PURETENSOR
INTELLIGENCE DISTILLED

Four Days Later

A 753-Billion-Parameter Model on One Workstation GPU, Reproduced

PT-R-2026-012 · Version 1.0 · 22 August 2026 · PureTensor Research
Series: Distributed Inference on Workstation Blackwell

Abstract

Serving a frontier-scale mixture-of-experts model has, until now, meant either a rack of accelerators or an API bill. On 18 August 2026 a Berkeley-led team published FreeToken (arXiv:2608.16157), an Apache-2.0 inference engine built on a different premise: hold the complete routed-expert weights in host memory as the source of truth, treat GPU memory as a cache whose size affects speed but never correctness, and split residual cache misses between the PCIe link and the host CPU using a closed-form ratio derived from each machine's measured bandwidths. Their headline configuration serves GLM-5.2, a 753-billion-parameter model with a 433 GB checkpoint, at 14.9 tokens per second on a single 96 GB RTX PRO 6000 Blackwell workstation GPU.

Every number in that paper is author-reported, the preprint is four days old and unrefereed, and the engine is at version 0.1.2. We had the same GPU, a host with half a terabyte of DDR5, and the identical checkpoint already on disk. So we measured it.

The claim reproduces, and on our machine it is slightly conservative: **16.9 tokens per second** warm mean across three 512-token runs, against the paper's 14.9, with warm time to first token between 7.2 and 7.8 seconds against their reported 7.5. Our first-ever cold start took 44.3 seconds, landing almost exactly on the paper's stated worst-case bound of 44 seconds, and a follow-up experiment shows that number is one-time kernel compilation rather than a steady-state cost: with compiled kernels cached, a cold start is about 8 seconds.

Two further tests complete the account. An A/B of the engine's two NVFP4 expert kernels moved the decode rate from 16.9 to 17.2 tokens per second, which is within run-to-run noise and tells us the arithmetic kernel is not where the time goes. And on a second machine carrying a pair of the same GPUs, a timeout-guarded NCCL all-reduce cleared tensor parallelism for use: 37.9 GB/s bus bandwidth with peer-to-peer transfers enabled, 30 percent above the same collective with them disabled, on hardware where a documented hang affects the identical SKU and where the capability probe reports success either way.

1. The Claim, and Why It Was Testable in an Afternoon

FreeToken's paper makes several performance claims across six test systems, from an 8 GB laptop GPU upward. Most of them we cannot check. The frontier one we could, because their flagship testbed is, almost line for line, a machine we operate: one RTX PRO 6000 Blackwell workstation GPU, PCIe 5.0 x16, and roughly half a terabyte of DDR5 behind a high-core-count

CPU. The checkpoint their configuration serves, NVIDIA's NVFP4 quantisation of GLM-5.2 at 433 GB across 47 shards, was already in our model store.

That coincidence matters beyond convenience. The viral framing of this result (“a 753B model on one GPU”) omits the half of the machine doing most of the work. The expert pool lives in host memory, so host capacity is the binding constraint, and host memory bandwidth is a first-order performance term. A 96 GB GPU in a 128 GB desktop cannot run this configuration at all. Reproductions on matched hosts are the only ones that test the paper's actual claim, which is one reason we published this one.

2. The Mechanism in One Page

FreeToken is a from-scratch single-GPU serving engine for mixture-of-experts models. Four design decisions carry the result.

The CPU-resident expert pool is the source of truth. All routed-expert weights stay pinned in host memory. Whatever GPU memory remains after the dense weights becomes one shared expert cache spanning all layers. Cache capacity therefore affects throughput, never correctness or model fidelity: the engine serves the unmodified checkpoint bit-exactly.

Decode follows the router, not a predictor. A global least-recently-used policy tracks which experts the model actually selects, in place of the predictive prefetch that prior systems in this line invested in.

