Enabling vLLM on ARM for scalable LLM inference on resource-constrained servers

Sanket Kale, Ashwin Sekhar, Nishant Prabhu, Abhishek Nair, Abhishek Jain, Masahiro Doteguchi, Priyanka Sharma

Fujitsu Limited

Abstract

- vLLM[1] is an LLM serving framework built for deploying large language models (LLMs) with memory efficiency. It provides **continuous batching** and **paged attention** functionality enabling lower memory footprint and the right balance of request-level latency and throughput.
- Originally, vLLM was supported on GPUs and demonstrated 2-4x better throughput compared to other methods for fixed latency requirements. Support for x86 platforms was recently enabled.
- **We enable vLLM for ARM CPUs,** extending support for PyTorch and OpenVINO backends. We develop specialized kernels using Neon and Scalable Vector Extension (SVE)[3] intrinsics for SIMD vectorization on ARM. We observe 1.5x overall latency improvement with PyTorch, ~51x improvement in prefill latency and ~3x improvement in per-token decoding latency with OpenVINO over the baseline.

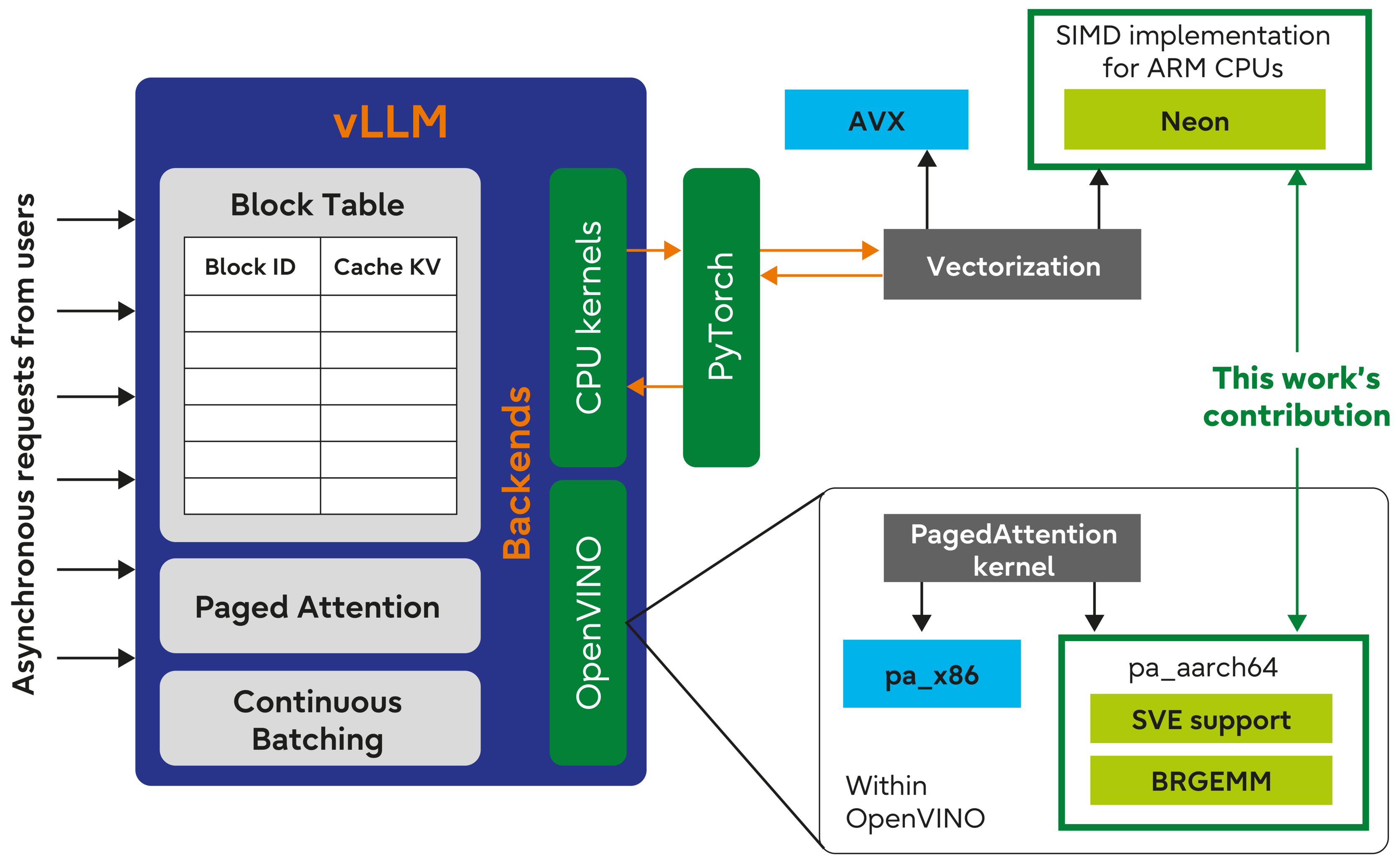

Figure 1: Schematic of vLLM's backend and bindings with PyTorch and OpenVINO

vLLM with PyTorch backend

- Added vectorized implementations of CPU kernels (`PagedAttention, LayerNormalization, PositionalEncoding,` etc.) using Neon intrinsics for FP32, FP16 and BF16 data types.
- Specialized execution pathways were added for hardware without BF16 support (e.g., Apple Metal CPUs) to bypass BF16 execution and use FP32 or FP16 precisions instead.

vLLM with OpenVINO backend

- Utilizes BRGEMM[2] matrix multiplication micro-kernel with threading and blocking logic implemented over them to achieve improved TTFT latency.
- Added implementations of paged attention executor and supporting functions using SVE intrinsics in OpenVINO. This enabled vLLM to use the OpenVINO backend for `PagedAttention` execution on ARM.

