Vivek L. Kale

Phone: 217-369-7996 | **Email**: vivek.lkale@gmail.com | **LinkedIn**: linkedin.com/in/vlkale
Github: http://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 tools for profiling and debugging, and research on adaptive runtime systems.
- Effective communicator and collaborator with a strong record of publications and software projects.

Relevant Experience

Sandia National Laboratories
Principal Member of Technical Staff II *July 2024 - Present*

- Technical lead for HPC Tools and Runtime Systems R&D for Sandia's Exascale Software.
- Pathfinding and software engineering for tools for Kokkos integrated with (1) HPC performance monitoring (LDMS) and feedback and (2) PMPI and adaptive runtime systems for MPI.
- Developing AI-assisted HPC Tools through coderosetta.com for Kokkos-CUDA and APEX Kokkos Tools autotuning, resulting in an NVIDIA GTC 2025 poster.

Senior Member of Technical Staff *August 2022 - June 2024*

- Code development and maintainance of Kokkos Tools by improving build system, tooling overheads, CI, auto-tuning, nvtx/roctx/vtune integration, with 15 github PRs merged.
- Developed an LLVM-based SymEx tool to debug Kokkos programs that detected 7 Kokkos bug examples with no false positives, leading to a paper at SC24's Correctness workshop.
- Implemented prototype LLVM OpenMP loop transformation directive 'split', leading to a 1.24x speedup for a OpenMP + CUDA benchmark and OpenMP 6.0's split directive.
- Pathfinding for OpenMP multi-GPU features for NVIDIA DGX, as well as AWS, Google Cloud and OCI, leading to 19 proposed features for OpenMP 6.1/7.0.

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

- Implemented OpenMP user-defined multiGPU scheduling for LLVM, offering 2x speedup over using an MPI parallelization, leading to two papers at IWOMP 24.
- Implemented performance optimizations in LLVM for OpenMP asynchronous GPU offloading that showed 1.23x speedup, leading to a paper at SC22's HiPar.
- Developed performance benchmarks that evaluated 5 major vendor OpenMP GPU implementations, leading to a journal paper and two workshop papers.
- Demonstrated technical leadership as technical Project Manager for ECP SOLLVE project, submitting 9 ECP milestones, organizing 6 OpenMP GPU hackathons, and defining 2 project KPIs.

Charmworks and ISI *Software Engineer* *Jan 2016 - April 2019*

- Implemented and researched LLVM OpenMP User-defined Loop Schedules (UDS).
- Added a UDS feature to Charm++'s CkLoop, with one PR merged.
- Analysis and optimization of science simulations on NVIDIA GPUs via CUPTI and auto-tuning.

University of Illinois *Postdoctoral Associate* *Jul 2015 - Dec 2015*

- Exhibited problem solving: made MPI+OpenMP+OpenACC science simulation code 1.24x faster.

Education

- Ph.D (Doctor of Philosophy), Computer Science, 2015, University of Illinois at Urbana-Champaign
  Dissertation: *Low-Overhead Scheduling to Improve Performance of Scientific Applications*
- B.S. (Bachelor of Science), Computer Science, 2007, University of Illinois at Urbana-Champaign

Technical Skills

Languages: C, C++, CUDA, python, Fortran, Java, bash, csh, VHDL, Matlab;
Programming Models: OpenMP (gomp, llvm), Kokkos, MPI (mpich), Charm++, OpenACC (pgi), Globus, mpi4py, pyomp;
Math: cuBLAS, MAGMA, PetSC
Data Analysis: matplotlib, pandas, R, ; **Tools**: Kokkos Tools, PMPI, ompt, PAPI, nvtx, NVIDIA Nsight, tau, hpcToolkit, VTune, clang-tidy, KLEE, gprof, gdb;
Utilities: git, cmake, spack, vi, clang-format, gnuplot, emacs, autoconf, LaTeX, docker;

Open-source Software Projects

1. **OpenMP multi-GPU support**: User-defined multi-GPU loop scheduling for clang/LLVM OpenMP.
   Repo: `https://github.com/sollve/openmp-rts`
2. **Kokkos Tools**: Kokkos Tools and runtime systems for C++.
   Code Repo: `https://github.com/kokkos/kokkos-tools`
3. **MPI Slack Predictor**: MPI runtime tool using libunwind to predict slack trace
   Code Repo: `https://github.com/vlkale/slack-trace`

Hobbies

- Piano: I played classical music for 8 years in high school and am currently practicing pop music.
- Tennis: I played competitively in high school and I now play relationally in a club.