Vivek L. Kale, PhD

Phone: +01 217-369-7996 . *Email*: vivek.lkale@gmail.com . *Web*: `http://vlkale.github.io`
U.S. Citizen

Education

Bachelor of Science, Computer Science
University of Illinois at Urbana-Champaign, May 2007

Doctor of Philosophy, Computer Science
University of Illinois at Urbana-Champaign, May 2015
Advisor: William D. Gropp

Courses Taken

Designing and Building Scientific Applications, Parallel Software Patterns, Program Optimization, Parallel Computer Architecture, Algorithms, Artificial Intelligence, Software Engineering

Technical Skills

Languages: C, C++, python, Fortran, bash, csh, CUDA

Tools: LaTeX, gnuplot, emacs, autoconf, cmake, svn

Libraries for Parallelism: POSIX threads (Pthreads), MPI (mpich3), OpenMP (gomp, llvm), OpenACC (pgi)

Performance Profiling Tools and Libraries: hpcToolkit, PAPI, nvprof, gprof

Honors

- SC 2017 Early Career Program Invitee and Participant
- 2015 Heidelberg Laureate Forum Invitee and Participant.
- Fellow at Lawrence Livermore National Laboratory through Lawrence Scholar Program.

Experience

Brookhaven National Laboratory Computer Scientist May 2019 - present

- Contributing to developing and ensuring quality of an LLVM OpenMP implementation, specifically the LLVM OpenMP implementation's compiler and its runtime, targeted for Department of Energy's (DoE) upcoming exascale supercomputers through DoE's SOLLVE project.
- Designing, prototyping, and implementing OpenMP adaptive and locality-aware loop scheduling strategies to improve intra-node load balancing for MPI+OpenMP applications running on supercomputers having heterogeneous nodes (i.e., nodes containing sets of a variety of processors, e.g., CPUs, GPUs) for LLVM's OpenMP implementation.
- Proposing OpenMP directives in the upcoming OpenMP 6.0 Specification for OpenMP parallelization and scheduling of applications on heterogeneous processors of a node for C, C++ and Fortran, utilizing user-defined schedules in OpenMP.

- Contributing to LLVM OpenMP loop transformations in the OpenMP 5.1 and 6.0 Specification, specifically tiling, unrolling, loop fission, and loop index splitting.
- Contributing to features in the BOLT runtime system for interoperability of MPI and OpenMP.
- Developing benchmarks for evaluating quality, particularly performance, of OpenMP implementations, e.g., LLVM's OpenMP, NVIDIA's OpenMP, on DoE exascale supercomputers.
- Maintainer of the Spack software package for DoE's fork of LLVM, and leading efforts for Continuous Integration of SOLLVE software within DoE's exascale software stack.
- Engaging with novel and timely applications, particularly machine learning, scientific workflows, and bio-molecular docking for COVID-19 therapeutics, to use HPC optimization strategies and in particular OpenMP optimization strategies.
- Leading hackathons (including virtual) for using OpenMP and its features in existing and new applications on DoE's exascale supercomputers, and identifying problems in performance and correctness of OpenMP's usage in HPC applications through profiling and debugging tools.
- Helped with proposals for research in HPC to DoE and National Science Foundation.
- Served as Technical Program Manager for DoE Exascale Computing Project's (ECP's) SOLLVE project, and serving as Brookhaven National Laboratory Representative in the OpenMP Architecture Review Board for the OpenMP Specification.

Charmworks, Inc. Research Software Developer June 2018 - April 2019

- Incorporated adaptivity to locality-sensitive loop scheduling added in CkLoop, using timings from previous invocations of loop to adjust parameters of scheduling.
- Tuning and further experimentation of strategy of synergizing CkLoop's inter-node loop scheduling and Charm++'s intra-node load balancing.
- Helped to integrate CkLoop with locality-sensitive scheduling strategies into Charm++ v. 6.
- Added user-defined, adaptive and locality-sensitive OpenMP loop schedules in a fork of the C++ performance portability library RAJA, and experimented with combination of RAJA with loop scheduling strategies and Charm++ load balancing.
- Made prototype library supporting user-defined and locality-sensitive loop schedules in OpenMP portable across different parallel programming libraries and languages, done with Oak Ridge National Lab through DoE Exascale Computing Program (ECP).
- Presented and obtained feedback on proposal on a user-defined loop schedule in C, C++ and Fortran to add in OpenMP 6.0 at the OpenMP forum meetings in January and May 2018.
- Assisted with slides for pitch and marketing of Charm++ software and helped to improve tutorials on Charm++, particularly with respect to its discussion on intra-node loop-level parallelism and inter-node load balancing.

