

## 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

- Highly skilled computational scientist and software developer with expertise in high performance computing (HPC), runtime systems, and parallel programming models for GPU-based clusters.
- Proven track record of contributions to parallel programming standards, open-source software for AI-assisted HPC tools for profiling and debugging, and research on adaptive load balancing.
- Effective communicator and collaborator with a strong record of publications and software projects.

### 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.
- 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 new features in LLVM OpenMP, leading to a 1.2x speedup for a Kokkos-OpenMP+CUDA benchmark, 3 OpenMP 6.0 features, and 19 feature proposals for OpenMP 6.1.

**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++, CUDA, python, Fortran, Java, bash, csh, VHDL, Matlab;

**Libraries:** OpenMP and OpenACC (GCC, LLVM), Kokkos, MPI (MPICH), Charm++;

**Tools:** Kokkos Tools, PMPI, ompt, PAPI, nvtx, NVIDIA Nsight, tau, hpcToolkit, VTune, clang-tidy, KLEE, gprof, nvprof, CUPTI, gdb, pandas, numpy, scikitlearn;

**Utilities:** 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>