Customers per GPU

48 to 56 customers' fine-tuned models on one GPU, each within its own SLO

Pranav Varshney
Nodus Compute

Viswa Kotra
Nodus Compute

Abstract

Customers who fine-tune a model usually get an endpoint of their own, and most such endpoints see a few requests per minute, so the GPU behind each one sits mostly idle. Serving many LoRA adapters of one base model from one engine is established, but how many customers one GPU can hold while every customer keeps its own latency SLO is rarely measured. Nodus, the infrastructure execution layer for AI workloads, measured it on one GPU with stock vLLM, Qwen3-8B and 256 customer adapters, with replayed production traffic, three SLO classes and a cold prefix cache. With the default 2048-token step budget the GPU held a mean of 32.8 customers across 5 seeds. With a 256-token step budget chosen for the customers' SLOs, it held 48 to 56 customers, a mean of 52.8 and 1.61× the default budget. Stock defaults held 4. An independent verifier reproduced the result on 2 fresh seeds and showed that a plain fixed 256-token budget gives exactly the same count, so the gain comes from choosing stock vLLM's max-num-batched-tokens setting of 256 for these SLOs, not from anything only Nodus has. The pre-registered gate of 2× on every seed was met on 2 of 5 seeds and failed.

1 Introduction

Fine-tuning is now a routine customer workload, and a fine-tuned model is often served from an endpoint that holds a GPU for one customer. Most of these customers send a few requests per minute, so a GPU per customer leaves almost all of each GPU idle. LoRA [1] adapters of one base model can share one engine, and multi-adapter serving is well studied [2] [3] and supported by stock vLLM [4]. What an operator needs to know is how many customers fit on one GPU while each customer, with its own traffic and its own latency target, still meets its SLO.

We measure that number directly. Each customer has its own adapter, its own request rate, its own replayed traffic and one of three SLO classes, and a configuration passes only if every single customer meets its own TTFT, TPOT and completion targets. We compare stock vLLM defaults, stock multi-adapter settings with the default step budget, and an SLO profile, a step budget chosen for the customers' SLOs.

The metric is customers per GPU, the highest number of customers on one GPU at which every customer meets its own SLO.

2 Experimental setup

System. One host with one GPU, stock vLLM 0.30.0 and Qwen3-8B [5] in BF16. The engine holds 16 adapters on the GPU and 512 in host memory, and 16 rank-64 GPU adapter slots cost 4.7 GiB.

Customers. 256 LoRA adapters on all 7 projections, with ranks 8, 16, 32 and 64. Request rates are lognormal with a median of 2 requests per minute, clipped to 0.5 to 30. 40% of customers are interactive with a TTFT p95 of 1 s and a TPOT p95 of 50 ms, 40% are standard at 2.5 s and 80 ms, and 20% are relaxed at 10 s and 150 ms, and every customer must complete at least 99% of its requests. Each customer replays its own segment of public production conversation, code and burst traces, with prompts up to 4096 tokens and answers of 16 to 512 tokens.

Measurement. Each window has 60 s of arrivals and up to 40 s of drain. Customers per GPU is found on a ladder of 1, 4, 8, 12, 16, 20, 24, 28, 32, 40, 48, 56, 64 and 80 customers. Prompt content is salted per window, so prefix-cache hits were 0.0 in all 49 researcher windows and all 21 verifier windows, and every configuration and seed runs on a freshly started engine. Arrivals and lengths depend only on the customer count and seed, so every configuration sees identical load.

Configurations. Stock defaults use vLLM's defaults with one adapter slot. Default budget uses 16 GPU adapter slots, 512 host slots, SLO-class priorities and vLLM's default step budget of 2048 tokens. The 512-token budget lowers that budget to 512. The SLO profile lowers it to 256, the value of stock vLLM's max-num-batched-tokens setting chosen for these SLOs. On seeds 1 to 3 the SLO profile applied the 256-token budget while an interactive request was decoding, which was 72.8% to 97.3% of scheduler steps, and the verifier ran a plain fixed 256-token budget as a control on seeds 4 and 5.

Gates. The pre-registered multiplier gate requires 2× the customers of the default budget on every seed. The gain gate requires more customers on every seed, with the gain above the spread of repeated runs.

3 Results

Table 1: Customers on one GPU with every customer within its own SLO, per seed.