University of Southern California Computer Scientist December 2016 - June 2018

- Worked in team to manage computational performance aspects of running an MPI+CUDA application program involving 3-D image reconstruction on a cluster of nodes having GPUs, in particular DoE supercomputers and USC ISI's private supercomputer.
- Identified performance bottlenecks of X-ray tomography application run on NVIDIA GPUs through NVIDIA's profiling tools, e.g., nvprof, and HPCToolkit.
- Worked on transformation of 3-D image reconstruction application code's computational kernels from CUDA to OpenACC.
- Implemented strategies that synergize CkLoop's intra-node locality-sensitive loop scheduling and Charm++'s inter-node load balancing to improve performance of scientific applications.
- Added support for locality-sensitive loop schedules in LLVM's OpenMP implementation.
- Designed and implemented MPI+OpenMP load imbalanced stencil benchmark.

- Developed performance optimizations for locality-sensitive loop scheduling in OpenMP for use on desktop computers or laptops to improve energy efficiency.
- Developed ideas for an OpenMP User-defined loop schedule, based on locality-sensitive loop scheduling strategies in OpenMP, in OpenMPCon 2017 extended abstract.
- Added examples of loop scheduling in OpenMP and source-level optimizations with it, particularly to improve data locality, in the Examples section of OpenMP Specification.
- Translated an x-ray tomography code written in Matlab code to C code and then parallelized it to run on USC ISI's private supercomputer having nodes with GPUs.
- Using the Globus Toolkit and considering DoE's ESNet, ensured external network infrastructure for transferring the application's input data files were adequate for the application's efficiency on USC ISI's private supercomputer.
- Collaborated on discussions and research of use of FPGAs as accelerators on supercomputers.

Charmworks, Inc. Research Software Developer Jan. 2016 - Nov. 2016

- Implemented and experimented with mixed static/dynamic loop scheduling strategies within Charm++'s loop parallelism and scheduling library CkLoop.
- Developed CkLoop microbenchmarks for experimenting with loop scheduling strategies implemented in CkLoop.
- Experimented with impact of compiler optimization levels and flags of Charm++ + CkLoop programs that use locality-sensitive loop scheduling.
- Developed open-source loop scheduling library that allows application programmers to use strategies from dissertation.
- Helped to improve performance portability of Charm++, particularly CkLoop library of Charm++, to a variety of platforms.
- Assisted with business aspects of a high-tech startup company.

University of Illinois Postdoctoral Associate Jul. 2015 – Dec. 2015

- Adapted a plasma physics application code to work on a NVIDIA GPU and Intel Xeon Phi through converting an MPI Fortran code to an MPI+OpenACC Fortran code.
- Explored within-node domain decompositions used within OpenACC adapted from across-node MPI domain docompositions.
- Assessed impact of loop transformations and of compiler optimization level to MPI+OpenACC version of code.
- Used locality-aware scheduling from dissertation and over-decomposition to improve performance of the MPI+OpenACC application code.

Lawrence Livermore Nat'l Lab Lawrence Scholar Feb. 2012 – Jun. 2014

- Measured MPI communication delays for micro-benchmarks codes run on supercomputers and worked to find tools to measure dequeue overheads of OpenMP loop schedulers.
- Created a software system for automated performance optimization and application programmer usability of low-overhead hybrid scheduling strategies.
- Developed a ROSE-based custom compiler for automatically transforming MPI+OpenMP applications to use low-overhead scheduling techniques and runtime.
- Assessed further opportunities for performance improvement of low-overhead schedulers, including improvement of spatial locality of low-overhead schedulers.

Lawrence Livermore Nat'l Lab Scholar Jun. 2011 - Sep. 2011

- Experimented with different OpenMP parameters of implemented MPI+OpenMP application code to understand performance optimizations on LLNL supercomputers.

- Developed software design for low-overhead loop scheduling library based on libgomp software design.

Lawrence Berkeley Nat'l Lab   Summer Scholar   Aug. 2010 - Sep. 2010

- Analyzed results for the performance tests developed on NERSC machines.
- Compared with collectives in reference to MPI (mpich2) runtime system.

