



Performance Evaluation of Scalable Computing Systems for Large Language Model Inference

Zhang Han¹, Summing Wang¹, Minhee Kim², Si-Young Noh³, Soon-Eun Kwon³, Jae-Hoon Lee³, Tae-Young Byun^{3*}

¹*Dept. of Computer Software, Daegu Catholic University, Gyeongsan, Rep. of Korea*

²*Sillasystem Co., Ltd., Daegu, Rep. of Korea*

³*School of Computer Software, Daegu Catholic University, Gyeongsan, Rep. of Korea*



ABSTRACT

This paper evaluates single- and dual-node NVIDIA DGX Spark systems for local LLM inference using OpenAI gpt-oss-120b. Both configurations use the same vLLM serving stack, while the dual-node system employs pipeline parallelism across two DGX Spark nodes connected by ConnectX-7. Four workload classes and five concurrency levels are evaluated using TTFT, generation throughput, end-to-end latency, tail latency, and aggregate throughput. Under the representative W4 workload at concurrency 16, the dual-node system improves mean TTFT by 10.52%, generation throughput by 13.78%, aggregate throughput by 13.92%, and P99 end-to-end latency by 13.91%, although mean end-to-end latency increases by 6.69%. Overall, the results show that dual-node scaling is workload-dependent and is most beneficial for long-generation and high-concurrency services.

INTRODUCTION

This study evaluates OpenAI gpt-oss-120b, a Mixture-of-Experts model with 116.83 billion total parameters, 5.13 billion active parameters per token, and an approximately 60.8 GiB checkpoint. The central question is whether adding a second DGX Spark provides a meaningful inference advantage when the same model can already run on a single node. Dual-node execution increases compute and memory capacity but also introduces pipeline, communication, and synchronization overheads, so near-linear scaling cannot be assumed. This paper makes four main contributions. First, it establishes a reproducible Single- versus Dual-node DGX Spark testbed using identical models, serving configurations, and workloads. Second, it implements native multi-node vLLM inference with TP=1 and PP=2 over ConnectX-7. Third, it evaluates four workload types across five concurrency levels using TTFT, generation throughput, end-to-end latency, tail latency, aggregate throughput, GPU utilization, and unified-memory utilization. Finally, it identifies the workload conditions in which Dual Mode is beneficial and translates the measured results into practical deployment guidance for closed local-LLM services.

SYSTEM ARCHITECTURE

A. Single-Node Baseline

The baseline runs one vLLM API server and one gpt-oss-120b model instance on Spark 1. The distributed executor backend is native multiprocessing, but the parallelism configuration is TP=1 and PP=1 with nnodes=1. No internode synchronization is required. The single-node configuration therefore establishes the reference for both interactive and capacity-oriented metrics.

B. Dual-Node Native-Multiprocessing Configuration

The final dual-node system uses vLLM native multiprocessing rather than Ray/KubeRay. Spark 1 acts as rank 0, hosts the OpenAI-compatible API server, and executes pipeline stage 0. Spark 2 acts as rank 1 in headless mode and executes pipeline stage 1. The configuration is TP=1, PP=2, and nnodes=2. The high-speed ConnectX-7 interfaces use link-local IPv4 addresses 169.254.199.119 (Spark 1) and 169.254.80.141 (Spark 2). VLLM_HOST_IP, NCCL_SOCKET_IFNAME, and GLOO_SOCKET_IFNAME are explicitly set so that model-parallel and coordination traffic use the ConnectX-7 path rather than the management network.

$$T_{E2E}(r) = T_{\text{prefill}}(r) + T_{\text{decode}}(r) + T_{\text{comm}}(r) + T_{\text{sync}}(r) + T_{\text{queue}}(r) \quad (1)$$

Here, T_{prefill} denotes prompt-processing time, T_{decode} denotes output-token generation time, T_{comm} captures inter-node transfer cost, T_{sync} represents distributed synchronization overhead, and T_{queue} is request waiting time. This decomposition explains why a second node can reduce compute or memory pressure while still increasing perrequest overhead through communication and synchronization.

EXPERIMENTAL METHODOLOGY

Both systems use the NVIDIA vLLM container nvc.io/nvidia/vllm:26.08-py3 with vLLM 0.27.1-series software. The model is openai/gpt-oss-120b. The maximum model length is fixed at 6,144 tokens, max_num_seqs at 16, and gpu_memory_utilization at 0.65 for both Single and Dual. The common memory target matters because an earlier 0.70 configuration exceeded the worker node's free unified-memory headroom; therefore, all final comparative results in this paper use the re-measured 0.65 condition. Eager execution is enabled, and both configurations expose the same OpenAI-compatible endpoint to the benchmark driver.

