s on one node, 40 s on the other; graph capture 14 to 15 s).

Metric	Value
Single-stream decode	95.2 and 94.0 tok/s
Time to first token	0.064 and 0.057 s
16 concurrent (256 tokens each)	4,096 tokens in 23.3 s (175.5 tok/s aggregate)

On 29 August this seat scored on our internal frontier benchmark (94 original items, ten dimensions, every verdict produced by code, no language-model judge): agentic 88.3, scorer 95.4, flat 91.4, at 78 to 82 tokens per second single-stream and 589 tokens per second aggregate at 16 concurrent, with a needle retrieval passing at 261,900 tokens.

The five fixes, in summary: disable custom all-reduce; gate the SM90 MLA backend to capability 12; set the attention block size to 2304; upgrade FlashInfer to 0.6.18; use the Marlin MoE backend.

5. A Public Recipe That Did Not Work

A community leaderboard entry claimed 1,004.9 tokens per second single-stream for GLM-5.3-Flash on two RTX PRO 6000 GPUs (SGLang, ModelOpt NVFP4, TP2, a sparse-attention backend with TileLang kernels, a 64-token draft speculative decoder, clocks locked at 3090 MHz and 450 W). The submitter's own note reads "acceptance 64/1.00 on locked attractor": the model was in a repeat loop, so every 64-token n-gram proposal was accepted. That number measures loop replay, not decode.

We made fourteen attempts on our two-GPU node. Three patches were needed just to boot: a loader patch treating unmatched layers as unquantised; an SM120 tiling patch for the TileLang

kernel, whose default asked 104 KB of shared memory against a 99 KB SM120 cap; and an environment flag. The server came up at 88.2 GB of weights per rank with a 16k context and one request slot. Every forward produced NaN. A per-layer tensor dump localised it: layers 0 to 2 (dense, BF16) were clean; the first NVFP4 expert layer's output was 100% NaN from clean gate and top-k inputs. The result was unchanged with CUDA graphs off, with the Marlin runner, with the sparse path forced, and with the 0.6.18 FlashInfer nightly. The TensorRT-LLM generated runner ships no SM120 kernels. We could not determine whether the kernel itself or the weight preparation is at fault, for want of a fitting checkpoint in the other quantisation format. Fit is marginal at TP2 for every NVFP4 checkpoint of this model (88 to 94 GB per rank of 95 GiB usable), so even fixed it would be a single-user 16k-context seat.

6. A Qualified Community Image, Cross-Node

On 2 September a community lab published a vLLM image with SM120-specific kernels (designated B12X) qualified for GLM-5.3-Flash on a single four-GPU node. Its launcher is single-node only. Five deviations from its runbook were needed to run cross-node: add Ray to the image; bypass the entrypoint chain to pass the Ray distributed executor; set the flag that disables the intra-node PCIe all-reduce (which adds the disable-custom-all-reduce flag); reduce GPU memory utilisation to 0.75 because co-tenant services stay resident on one node (at 0.78 the run aborted with 73.98 GiB free against 74.11 requested); cap the CUDA-graph capture size at 256.

Configuration was TP4, 131k context, FP8 KV cache. Startup took 67.5 s for engine init, 13 s for graph capture, and allocated 21.0 GiB KV per rank (17.5 GiB with speculative decoding).

Mode	Concurrency	Aggregate tok/s	Per-stream decode	Time to first token
No speculation	1	100.1	105.6	275 ms
No speculation	4	294.5	79.5	509 ms
No speculation	8	478.8	64.7	642 ms
No speculation	16	691.3	46.2	775 ms
MTP, 3 draft tokens	1	167.1 and 171.5	180–185	about 225 ms
MTP, 3 draft tokens	8	engine death	engine death	not applicable

Prefill throughput was 4,542 tokens per second on a 31,218-token prompt (6.87 s).

Configuration	Single-stream tok/s	Aggregate tok/s
Lab's qualified single-node TP4 (four workstation-variant GPUs, stock clocks)	163.4	735 at 8
This run, cross-node TP4, power-capped	100.1 (167–171 with MTP)	478.8 at 8; 691.3 at 16
Our SGLang FP8 cross-node TP4 (27 August)	91.3	555 at 16
Our adopted single-node EXL3 TP2 seat	102–125	246 at 16

We reach 61% of the lab's single-stream and 65% of their eight-way aggregate, with two handicaps their box does not have: our collectives cross a network (theirs use an intra-chassis PCIe all-reduce), and our cards are power-limited (325 W and 250 W against 600 W parts; observed draw 278 to 289 W on one node and 184 to 188 W on the other, so both sat at their caps).