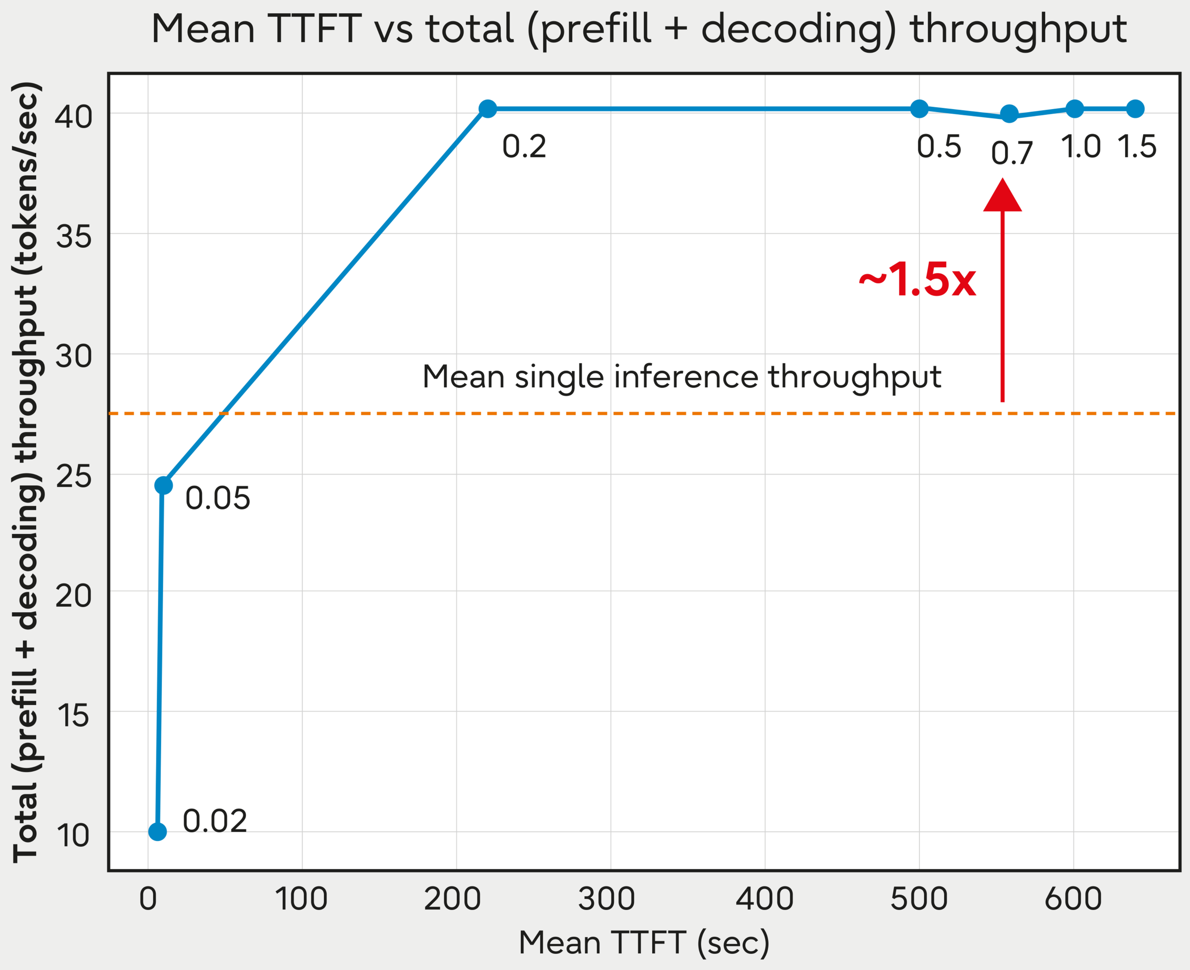

Figure 2

Variation of total throughput (prefill + decoding) with mean TTFT for LLaMA-2 7B model.

Fujitsu's optimizations provide **~1.5x** better throughput at higher request rates (>0.5) compared to naïve inference (PyTorch).

- Annotations are request rates.

