SLO-Tuned Serving Profiles

One setting measured against the workload's own SLO sustained 1.304× the long-prompt traffic of stock vLLM per GPU

Pranav Varshney
Nodus Compute

Viswa Kotra
Nodus Compute

Abstract

Most inference engines run with their default settings, and those defaults are not chosen for any particular workload or SLO. In this study, a research harness from Nodus, the infrastructure execution layer for AI workloads, measured serving configurations on a real GPU against the workload's own traffic shape and SLO and kept the one that sustained the most load. On one GPU serving Qwen3-8B-FP8 with stock vLLM, for long-prompt RAG traffic with a 2000 ms TTFT and 50 ms TPOT SLO, the configuration found this way, a prefill chunk budget of 1024 tokens, sustains 1.304× the load of the stock defaults at the SLO. The researcher measured 1.304 on 3 seeds and an independent verifier measured 1.303 on 3 fresh seeds, with every seed above the default and an A/A repeat of the default identical. A hand-tuned configuration from public tuning guidance matched the defaults exactly. For chat traffic the defaults were already at the knee: the same setting gave 1.0000 on both fresh seeds, and the search kept the stock configuration.

1 Introduction

A serving engine's default settings are a compromise across workloads. A deployment that serves long prompts under a tight decode SLO can be limited by a setting that is fine for short chat, and the operator rarely knows which. Each extra request a GPU can carry at the SLO is capacity that does not have to be bought.

The setting at work here, the token budget for chunked prefill, is known to trade time to first token against time per output token [1]. Our contribution is not a new knob. In this study, a research harness measured each workload on a real GPU against its own SLO, with seeded production traffic, an A/A control and fresh-seed confirmation, and changed a setting only when the measurement showed a gain. We report one workload where it does and one where it does not.

The metric is the sustainable load at the SLO: the highest request rate per GPU at which p95 TTFT and p95 TPOT both meet the SLO, confirmed by a recheck at that load.

2 Experimental setup

Workloads. Both run Qwen3-8B-FP8 [2] on stock vLLM [3] on one GPU. The long-prompt RAG profile replays the Mooncake conversation trace [4] with shared prefix blocks, a mean of 2828 input and 174 output tokens, and an SLO of 2000 ms TTFT and 50 ms TPOT at p95. The chat profile replays a public production conversation trace [5] with a mean of 1056 input and 251 output tokens and an SLO of 1000 ms TTFT and 40 ms TPOT.

Arms. The default arm is stock vLLM, which on this GPU uses a prefill chunk budget of 8192 tokens and up to 1024 sequences. The expert arm is a configuration written in advance from public vLLM tuning guidance. The Nodus arm is the configuration found by measuring candidate settings on the GPU with short probes at the workload's own SLO, then confirming the winner with full-length runs: a prefill chunk budget of 1024 tokens. The default arm runs twice per seed as an A/A control, and every arm starts from a freshly started engine.

Gate. The pre-registered gate requires a mean of at least 1.10× the default's sustainable load, a gain on every seed, and a gain larger than the A/A spread. The researcher used seeds 1, 2 and 3, and an independent verifier reran the comparison with the same code on fresh seeds 17, 23 and 31.

3 Results

Table 1: Sustainable load at the SLO on one GPU relative to stock vLLM defaults, per seed.

Profile and arm	Seeds	Ratio per seed	Mean
RAG, Nodus setting	1, 2, 3	1.3975, 1.1180, 1.3975	1.304
RAG, Nodus setting, verifier	17, 23, 31	1.2500, 1.4777, 1.1822	1.303
RAG, expert configuration	1, 2, 3	1.0000, 1.0000, 1.0000	1.0000
RAG, default repeated (A/A)	1, 2, 3	1.0000, 1.0000, 1.0000	1.0000
Chat, the same setting, verifier	17, 23	1.0000, 1.0000	1.0000



Figure 1: Sustainable load at the SLO relative to stock defaults, mean over seeds.

Long-prompt RAG. With the default chunk budget of 8192 tokens, long prefills share each engine step with ongoing decodes and push the TPOT of running requests over the 50 ms SLO first. A budget of 1024 tokens splits those prefills into smaller pieces, so decodes stay within their SLO at higher load while TTFT rises toward, but stays inside, its 2000 ms limit. The sustainable load rises from 5.6371, 5.9605 and 5.6371 req/s to 7.878, 6.664 and 7.878 req/s on seeds 1, 2 and 3. On the verifier's fresh seeds it rises from 5.3312, 4.7684 and 6.3024 req/s to 6.664, 7.0463 and 7.4506 req/s.

The expert configuration did not help. Public tuning guidance suggests larger budgets for better TTFT and smaller ones for better TPOT, and the expert arm, written before any measurement, matched the defaults on every seed. Finding the right direction for this workload took a measurement against its own SLO.

Chat is already at the knee. On chat traffic the same budget of 1024 tokens gives exactly the default's sustainable load on both fresh seeds, 15.3868 and 13.0156 req/s. The search kept the stock configuration for this profile, which is the right answer: a measured profile should change a setting only when the workload gains from it.

4 Independent verification

A separate verifier reran the RAG comparison with the researcher's code on 3 fresh seeds and measured 1.2500, 1.4777 and 1.1822, a mean of 1.303 against the researcher's 1.304, with every seed above its default. Its A/A repeat of the default was identical to the first run on every seed. On chat it found no gain from the same setting on 2 fresh seeds, consistent

with the researcher's search keeping the stock configuration there. The gain uses a known technique [1], and the search procedure overlaps with existing configuration search tools for vLLM.

5 Conclusion

On one GPU, a serving setting chosen by measuring the workload against its own SLO carries 1.304× the long-prompt RAG traffic of stock vLLM, reproduced at 1.303× on fresh seeds, while the same measurement correctly leaves chat traffic on its defaults. The gain comes from one known setting applied where the measurement shows it helps. The result covers one model on one GPU type and two traffic profiles. For fleets that serve long-prompt traffic on default settings, the result suggests a short measured tuning pass could raise capacity per GPU by about a third.

References

[1] A. Agrawal et al. Taming Throughput-Latency Tradeoff in LLM Inference with Sarathi-Serve. OSDI 2024.

[2] A. Yang et al. Qwen3 Technical Report. arXiv preprint, 2025.

[3] W. Kwon et al. Efficient Memory Management for Large Language Model Serving with PagedAttention. SOSP 2023.

[4] R. Qin et al. Mooncake: A KVCache-centric Disaggregated Architecture for LLM Serving. FAST 2025.

[5] P. Patel et al. Splitwise: Efficient Generative LLM Inference Using Phase Splitting. ISCA 2024.