Benchmarking Large Language Model Inference on Kubernetes: Comparing Triton, TensorRT-LLM, and vLLM on Llama 3.1 and Granite 3.1

C. Kang, C. Latschkowski, D. Marcus

02/27/2025

Abstract

Large Language Model (LLM) inference is a crucial aspect of deploying AI systems efficiently. In this paper, we benchmark the inference performance of three popular inference engines—Triton, TensorRT-LLM, and vLLM—on two leading LLM architectures: Llama 3.1 and Granite 3.1. Our evaluation includes latency, throughput, memory consumption, and scalability across two different platforms, Linux, Kubernetes configurations. The results provide insights and optional performance configurations into the trade-offs among these frameworks, guiding AI practitioners in selecting the most suitable solution for their needs.

1 Introduction

Large Language Models (LLMs) have seen widespread adoption, with their performance heavily influenced by inference efficiency. Combining the right combination of inference framework, model architecture and platform is essential to maximize throughput and minimize latency. In this paper, we conduct a systematic benchmarking of Triton, TensorRT-LLM, and vLLM with Llama 3 and Granite 3.1 on Linux and Kubernetes.

Model	Framework	Platform	Latency (ms)	Throughput (tokens/sec)
Llama-3.1-8B-Instruct	Triton	Linux	50	200
Llama-3.1-8B-Instruct	TensorRT-LLM	Linux	50	200
Llama-3.1-8B-Instruct	vLLM	Linux	50	200
Granite-3.1-8B-Instruct	Triton	Kubernetes	50	200
Granite-3.1-8B-Instruct	TensorRT-LLM	Kubernetes	50	200
Granite-3.1-8B-Instruct	vLLM	Kubernetes	50	200

Table 1: Inference performance comparison.

2 Related Work

Previous studies have examined inference optimizations for transformer-based models. NVIDIA's TensorRT-LLM has been optimized for GPU inference, while Triton provides flexibility across hardware, and vLLM introduces paged attention for improved memory efficiency. However, a direct comparison across these frameworks on the latest models is lacking.

3 Methodology

We evaluate the inference performance based on the following metrics:

- **Latency:** Time taken to generate a response per token.
- **Throughput:** Number of tokens processed per second.
- **Memory Utilization:** GPU memory consumption under different loads.
- **Scalability:** Performance across varying batch sizes and hardware configurations.

3.1 Experimental Setup

- **Hardware:** We use A100 and H100 GPUs for evaluation.
- **Models:** Llama 3 (70B) and Granite 3.1.
- **Software:** TensorRT-LLM vX.Y, Triton Inference Server vX.Y, vLLM vX.Y.

4 Results

4.1 Latency Comparison

We measure token generation latency across different batch sizes. Figure **??** shows the results.

4.2 Throughput Comparison

Table **??** compares the token throughput across frameworks.

4.3 Memory Utilization

We analyze memory usage across different loads.

5 Discussion

The results indicate trade-offs between latency and throughput. TensorRT-LLM excels in optimized GPU performance, Triton provides a flexible deployment solution, and vLLM achieves high throughput through memory-efficient optimizations.

6 Conclusion

Our benchmarking study highlights the strengths and weaknesses of each inference engine for Llama 3 and Granite 3.1. Future work includes extending the evaluation to additional hardware and model architectures.

—new material—