Vijay Prabhas Kodamalla

Atlanta, GA
470-830-4487 vkodamalla3@gatech.edu linkedin.com/in/vijaykodamalla github.com/vkodamalla3

Education

Georgia Institute of Technology **Aug. 2024 – May 2026**
M.S. in Computational Science and Engineering *Atlanta, GA*

Indian Institute of Technology, Dharwad **Aug. 2020 – May 2024**
B.Tech. in Mechanical, Materials and Aerospace Engineering *Dharwad, India*

Technical Skills

Languages: C/C++, Fortran (F90), CUDA C/C++, CUDA Fortran, Python
Parallel and Runtime: MPI (Open MPI/HPC-X), CUDA-aware MPI, NCCL, UCX, OpenACC, NVHPC
ML/DL: PyTorch, TensorFlow
Profiling/Debugging: NVIDIA Nsight Systems, Nsight Compute, TensorBoard
Systems Tools: Linux, Git, Slurm

Experience

Georgia Institute of Technology — Computational Combustion Lab (CCL) **Dec. 2024 – Present**
Graduate Research Assistant *Atlanta, GA*

- Focused on **GPU optimization** of **LESLIE** (finite-volume multi-physics CFD solver; Fortran90 + MPI) using **OpenACC + NVHPC**. Guided changes via profiling with **Nsight Systems/Compute**.
- Achieved ∼**38**× speedup on **A100** vs **60 CPU cores** (Xeon Gold 6338) on a **9.53M** cell case; improved **SM busy** >54%, reduced **warp long-scoreboard** stalls by 93%, reached **57% of FP64 peak** in the *ConsToPrim* kernel.
- Kernel-level: loop unrolling and fusion, manual inlining to amortize call overhead, register reuse. Instruction-level: DFMA, branchless `MERGE` to mitigate divergence, prefetch/caching tuned from PTX/SASS inspection.
- Integrated and evaluated **communication runtimes** (CUDA-aware MPI, NCCL, UCX) across **NVLink, PCIe, and InfiniBand** to minimize latency and staging overhead.
- Stack: **NVHPC 24.5**, OpenACC, MPI (HPC-X), Nsight Systems/Compute, Slurm; experiments on Georgia Tech PACE.
- **Research Poster submitted:** GPU Acceleration and Optimizations of LESLIE — SC25 (St. Louis).

Selected Projects

Multi-GPU Communication Benchmarks (AXPY) | *C++/CUDA, MPI/HPC-X, NCCL, UCX* **Jun. 2025 – Present**

- Developed multi-GPU communication benchmarks in **C++ with embedded CUDA**, implementing progressive levels: L1 baseline MPI; L2 CUDA-aware MPI (host staging removed); L3A CUDA streams; L3B NCCL collectives; L3C tree-based reduction in shared memory + MPI; L3D CUDA-IPC/`cudaMemcpyPeer` intra-node; L4A UCX data path with MPI bootstrap.
- On **256M** elements, measured comm latencies: L1 **45 ms** → L2 **23.209 ms**; L3A **19.807 ms** (8 streams); L3C **5.283 ms** (**4.4**× over L2); L3D **1.267 ms** (**18.3**× over L2).
- Benchmarked UCX's **transport selection** (InfiniBand, NVLink, PCIe) and analyzed latency/throughput tradeoffs for collective ops. Instrumented with Nsight Systems.

3D CNN with Non-Local Self-Attention (UCF-101 subset) | *TensorFlow 2.x, CUDA* **Jun. 2025 – Present**

- Designed a strong 3D CNN baseline and integrated a **Non-Local** attention block; studied placement: early vs late. Late placement yielded **0.8357** accuracy (16-class subset); early stacking degraded to **0.284**.
- Profiled training on **H100**; identified **HtoD copies** as a 65% bottleneck and many small ops (e.g., BN) limiting utilization; planning a fused **custom TensorFlow op** (C++/CUDA) for the attention block.

Relevant Coursework

- GPU Hardware Software
- Domain-Specific Languages (HPC)
- High-Performance Computer Architecture
- Algorithms
- Computer Vision