Isaac Levinsky · MCMLV1 LLC www.linkedin.com/in/isaaclevinsky/ · mcmlv1.com · github.com/IsaacLevinsky

PROJECT AT A GLANCE · NOVEMBER 2026

Same model. Same H100. Three LLM serving engines.

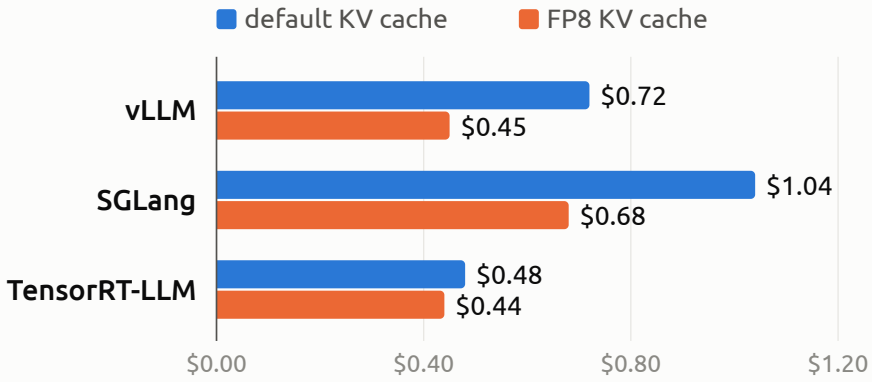
I measured vLLM, SGLang and TensorRT-LLM serving OpenAI's gpt-oss-120b on one NVIDIA H100 — accuracy, cost per token, repeatability under load, and GPU profiles — and found the settings that change cost and behaviour.

3 vs 82
distinct outputs, same request ×1,000 under load — SGLang vs vLLM, defaults

–37%
cost per token, FP8 KV cache on vLLM: 64 users fit vs 32; no detectable accuracy change

1 flag
made vLLM's batch-invariant mode exact on gpt-oss-120b

$93
of cloud GPU time, under a hard $100 cap — every run sealed



USD per 1M output tokens, at the most users that fit

FP8 KV cache doubles how many users fit on the GPU, so each token costs less. Accuracy across all six engine settings stayed within run-to-run noise.

Built

- Benchmark harness in Rust
- Cloud runner with a hard budget cap and two independent watchdogs
- Rehearsed locally before paying for GPU time

Measured

- Accuracy with the MLPerf method, against a measured noise floor
- Cost and latency from 1 to 64 users
- Repeatability under load; Nsight Systems profiles

Shipped

- Public report and summary
- Open-source fix proposed to SGLang (PR #52026, open)
- Kubernetes deployment with the NVIDIA GPU Operator

What this means for you

Hiring teams

- Serving-engine depth: vLLM, SGLang, TensorRT-LLM
- Root-causing with profiles and controlled experiments
- Open to GPU inference and ML infrastructure roles (W-2)

Contract work

- Evaluate and tune your serving stack
- Cut cost per token without losing accuracy
- Reproducibility checks for regulated use

GPU providers

- Tested serving profiles per GPU
- Acceptance checks a non-developer can run
- Clear cost-per-token numbers for customers

Full report: mcmlv1.com/h100-serving-engines · Measured on 1× H100 SXM5 80GB with vLLM 0.33.0, SGLang 0.5.23, TensorRT-LLM 1.4.0rc2; one GPU type and one model.