Lawrence Livermore Nat'l Lab   Scholar   May. 2010 - Aug. 2010

- Modified libgomp runtime system in order to integrate low-overhead schedulers within it.
- Developed an algorithm multi-stage low-overhead loop scheduler with each stage associated with a level in the memory hierarchy, allowing for MPI-shared memory extensions to be used in conjunction with the low-overhead loop scheduling strategies.

Goldman-Sachs   Summer Analyst   Jun. '09 – Sep. '09

- Wrote code for testing trading system infrastructure functions under extreme market conditions.
- Analyzed performance bottlenecks of system infrastructure functions.

Proteus Technologies, LLC   Software Developer   Aug. 2007 – Apr. 2008

- Primarily responsible for developing, testing and documenting a service-oriented software application for health and status monitoring of large-scale parallel and distributed networked systems.
- Developed company standards for software development (System Requirements Specifications, Design Documentation).
- Designed and implemented algorithms for cost optimization applications. Used dynamic programming, discrete optimization heuristics, and APIs.

List of Publications

Papers

1. Mathialakan Thavappiragasam, Vivek Kale, Oscar Hernandez and Ada Sedova. ***Addressing Load Imbalance in Bioinformatics and Biomedical Applications: Efficient Scheduling across Multiple GPUs*** In Proceedings of 12th International Workshop on High Performance Bioinformatics and Biomedicine. December 9, 2021. Houston, Texas, USA.
2. Barbara Chapman, Buu Pham, Charlene Yang, Christopher Daley, Colleen Bertoni, Dhruva Kulkarni, Dhruva Kulkarni, Dossay Oryspayev, Ed D'Azevedo, Helen He, Johannes Doerfert, Keren Zhou, Kiran Ravikumar, Mark Gordon, Mauro Del Ben, Meifeng Lin, Melisa Alkan, Michael Kruse, Oscar Hernandez, P. K. Yeung, Paul Lin, Peng Xu, Swaroop Pophale, Tosaporn Sattasathuchana, Vivek Kale, William Huhn and Dhruva Kulkarni. ***Outcomes of OpenMP Hackathon: OpenMP Application Experiences with the Offloading Model: Part 1*** In Proceedings of 17th International Workshop on OpenMP, IWOMP 2021, Bristol, UK, September 14–16, 2021.
3. Barbara Chapman, Buu Pham, Charlene Yang, Christopher Daley, Colleen Bertoni, Dhruva Kulkarni, Dhruva Kulkarni, Dossay Oryspayev, Ed D'Azevedo, Helen He, Johannes Doerfert, Keren Zhou, Kiran Ravikumar, Mark Gordon, Mauro Del Ben, Meifeng Lin, Melisa Alkan, Michael Kruse, Oscar Hernandez, P. K. Yeung, Paul Lin, Peng Xu, Swaroop Pophale, Tosaporn Sattasathuchana, Vivek Kale, William Huhn and Dhruva Kulkarni. ***Outcomes of OpenMP Hackathon: OpenMP Application Experiences with the Offloading Model: Part 2*** In Proceedings of 17th International Workshop on OpenMP, IWOMP 2021, Bristol, UK, September 14–16, 2021.

4. Seonmyeong Bak, Colleen Bertoni, Swen Boehm, Reuben Budiardja, Barbara M. Chapman, Johannes Doerfert, Markus Eisenbach, Hal Finkel, Oscar Hernandez, Joseph Huber, Shintaro Iwasaki, Vivek Kale, Paul R.C. Kent, JaeHyuk Kwack, Meifeng Lin, Piotr Luszczek, Ye Luo, Buu Pham and P.K. Yeung. ***OpenMP Application Experiences: Porting to Accelerated Nodes***. In Journal of Parallel Computing. October 23rd, 2021.

5. Vivek Kale, Wenbin Lu, Anthony Curtis, Abid Malik, Barbara Chapman and Oscar Hernandez. **Toward Supporting MultiGPU targets via taskloop and User-defined Schedules**. Proceedings of the 2020 International Workshop of OpenMP. September 23-25, 2020. Austin, USA. (virtual)

6. Jonas H Müller Korndörfer, Florina M. Ciorba, Akan Yilmaz, Christian Iwainsky, Johannes Doerfert, Hal Finkel, Vivek Kale, Michael Klemm. ***A Runtime Approach for Dynamic Load Balancing of OpenMP Parallel Loops in LLVM***. The International Conference for High Performance Computing Networking, Storage, and Analysis. November 19, 2019. Denver, Colorado, USA.