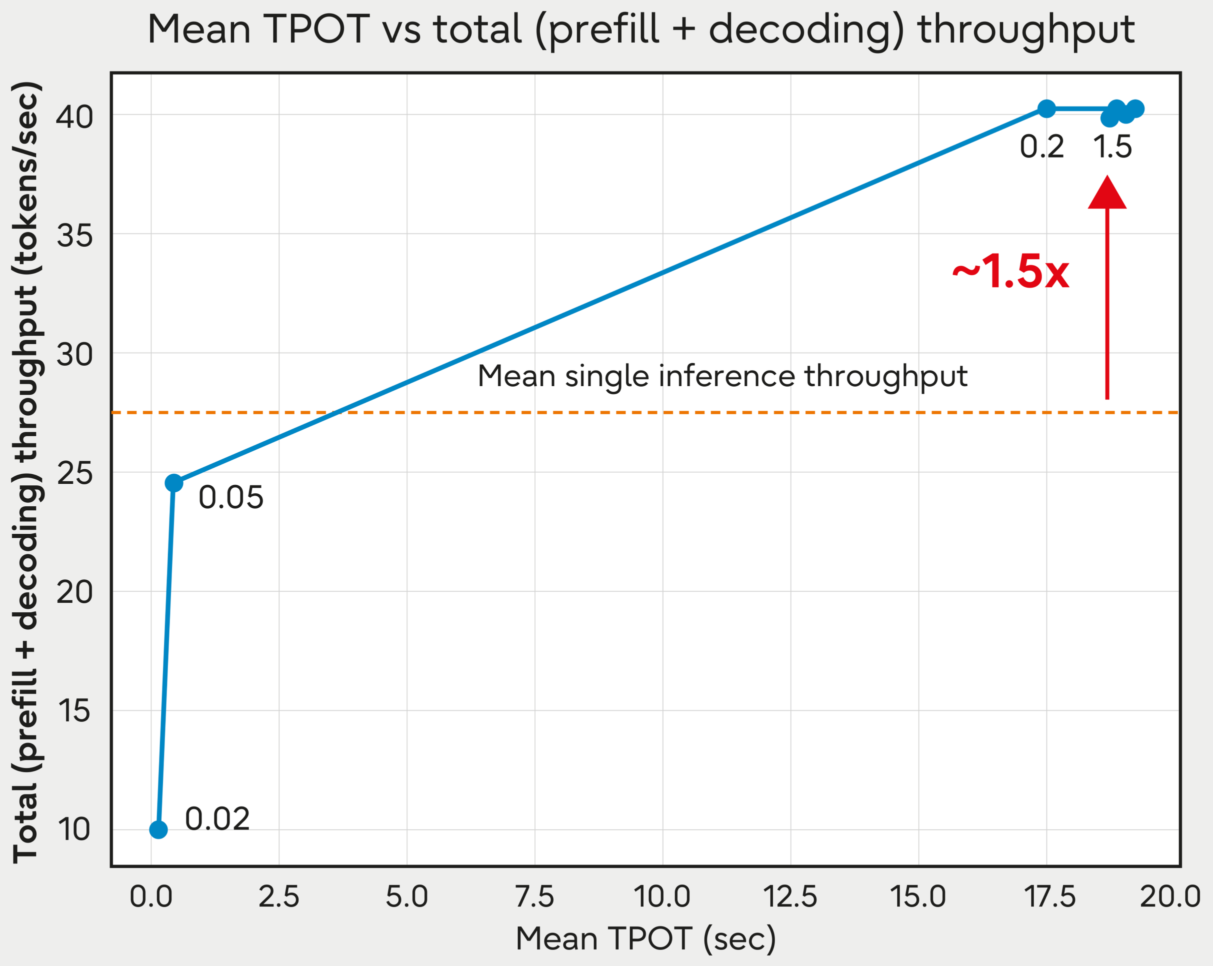

Figure 3

Variation of total throughput (prefill + decoding) with mean TPOT for LLaMA-2 7B model.

Fujitsu's optimizations provide **~1.5x** better throughput at higher request rates (>0.5) compared to naïve inference (PyTorch).

- Annotations are request rates.

