

Vivek L. Kale

Phone: (650) 733-9327 | **Email:** vivek.lkale@gmail.com | **LinkedIn:** linkedin.com/in/vlkale
Github: github.com/vlkale | **Website:** vlkale.github.io | **US Citizen with Secret Clearance**

Professional Summary

- Software engineer and technical leader specializing in GPU-based cluster computing and communication for high-performance science and AI infrastructure.
- Proven expertise in developing software for multi-core and GPU compiler optimization and multi-GPU communication, with demonstrated leadership in managing technical teams, driving open-source initiatives, and delivering production-grade parallel programming tools and runtimes.
- Strong track record of technical management and cross-functional collaboration for production-level software projects, multi-million dollar research initiatives, publications, and interaction with hardware vendors and research institutions.

Relevant Experience

Technical Leadership and Industry-grade Open-source HPC Software

Sandia National Laboratories

Principal Member of Technical Staff II

July 2024 - Present

- Software design and engineering for tools for distributed Kokkos, specifically (1) inter-process profiling and adaptivity via PMPI and (2) job-level monitoring and feedback via LDMS.
- Investigating MPICH features for runtime auto-tuned multi-GPU collective communication via RDMA using LDMS feedback and its performance impact to a MPI+Kokkos science application.
- Released a Spack package for Kokkos Tools for distributed profiling, having 4 configurations each for MPICH and OpenMPI implementations, resulting in 16 new users of Kokkos Tools.
- Developed AI-assisted HPC Tools through LLMs (coderosetta.com) and autotuning (TAU+APEX) for Kokkos applications run on NVIDIA GPUs, resulting in a poster presentation at GTC 2025.
- Research and pathfinding on the use of AI chips, e.g., Cerebras WSE-3, for science simulations.
- Submitted two proposals on correctness tools for HPC, each with \$1.5M in funding for 3 years.

Senior Member of Technical Staff

August 2022 - July 2024

- Developed and maintained Kokkos Tools for the CMake and Spack build system, tooling overheads, CI/CD, auto-tuning, and nvtx/roctx/vtune integration, leading to 15 merged github PRs.
- Developed a debugging tool that detected 7 common Kokkos user bugs by analyzing LLVM IR of Kokkos programs via symbolic execution, leading to a paper at SC24's Correctness workshop.
- Implemented 5 new loop transformation features in LLVM OpenMP, leading to a 1.7x speedup for a Kokkos-OpenMP+CUDA benchmark using the index set split construct, 3 accepted OpenMP 6.0 features, and 11 feature proposals in OpenMP 7.0.
- Implemented multi-GPU reduction operation and broadcast operation via USM and RDMA in a prototype library for LLVM OpenMP, leading to a 1.3x speedup over the corresponding MPICH multi-GPU collective, and 9 OpenMP feature proposals for OpenMP 7.0.

Brookhaven National Laboratory Assistant Computational Scientist *May 2019 - August 2022*

- Implemented OpenMP user-defined multi-GPU scheduling for LLVM, offering 2.1x speedup over using MPI parallelization, leading to papers at IWOMP 2020 and BCB 2021.
- Implemented performance optimizations in LLVM for OpenMP asynchronous GPU offloading that achieved a 1.2x speedup, leading to a paper at SC22's HiPar workshop.
- Developed performance benchmarks that evaluated 5 major vendor OpenMP GPU implementations, leading to an ACM journal paper and an IWOMP 2021 workshop paper.
- Demonstrated technical leadership as technical project manager for the ECP SOLLVE project, submitting 12 ECP milestone reports, organizing 7 GPU hackathons, and defining 3 project KPIs.

HPC Software Development and Performance Engineering

USC/ISI + Charmworks, Inc. Software Engineer *Dec 2015 - May 2019*

- Implemented User-defined Loop Schedules (UDS) for OpenMP and RAJA via a prototype library for LLVM and GCC, leading to a paper at IWOMP 2018 and 3 github PRs merged in Charm++.
- Performance analysis and optimization of MPI+CUDA scientific applications on NVIDIA GPUs via CUPTI and auto-tuning, leading to 1.4x speedup of an application for computer chip design.
- Developed novel and efficient multi-level loop schedulers in Charm++, leading to a 1.2x speedup on the PRK particle-in-cell benchmark code and a Best Poster Candidate at SC18.

LLNL + UIUC Researcher *Jun 2010 – Dec 2015*

- Implemented a ROSE-based compiler pass and PMPI-based runtime system for MPI+OpenMP applications to use loop scheduling techniques, leading to a 1.4x speedup on a multicore cluster.
- Implemented shared memory extensions for MPICH, leading to a paper with 140+ citations.
- Implemented multicore and GPU performance optimizations for domains of linear algebra, blood flow, fusion, and combustion, leading to 2 papers at IPDPS.

General Software Development

Proteus Technologies + Wolfram Software Developer *Aug 2007 – September 2008*

- Developed and tested service-oriented software to monitor the health of a large-scale distributed system for the US government, leading to an internal white paper and software package.
- Implemented functionality in Mathematica for users to send emails from within a Mathematica evaluation kernel, via sendmail and TLS, leading to a new software feature in Mathematica.

Technical Skills

Languages: C, C++, Python, Fortran, Java, bash, csh, VHDL, Matlab;

Libraries: OpenMP and OpenACC (GCC, LLVM), CUDA, HIP, Kokkos, MPI (MPICH), Charm++, MLIR, Kokkos Kernels, LAMMPS, numpy, pandas, scikitlearn, PyTorch;

Tools: Kokkos Tools, PMPI, ompt, PAPI, nvtx, roctx, NVIDIA Nsight Systems+Compute, HPCToolkit, CUPTI, tau, VTune, clang-tidy, KLEE, gprof, nvprof, gdb, slurm;

Utilities: Cursor AI, Claude Code, git, CMake, Spack, vi, clang-format, gnuplot, emacs, autoconf, LaTeX, docker, matplotlib;

Education

- Certification, Technical Management Program, 2024, University of California at Los Angeles
- Ph.D (Doctor of Philosophy), Computer Science, 2015, University of Illinois at Urbana-Champaign
- B.S. (Bachelor of Science), Computer Science, 2007, University of Illinois at Urbana-Champaign

Open-source Software Projects

1. **OpenMP Locality-aware Loop Scheduling:** <https://github.com/vlkale/lw-sched>
2. **Multi-GPU Stencil Benchmark:** <https://github.com/vlkale/ParallelProgrammingWithOpenACC>
3. **MPI Slack Predictor:** <https://github.com/vlkale/slack-trace>