Misses are split, not fetched. When routed experts are absent from the cache, a closed-form policy divides them between two concurrent paths: some stream over PCIe into the cache, the rest are executed in place on the CPU. The split ratio comes from two per-machine measured bandwidths, taken once by a calibration tool on real tensor shapes. On our host that calibration measured the CPU expert kernel at 170.1 GB/s of effective bandwidth and the PCIe host-to-device path at 57.0 GB/s, a 3.26x ratio that selects the hybrid backend, with the CPU executing roughly seven in ten misses in place.

The control path lives on the GPU. Routing, cache lookup, eviction, and the miss split are resolved by a device-resident kernel inside a captured CUDA graph, which is the engine's stated differentiator against prior hybrid CPU/GPU systems whose host-side heuristics cannot be captured that way.

3. Method

The serving host carries one RTX PRO 6000 Blackwell workstation GPU (96 GB), PCIe 5.0 x16, and 503 GB of DDR5. The engine was FreeToken v0.1.2 from PyPI, CUDA 13 toolkit, serving nvidia/GLM-5.2-NVFP4 unmodified. Loading takes about four minutes: dense weights to the GPU in seconds, then the 419 GB expert pool streamed from NVMe into pinned host memory at roughly 4.5 GB/s. At steady state the server held 88.5 GB of GPU memory and about 402 GB of host memory.

Measurement was deliberately simple: streaming chat completions of 512 tokens against a competition-mathematics prompt, one cold run then three warm runs, timing every token. Time to first token is the interval from request to first streamed token; decode rate is tokens after the first divided by the time they took. This mirrors the paper's single-stream protocol at batch size one. It is a spot check, not a benchmark suite, and section 7 treats it as such.

4. The Reproduction

Run	TTFT (s)	Decode (tok/s)	Tokens
Cold (first ever)	44.31	17.0	511
Warm 1	7.80	17.4	511
Warm 2	7.23	16.1	511
Warm 3	7.24	17.2	511

Warm mean decode is **16.9 tokens per second** against the paper's 14.9 on the same GPU model, a 13 percent margin in the paper's favour to claim and ours to observe. Warm time to first token, 7.2 to 7.8 seconds, brackets their reported mean of 7.5. The cold start of 44.3 seconds sits on their stated worst-case bound of “under 44 seconds” almost to the decimal.

Two readings of the margin are available and we cannot fully separate them. Our host measured a CPU expert kernel at 170.1 GB/s where the paper's flagship host reports 178 GB/s, so raw host bandwidth does not explain us being faster. Our host does differ in CPU generation and memory configuration, and the engine had four more days of commits than the paper's evaluation build. We note the margin and do not lean on it. The claim that matters, interactive-rate decode of a 753B model from one workstation GPU, holds on independent hardware.

For calibration, the paper's own baseline on this configuration is llama.cpp at 7.3 tokens per second, a comparison we did not rerun.

5. The Kernel That Did Not Matter, and What Cold Start Actually Is

FreeToken defaults its NVFP4 expert arithmetic to a portable Triton kernel. The engine also ships a FlashInfer-backed kernel it describes as the fast path for our GPU generation on CUDA 13. Our reproduction ran the default, so the obvious follow-up was a one-flag A/B on the same seat, same checkpoint, same protocol.

Arm	Warm mean decode (tok/s)	Warm TTFT (s)	Cold TTFT (s)
Triton (default)	16.9	7.2–7.8	44.3
FlashInfer	17.2	7.2	8.1

The decode difference, 16.9 against 17.2, is within the noise of three-run means and we treat it as no effect. That is itself informative: at this cache ratio the decode budget is dominated by servicing expert misses over PCIe and the CPU path, so a faster GEMM kernel has little to act on. Anyone tuning this engine should spend their effort on the cache and the miss split, not the arithmetic.

The column that does move is the interesting one. The FlashInfer arm's first request of a fresh server process completed its prefill in 8.1 seconds, because the just-in-time kernel compilation that consumed the original 44-second cold start was already cached on disk from the earlier session. The scary first-boot number is compilation, paid once per machine and kernel configuration, not a recurring cost of the architecture. Steady-state cold start, a fresh process with warm compile caches, is about 8 seconds; the paper's sub-44-second framing describes the worst case, and an operator's daily experience is five times better.