Configuration	Seed 1	Seed 2	Seed 3	Seed 4	Seed 5	Mean
SLO profile, 256-token budget	48	56	56	56	48	52.8
Fixed 256-token budget, verifier control	n/a	n/a	n/a	56	48	n/a
512-token budget	48	40	48	n/a	n/a	45.3
Default 2048-token budget	40	28	40	32	24	32.8
Stock defaults, one adapter slot	4	n/a	n/a	n/a	n/a	n/a

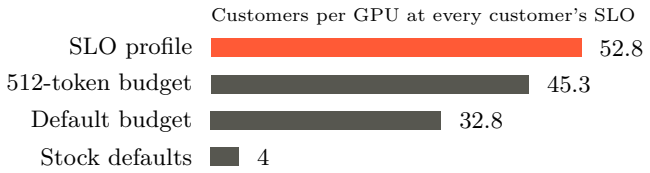


Figure 1: Mean customers on one GPU with every customer within its own SLO.

48 to 56 customers on one GPU. With the SLO profile the GPU kept every customer within its own SLO at 48, 56 and 56 customers on seeds 1 to 3, and at 56 and 48 on the verifier's fresh seeds 4 and 5, a mean of 52.8. The default budget held 40, 28, 40, 32 and 24, a mean of 32.8, so the SLO profile holds 1.61× the customers by ratio of means, with per-seed ratios of 1.20, 2.00, 1.40, 1.75 and 2.00. Stock defaults held 4 customers on seed 1, where the SLO profile held 48, 12× as many. A GPU per customer would use 48 to 56 GPUs for the same customers.

The step budget is the setting that matters. Every failure under the default budget had the same shape: an interactive customer missed its TPOT target by a few milliseconds while other customers' long prompts were being prefilled in the same steps. At 48 customers on seed 1, the busiest interactive customer, with 30 requests per minute and 16-token answers,

had a TPOT p95 of 50.8 ms under the default budget against a 50 ms target, and 42.9 ms under the SLO profile. A 512-token budget held 48, 40 and 48 customers on seeds 1 to 3, a mean of 45.3, and the 256-token budget held more.

A plain setting is enough. The verifier's fixed 256-token budget, with no condition on which customers are decoding, held exactly the same number of customers as the SLO profile on both fresh seeds, 56 and 48, with the busiest interactive customer's TPOT p95 at 42.0 against 42.0 ms and 46.0 against 46.9 ms in paired windows. Applying the budget only while interactive requests decode adds no measured gain, so the profile is a single stock engine setting, and the gain comes from choosing it for the SLOs, not from anything only Nodus has.

Repeats and cost. Repeating a configuration on a fresh engine gave the same customer count every time, 40 against 40, 32 against 32, 48 against 48, 56 against 56 and 48 against 48, so the gains are above the repeat spread of 0 grid steps and the gain gate passes on every seed. The cost is TTFT: the busiest interactive customer's TTFT p95 rose from about 100-190 ms to about 430-880 ms, still inside its 1 s target, and TTFT is the next limit at 64 customers. SLO-met requests per second at the highest passing count rose from 2.35, 1.783 and 1.8 to 2.467, 2.533 and 2.683 on seeds 1 to 3, a smaller gain than the customer count because most added customers send few requests.

Pre-registration. The pre-registered multiplier gate of 2× on every seed was met on 2 of 5 seeds and failed. Outputs were checked by exact match on 64 greedy prompts and differed from the default budget no more than two identical engines differ from each other, 0.766 to 0.828 against 0.781, but the adapters have random weights, so this check cannot detect a real quality change.

4 Independent verification

A separate verifier recomputed all 49 researcher windows from the per-request logs with its own SLO code and matched every pass or fail and every customer count. It then reran the default budget and the SLO profile on fresh seeds 4 and 5, in alternating order on fresh engines, and measured 56 against 32 and 48 against 24, plus the fixed 256-token control. The verifier classed the result as a measurement, not a new technique, since chunked-prefill step budgets sized to a latency target [6] and step budgets conditioned on the service classes in flight [7] are published. The verifier also noted that every seed uses one draw of the customer fleet, so the count is set by a few interactive customers and could move with a different fleet.

5 Conclusion

On one GPU with stock vLLM, 48 to 56 customers' fine-tuned Qwen3-8B adapters were served from one engine with every customer within its own SLO, 1.61× the customers of the default step budget across 5 seeds and 12× stock defaults. The gain comes from one setting, a 256-token step budget, and costs interactive TTFT that stays within target. The result covers one GPU type, one base model, random-weight adapters and one fleet draw. For a platform that serves many low-traffic fine-tuned models, customers can share GPUs at their own SLOs instead of each holding one.

References

[1] E. Hu et al. LoRA: Low-Rank Adaptation of Large Language Models. ICLR 2022.

[2] Y. Sheng et al. S-LoRA: Serving Thousands of Concurrent LoRA Adapters. MLSys 2024.

[3] L. Chen et al. Punica: Multi-Tenant LoRA Serving. MLSys 2024.

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

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

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

[7] K. Goel et al. Niyama: Breaking the Silos of LLM Inference Serving. arXiv preprint, 2025.