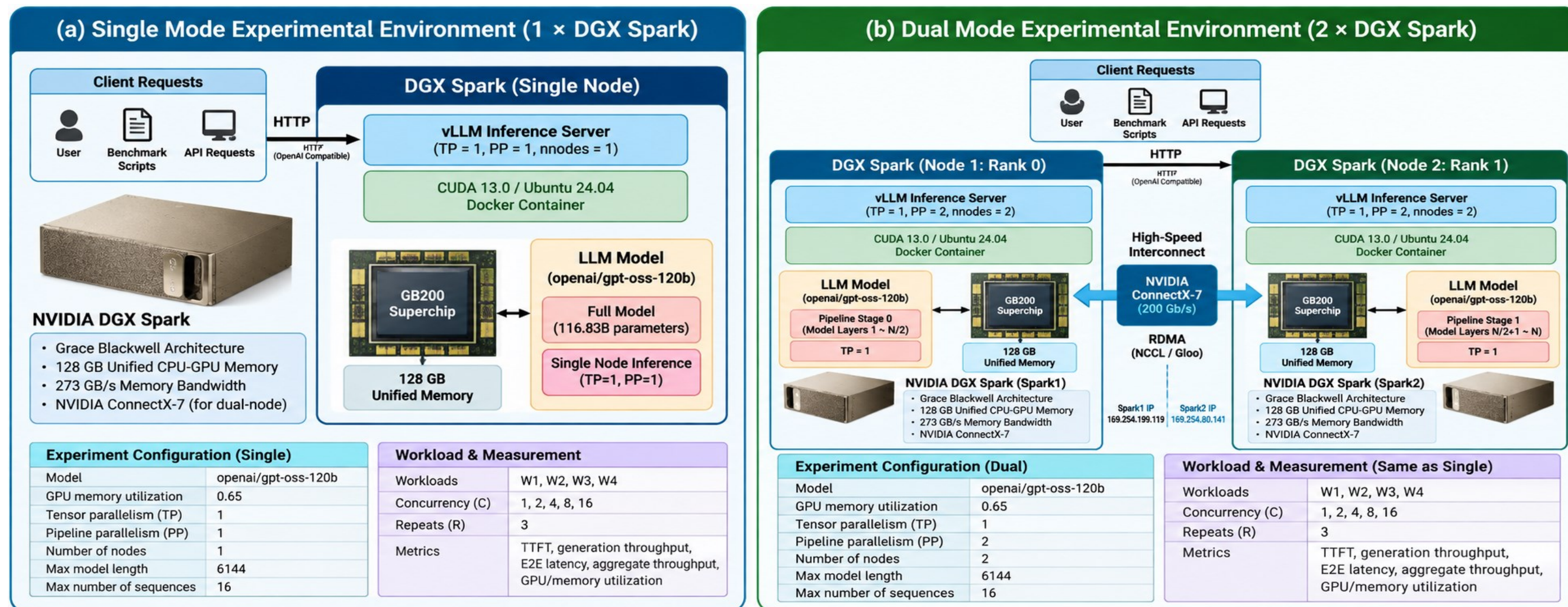


Fig. 2. System architecture for single mode and dual mode

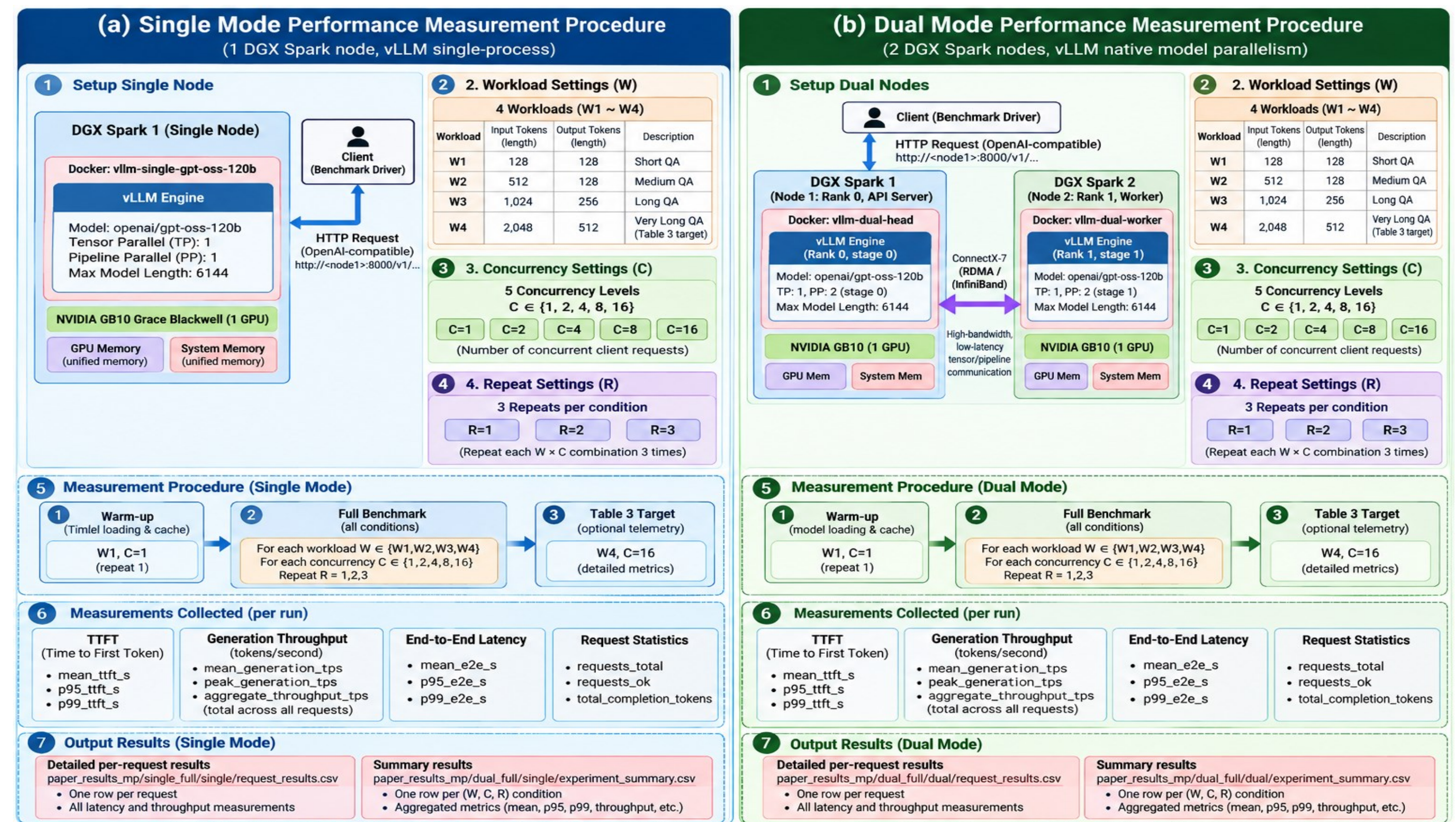


Fig. 3. Performance measurement procedure

ANALYSIS AND RESULTS

Table 1. Final Single-vs-Dual Results for W4 at Concurrency 16, Averaged Over Three Repeats

Positive performance change denotes improvement; for latency and memory-pressure metrics, lower values are better.

Metric	Single	Dual	Performance Change
Mean GPU utilization (%)	94.485	93.484	-1.06%
Peak GPU utilization (%)	96.000	96.000	+0.00%
Mean unified-memory utilization (%)	75.780	73.862	+2.53%
Peak unified-memory utilization (%)	76.298	76.020	+0.36%
Mean TTFT (s)	0.441	0.394	+10.52%
P95 TTFT (s)	0.548	0.530	+3.16%
P99 TTFT (s)	0.553	0.541	+2.25%
Mean generation TPS	13.773	15.672	+13.78%
Peak generation TPS	18.614	21.666	+16.40%
Mean E2E latency (s)	15.371	16.400	-6.69%
P95 E2E latency (s)	27.351	26.714	+2.33%
P99 E2E latency (s)	33.261	28.635	+13.91%
Aggregate throughput (tokens/s)	185.612	211.452	+13.92%