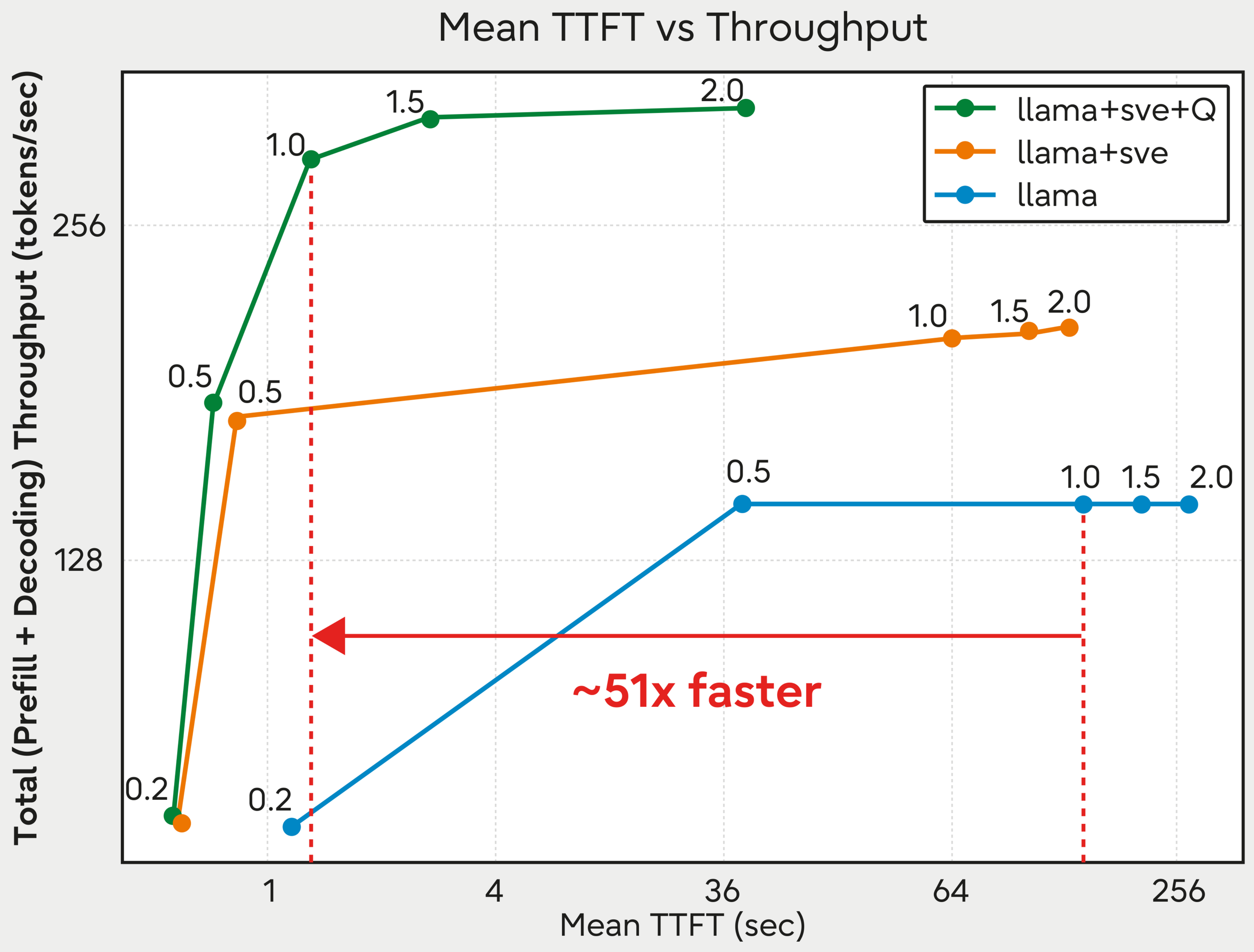

Figure 4

Variation of total throughput (prefill + decoding) with mean TTFT for LLaMA-2 7B.

Fujitsu's optimizations give **~51x** better TTFT on an average for the same request rate.

- **Q**: 8-bit KV-cache quantization.
- Annotations are request rates.