At 16 concurrent this beats every other configuration measured on our fleet, including the EXL3 seat by 2.8 times. Single-stream does not beat that seat.

The speculative-decoding failure at concurrency (an RPC timeout in the sampling step that kills the server) is the same defect recorded on 27 August under a different engine: accepted draft length is data-dependent, so ranks diverge in the number of collectives they issue and the sampling collective stalls. It is a property of speculative decoding across a network, not of this image.

Adopting this seat costs both nodes' resident services and about 30 GB of co-tenant headroom: a fleet-wide trade, not a drop-in replacement.

7. Earlier Related Measurements

On 27 August, SGLang FP8 cross-node TP4 achieved 91.3 tokens per second single-stream and 555 at 16 concurrent, but only after removing three stale flags inherited from a failed prior run that had cost an 8.1-times self-inflicted throughput loss (11.2 to 91.3 single, 169 to 555 at 16). Speculative decoding via the model's next-token head produced an accepted length of 2.2 to 3.0 of 4, then a collective timeout on the second or third request. At FP8, 75.06 GB of weights per rank leaves no room for a production KV cache, so FP8 cannot be a resident seat on this hardware.

On 23 August, a capacity ladder of very large GGUF models via llama.cpp RPC across the same four GPUs yielded: a 554 GB 1-bit-class quant of a 2.8T MoE at 8.9 tokens per second (17% over single-node); an 861 GB 2-bit quant at 4.2 tokens per second; an 801 GB near-lossless 8-bit quant of a 755B model at 3.6 tokens per second. Two findings emerged: bytes per token beats parameter count at the memory-bandwidth frontier (the 8-bit smaller model is slower than the 2-bit larger one), and page-cache eviction is a 26-times throughput swing on an identical configuration (0.16 versus 4.2 tokens per second) when a concurrent download evicts the weights.

8. What These Numbers Will Not Carry

Sample sizes are small. The Nemotron soak ran 90 minutes at four streams; longer horizons and higher concurrency remain unproven. The GLM-5.3-Flash benchmark run is a single pass of 94 items. Throughput figures are point estimates from one or two runs, not distributions.

Our cards were power-capped (250 W and 325 W against 600 W stock), and our collectives crossed a 200 GbE fabric rather than an intra-chassis bus. The throughput numbers are therefore lower bounds for this hardware at stock power and single-node TP4. The lab's qualified image at stock clocks achieved 163.4 tokens per second single-stream against our 100.1; the gap is attributable to power and fabric, not to configuration error, but we have not isolated the two factors.

The FP8 attempt on GLM-5.3-Flash produced no model-quality evidence. The benchmark artefacts are zero bytes; partial fallback output was not scored. That session is not evidence about the model's quality.

The SGLang recipe that claimed 1,004.9 tokens per second was measured in a repeat loop with 100% speculative acceptance. That number measures loop replay, not decode. We could not reproduce correct output from that recipe on our hardware.

Speculative decoding across a network failed at concurrency on every engine tested. Accepted draft length is data-dependent; ranks diverge in the number of collectives they issue; the sampling collective stalls. This is a structural property of multi-node tensor parallelism with speculation, not a bug in any single engine. Single-stream speculation worked; concurrent speculation did not.

The five-fix list (disable custom all-reduce; gate the SM90 MLA backend; set block size 2304; upgrade FlashInfer; use Marlin MoE) was necessary for GLM-5.3-Flash NVFP4 under vLLM on this hardware. We do not know which fixes generalise to other models or engines.

9. Next Steps

Raising the power caps is the cheapest next experiment: the cards sat at their limits throughout, and the lab's stock-power single-stream figure is about 1.6 times ours. Adopting the qualified community image as a production seat requires displacing co-tenant services from both nodes; we will measure the fleet-wide cost before committing. The speculative-decoding failure at concurrency is structural; we will not attempt to fix it, but will note it in future multi-node work.

PureTensor operates a sovereign AI infrastructure fleet (on-premises GPU compute, storage, and serving) run day-to-day by autonomous agents under human direction. Measurements were taken 21 August to 2 September 2026; raw artefacts are retained. This paper is PT-R-2026-017.