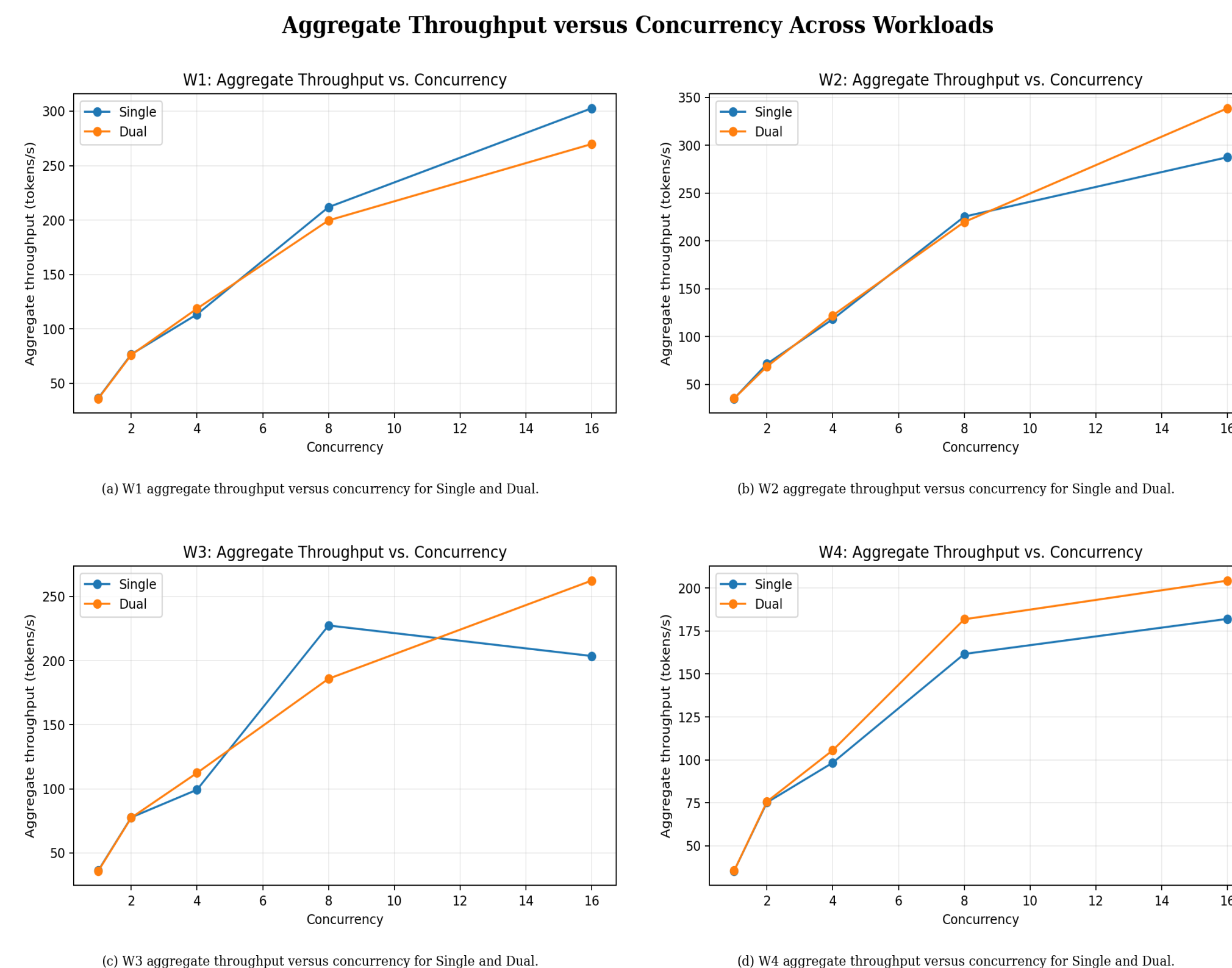


Fig. 4. Aggregate throughput versus concurrency for Single and Dual

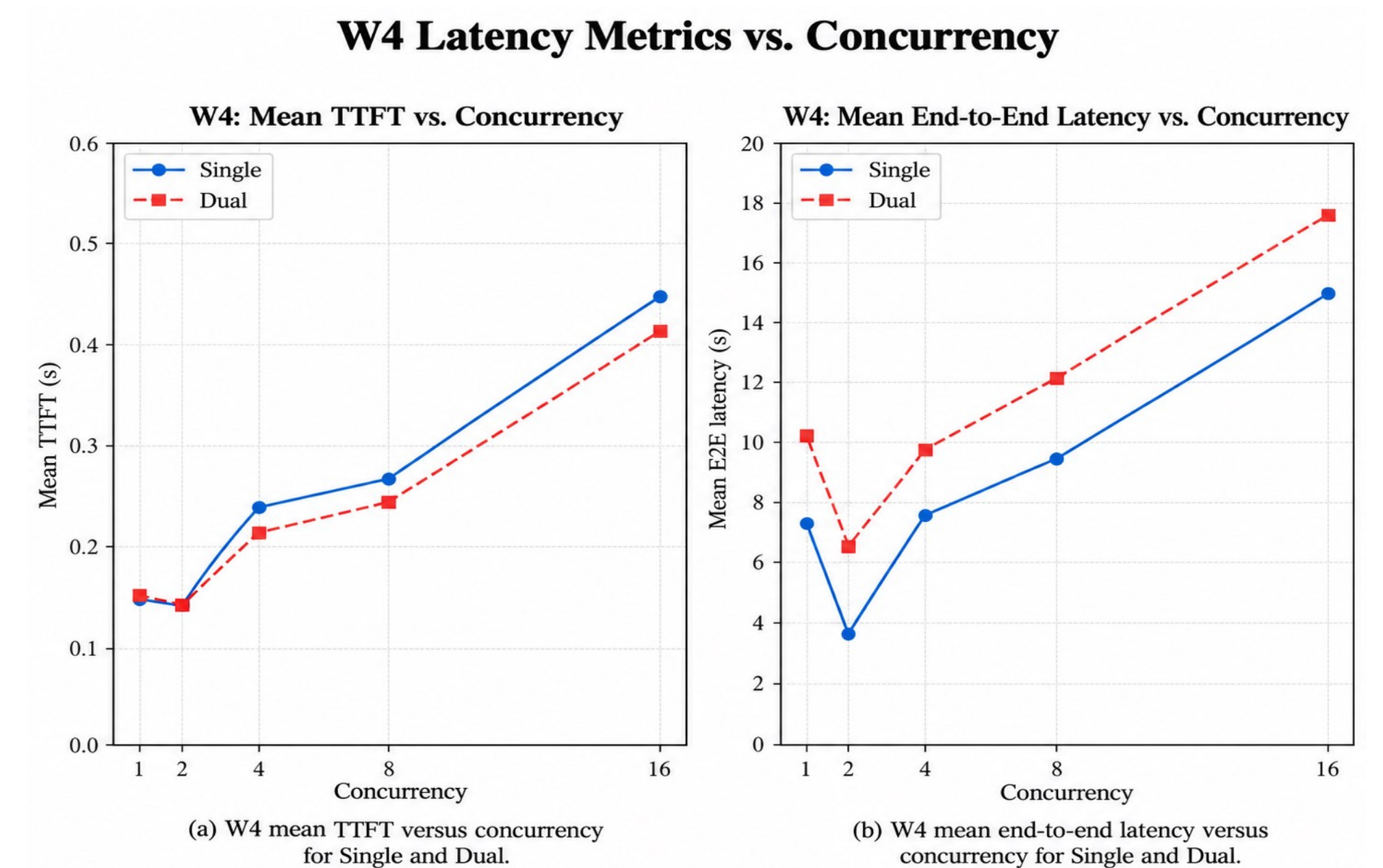


Fig. 5. W4 latency metrics versus concurrency for single and dual

CONCLUSIONS

This paper experimentally characterizes single- and dual-node DGX Spark inference for gpt-oss-120b under controlled, identical serving parameters. The final dual-node implementation uses vLLM native multiprocessing with TP=1 and PP=2 over ConnectX-7. The results reject the assumption that a second node produces uniform or near-linear speedup. In the representative W4/C=16 condition, Dual improves mean TTFT by 10.52%, mean generation throughput by 13.78%, aggregate throughput by 13.92%, and P99 E2E latency by 13.91%, but mean E2E latency worsens by 6.69%. Across the full C=16 matrix, throughput ranges from a 10.85% loss on W1 to a 28.87% gain on W3. The key finding is therefore workload selectivity: dual-node pipeline parallelism is most valuable when concurrency and generation length are sufficient to amortize communication and pipeline overhead. From a service deployment perspective, Single mode is better aligned with latency-oriented interactive local assistants, whereas Dual mode is better aligned with long-generation, document-centric, and high-concurrency services. A closed local-LLM platform can therefore benefit from workload-aware routing that selects the execution topology according to prompt length, expected output length, queue state, and service-level objectives. These findings provide both a quantitative scaling characterization and practical guidance for deploying gpt-oss-120b on compact institution controlled infrastructure.

Acknowledgement. This research was supported by the Gyeongbuk ANCHOR system through the Gyeongbuk ANCHOR CENTER, funded by the Ministry of Education(MoE) and Gyeongsangbuk-do, Republic of Korea (2026-ANCHOR-15-107).