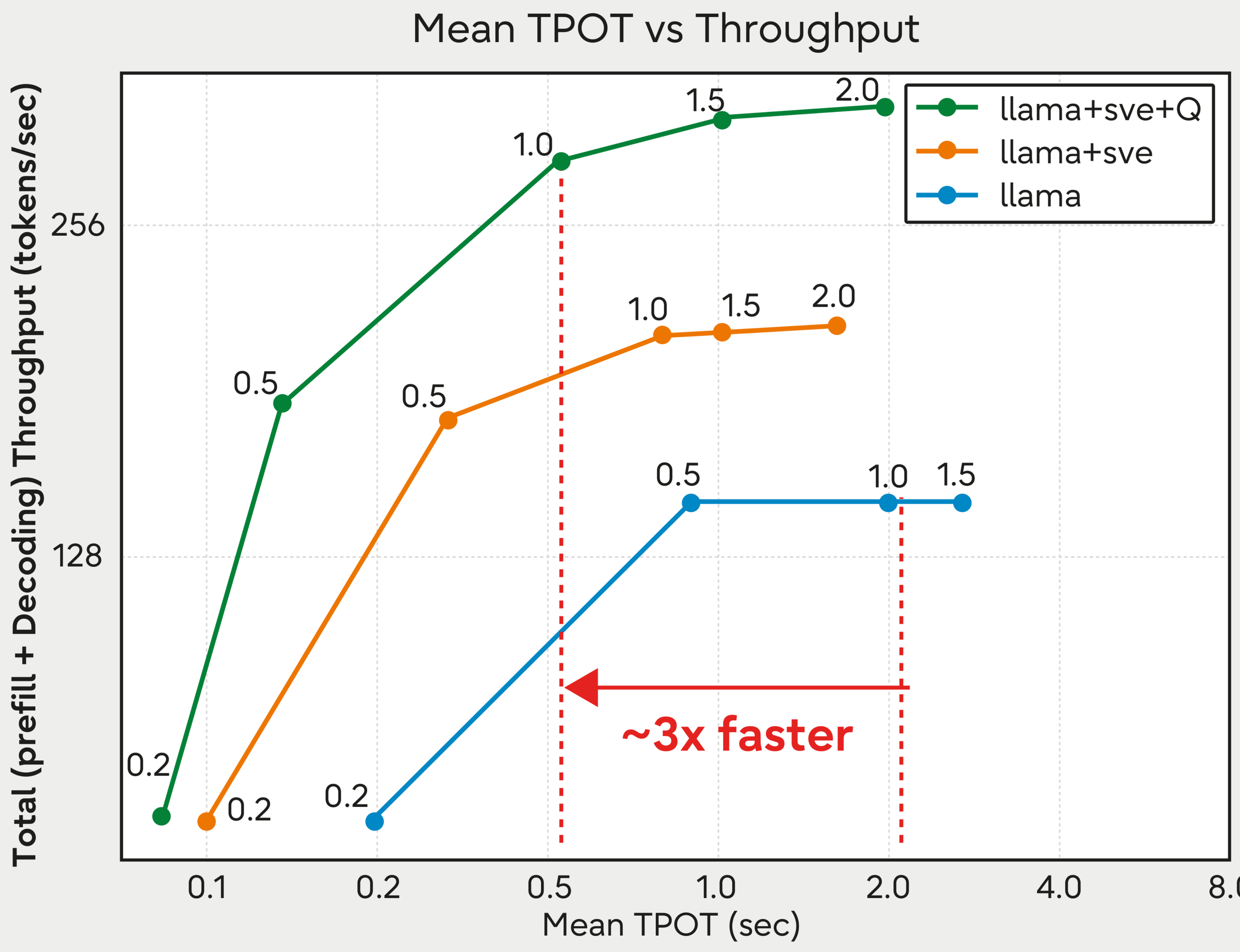

Figure 5

Variation of total throughput (prefill + decoding) with mean TPOT for LLaMA-2 7B.

Fujitsu's optimizations provide **~3x** better TPOT on an average for the same request rate.

- **Q**: 8-bit KV-cache quantization.
- Annotations are request rates.

All measurements performed on AWS Graviton3E (arm64, 64 cores, 128GB RAM, 2.6 GHz)

TTFT – Time to first token (prefill latency), **TPOT** – Time per output token (per-token decoding latency)

Conclusions and Future Work

- This work enables vLLM on the ARM CPUs, with kernel-level Neon and SVE optimizations for PyTorch and OpenVINO backends, giving us strong performance on ARM CPUs.
- We will extend support for `PagedAttention` in FP16 precision in both backends by **July 2025** to enhance performance on Arm CPUs further.

References

[1] Kwon et al., Efficient Memory Management for Large Language Model Serving with PagedAttention. 2023. arXiv:2309.06180.

[2] Georganas et al., High-Performance Deep Learning via a Single Building Block. 2019. arXiv:1906.06440.

[3] Scalable Vector Extensions, Arm Developer, https://developer.arm.com/Architectures/Scalable%20Vector%20Extensions

Links to pull requests

vLLM + PyTorch

vLLM + OpenVINO

This poster is based on results obtained from a project, JPNP21029 subsidized by the New energy and Industrial Technology Development Organization (NEDO).