6. Clearing the Second Card

The serving host's GPU has a twin on another machine in the lab: a pair of the Max-Q variant of the same card on one board, PCIe 5.0, no NVLink. Tensor parallelism had never run on that pair.

That gap was worth closing with a measurement rather than an assumption, because this exact SKU has a documented failure mode: an open NCCL issue reports all-reduce hanging indefinitely with peer-to-peer transfers enabled on dual RTX PRO 6000 Blackwell systems, triggered by platform IOMMU and ACS state, and reproduced by multiple parties on boards close to ours. The trap in that report is that every capability probe passes anyway: the driver advertises peer-to-peer support, the topology tools print OK, and the hang appears only when a real collective runs.

So we ran the real collective, with a timeout guard so that a hang would report as a result rather than a stuck terminal: the standard NCCL all-reduce benchmark across both GPUs, sweeping message sizes to 512 MB, once with peer-to-peer enabled and once with it disabled.

Configuration	Peak bus bandwidth (GB/s)	Outcome
Peer-to-peer enabled (default)	37.9	Completed, no hang
Peer-to-peer disabled	29.1	Completed

No hang, on a driver newer than every published failure report, despite our platform carrying the ACS state the reports implicate. The 30 percent margin is the important secondary reading: it proves direct GPU-to-GPU transfers are genuinely engaged rather than silently staged through host memory. It also means the widely-circulated advice for this card class, disabling peer-to-peer as a precaution, would cost real bandwidth on a healthy platform. The correct order of operations is to run the sixty-second collective first and only reach for the workaround if it actually hangs.

The pair is now cleared for tensor-parallel serving, with a recorded baseline to compare against after any driver or firmware change.

7. What These Numbers Will Not Carry

Our reproduction is one prompt class, three warm runs, batch size one, on one host. Three-run means on this protocol carry at least a few tenths of a token per second of noise, which is why we read the kernel A/B as no effect rather than a 1.8 percent win. The 13 percent margin over the paper's figure is real on our machine but single-host; we would not quote it as a property of the engine.

We reproduced FreeToken's number, not its comparisons. The llama.cpp baseline of 7.3 tokens per second is the paper's measurement, and their evaluation harness for agentic workloads is not shipped, so the relative claims against other engines remain author-reported.

The configuration we validated is single-stream. FreeToken's design centre is a handful of concurrent requests, its own paper measures per-request throughput only, and nothing here says what happens to the expert cache hit rate under concurrent load with divergent routing. Our batch and fan-out serving stays on engines built for it; this result adds a capability, interactive frontier-scale serving on one card, without displacing anything.

Finally, the host is half the story. This configuration needs host memory comfortably above the checkpoint size, 433 GB here, and the engine's own repository currently documents no memory requirements at all. Anyone reading the headline as “a gaming PC runs 753B” should read the spec sheet of the flagship testbed first.

8. What This Changes

For a lab that owns workstation-class Blackwell hardware, the practical reading is narrow and useful: frontier-scale models are now a one-command interactive seat on a single GPU, provided the host memory exists, at rates comfortable for a single agent or operator session.

The published claim survived contact with independent hardware four days after release, which is worth recording in a field where impressive numbers often do not.

The two supporting results generalise further than the headline. A first-boot measurement of a JIT-compiled engine is a measurement of the compiler, not the engine; decompose it before quoting it. And on multi-GPU workstation platforms, capability probes are not evidence: the collective either runs or it does not, the test takes a minute, and the difference between a probe and a proof was, on our hardware, the difference between folklore and 30 percent of the bus.

References: FreeToken, “Efficient Edge-Native MoE Serving with Bandwidth-Adaptive Execution”, arXiv:2608.16157 (17 August 2026). Engine: github.com/FlashML-org/FreeToken, v0.1.2, Apache-2.0. Checkpoint: nvidia/GLM-5.2-NVFP4. NCCL hang report: NVIDIA/nccl issue 1999. Companion papers in this series: PT-R-2026-004 through PT-R-2026-007.