Harshith Kantamneni

kantamneniharshith@gmail.com | +1 (414) 916-5799 | linkedin.com/in/hk4231 | github.com/Drogon4231
Milwaukee, WI

M.S. ECE student specializing in CUDA Kernel Optimization, GPU Memory Systems, and Performance Engineering

Skills

- **Languages:** C, C++, Python, CUDA
- **Parallel Programming:** Triton, OpenMP
- **Tools & Infrastructure:** PyTorch, Slurm, gem5, McPAT, Git
- **Domains:** GPU Kernel Optimization, GPU Memory Systems, High-Performance Computing, Computer Architecture, ML for Systems

Education

University of Wisconsin–Madison *Madison, USA*
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 *Amaravati, India*
B.Tech in Electronics and Communication Engineering 2020 – 2024

Projects

GPU Kernel Performance Benchmarking (CUDA & Triton) Dec 2025 – Present
C++, CUDA, Triton, Python

- Implemented and optimized GPU kernels for **GEMM** and **parallel reductions** using both **CUDA** and **Triton** kernels.
- Applied shared-memory tiling, block-level parallelism, and launch configuration tuning to study performance trade-offs across implementations.
- Benchmarked kernel throughput and effective memory bandwidth using **CUDA event timing** across varied configurations.
- Analyzed sensitivity to tile shape, block size, and memory access patterns to identify occupancy and bandwidth bottlenecks.
- Built a reusable benchmarking harness to enable fair, repeatable performance comparisons between CUDA and Triton kernel variants.

ML-Guided CUDA Kernel Configuration Optimizer Jan 2025 – May 2025
Python, PyTorch, CUDA, Slurm GitHub

- Trained a learning-based model to predict near-optimal **CUDA grid and block sizes** from workload characteristics.
- Reduced exhaustive kernel tuning overhead by $>$**95%**, achieving up to **30% runtime improvement** on matrix workloads.
- Automated large-scale kernel benchmarking using Slurm pipelines and CUDA event-based timing.

MI300X Memory-System Reverse Engineering & gem5 Calibration Sep 2025 – Dec 2025
CUDA, gem5, Python

- Developed CUDA microbenchmarks (pointer-chasing, stride sweeps) to characterize **L2/L3/HBM latency and bandwidth** behavior.
- Analyzed access patterns and timing behavior to infer chiplet-level cache organization and memory hierarchy effects.
- Calibrated gem5 MI300X memory parameters (cache size, line size, latency) to better align simulation with measured hardware trends.
- Established a reproducible methodology for studying modern GPU memory systems using microbenchmarks and simulation.

ML-Assisted Task Graph Partitioning Jan 2025 – May 2025
Python, XGBoost, Slurm GitHub

- Modeled task-graph structural features to predict runtime-efficient GPU partitioning strategies.
- Achieved $<$**5% prediction error**, reducing design-space exploration time by **25%**.
- Enabled faster performance evaluation by replacing exhaustive sweeps with learned configuration selection.

ABFT-GEMM Reliability and Performance Analysis Sep 2025 – Dec 2025
Python, Slurm

- Performed Monte-Carlo fault injection on GEMM workloads to study numerical robustness under soft-error conditions.
- Measured performance and accuracy trade-offs across matrix sizes using batch execution on HPC clusters.
- Analyzed resilience overheads in compute-intensive kernels relevant to large-scale GPU workloads.