Harshith Kantamneni

kantamneniharshith@gmail.com | +1 (414) 916-5799 | Milwaukee, WI (open to relocation)
linkedin.com/in/hk4231 | github.com/harshithkantamneni | harshithkantamneni.github.io

SUMMARY

LLM inference engineer (M.S. ECE, UW-Madison, Dec 2025) focused on decode: **KV cache**, **speculative decoding**, **MoE** from host memory and **vLLM** measurement. 3 PRs merged in NVIDIA cuDNN frontend and flash-linear-attention.

OPEN-SOURCE EXPERIENCE

GPU Kernel Contributor Mar 2026 - Present

- **NVIDIA cuDNN frontend, tests merged** (#1287). Prototyped native grouped-query attention (GQA) on A100 (#1223), removing K/V head copies that outlasted the kernel at 64/8 heads (7.28 vs 6.0-6.7 ms). At a 128K **KV cache**, multi-token steps ran 1.15-2.19x faster and 6 batched shapes **2.6-3.3x**. NVIDIA then shipped GQA (#1239) and merged my tests.
- **vLLM RFC, measurement comment** (#46295). Measured **speculative decoding** on MoE on H200, counting distinct experts per verify step. With CPU ngram drafts, a per-workload oracle improved on the best static batch-size table by at most **1.8%**, and vLLM's MFU byte model overcounted expert reads up to 3x.
- **flash-linear-attention, merged** (#1412). Added **KV-cache decoding**, tested against the uncached forward. Flagged where it slows at batch 1 and 64K keys: 32 programs on 132 SMs take **4.3 ms** while K/V reads take about 0.06 ms.
- **flash-linear-attention, merged** (#1319). Continued #599 to add stick-breaking attention as **Triton** kernels. Rewrote the backward to fix a silent fp16 zero gradient caught in review and retiled the forward for 1.29-1.48x the PR's previous head.
- **vLLM, not merged** (#38338). Built a **Triton NVFP4 GEMM** reference. Benchmarked against **CUTLASS** on RTX 5090 (Llama QKV, hot L2): 1.38-3.04x up to batch 128, 0.77-0.91x above.

PROJECTS

Where the Seconds Go: MoE Decode from Host Memory (GitHub) Sep 2026 - Present

llama.cpp, C++, CUDA, Python, Mixture-of-Experts (MoE), on RTX 5090, 4090 and 3090. Manuscript and code public.

- Built a **llama.cpp expert cache** for batch-1 MoE decode when the model doesn't fit on the GPU, with per-layer GPU slots, a decayed-frequency admission rule, GPU-signaled CPU helpers and a per-machine PCIe copy table.
- **Benchmarked it** on RTX 5090, 4090 and 3090 hosts. On RTX 5090s, gpt-oss-120b ran at **58-153 tokens/s** with 11-40% of its experts on the GPU. Across 12 configurations of 2 models, it was faster than FreeToken in 11, by up to 1.29x, and **2.0-4.0x** faster than stock llama.cpp in all 12.
- Computed each machine's decode speed limit with **Belady's MIN** on real routing traces and probed read rates. None of 302 timed runs exceeded it, and at gpt-oss's smallest cache on 25 consumer machines the cache reached 31-54% of it.

vLLM Serving Benchmark (GitHub) Jun 2026

vLLM, Qwen2.5-7B-Instruct, Python, Poisson arrivals, on H100 NVL

- Load-tested **vLLM** across offered rates. **p99 TTFT** stayed at 45-58 ms up to 16 req/s and 102 ms at 32 req/s. An unbounded burst pushed first-token latency to 1,970 ms while **TPOT** rose only 2.3x.
- Measured TPOT **31-34% lower** with **CUDA graphs** than eager mode, about 3 ms per token. **FP8** cut it 27-36% and raised peak throughput 25%.

Triton vs cuBLAS LLM Kernel Benchmarking (GitHub) Jan 2026 - Feb 2026

Triton, CUDA, PyTorch, on A100

- Built **Triton** GEMM kernels, fused linear, bias and GELU, and benchmarked 76 LLM shapes against **cuBLAS**. In fp16, fusion raised median speed from 0.79x to 0.96x of cuBLAS and reached **1.73x** on one small-batch shape.

SKILLS

Languages and tools: Python, C++, CUDA, Triton, PyTorch, Linux, Git, Docker
Inference: vLLM, llama.cpp, model serving, load tests, KV cache, speculative decoding, MoE, FP8, NVFP4, quantization
Kernels: GPU architecture, vLLM and SGLang MoE kernels, CUTLASS, kernel fusion, tiling, Tensor Cores, CUDA graphs
Performance analysis: TTFT, TPOT, tail latency, throughput, profiling, Nsight Compute, roofline analysis
Hardware: NVIDIA B200, H200, GH200, H100, A100, RTX 5090

EDUCATION

University of Wisconsin-Madison, M.S. Electrical and Computer Engineering Sep 2024 - Dec 2025

Coursework: High Performance Computing, Advanced Computer Architecture, Machine Learning, Fault-Tolerant Computing

Vellore Institute of Technology, B.Tech Electronics and Communication Engineering 2020 - 2024