7. Vivek Kale, Christian Iwainsky, Michael Klemm, Jonas H. Muller Korndorfer, Florina M. Ciorba. ***Toward a Standard Interface for User-Defined Scheduling in OpenMP***. International Workshop on OpenMP. September 23, 2019. Auckland, New Zealand.

8. Vivek Kale and William D. Gropp. ***Composing Low-Overhead Scheduling Strategies for Improving Performance of Scientific Applications***. IWOMP 2015. October 2015. Aachen, Germany.

9. Simplice Donfack, Vivek Kale, Laura Grigori and William D. Gropp. ***Hybrid Static/Dynamic Scheduling for Already Optimized Dense Matrix Factorizations***. IPDPS 2012. May 2012. Shanghai, China.

10. Vivek Kale, Abhinav Bhatele and William D. Gropp. ***Weighted Locality-sensitive Scheduling for Noise Mitigation on Multicore Clusters***. HiPC 2011. December 2011. Bangalore, India.

11. Vivek Kale and William D. Gropp. ***Load Balancing for Regular Meshes on a Cluster of SMPs with MPI***. EuroMPI 2010. September 2010. Stuttgart, Germany. *(Selected as a Best Paper)*

12. Torsten Hoefler, James Dinan, Darius Buntinas, Pavan Balaji, Brian Barrett, Ron Brightwell, William Gropp, Vivek Kale and Rajeev Thakur. ***MPI+MPI: A New Hybrid Approach to Parallel Programming with MPI Plus Shared Memory***. EuroMPI 2012. September 2012. Madrid, Spain.

13. Amanda Randles, Vivek Kale, Jeff Hammond, William D. Gropp and Efthimios Kaxiras. ***Performance Analysis of the Lattice Boltzmann Model Beyond Navier-Stokes***. IPDPS 2013. May 2013. Boston, USA.

14. Vivek Kale. ***Towards Using and Improving the NAS Parallel Benchmarks: A Parallel Patterns Approach***. ParaPLoP 2010. April 2010. Carefree, USA.

15. Vivek Kale and Edgar Solomonik. ***Parallel Sorting Pattern***. ParaPLoP 2010. April 2010. Carefree, USA.

16. Vivek Kale. ***The Correlation between Parallel Patterns and the NAS Parallel Benchmarks***. ICSE 2010. May 2010. Johannesberg, South Africa.

Extended Abstracts

1. Vivek Kale. ***More Frequent and Regular Engagement for Success of Scientific-Software and HPC Software on Next-generation Supercomputers***. 2021 U.S. Department of Energy's Workshop on the Science of Scientific-Software. Virtual. December 15, 2021.

2. Vivek Kale and Martin Kong. ***Enhancing Support in OpenMP to Improve Data Locality in Application Programs Using Task Scheduling***. OpenMPCon 2018. September 2018. Barcelona, Spain.
3. Vivek Kale and William D. Gropp. ***A User-defined Schedule for OpenMP***. Extended Abstract. OpenMPCon 2017. September 2017. New York, USA.
4. Vivek Kale. ***A Pattern Language for Dynamic Scheduling***. ParaPLoP 2011. May 2011. Carefree, USA.

Posters

1. Raul Torres, Vivek Kale, Abid Malik, Tom Scogland, Roger Ferrer and Barbara M. Chapman. ***Support in OpenMP for Multi-GPU Parallelism***. The International Conference for High Performance Computing Networking, Storage, and Analysis. November 19, 2021. St. Louis, Missouri, USA.
2. Vivek Kale and Oscar Hernandez. ***Performance Portability of User-defined Loop Schedules***. DOE PPP 2019. April 2019. Denver, USA.
3. Vivek Kale, Harshitha Menon and Karthik Senthil. ***Adaptive Loop Scheduling with Charm++ to Improve Performance of Scientific Applications***. SC 2017. November 2017. Denver, USA. *(Selected as a Candidate for Best Poster)*
4. Vivek Kale, Simplice Donfack, Laura Grigori and William D. Gropp. ***Balancing the Tradeoff Between Load Balancing and Locality to Improve Performance of Scientific Applications***. SC 2014. November 2014. New Orleans, USA.
5. Vivek Kale, Amanda Randles and William D. Gropp. ***Locality-Optimized Mixed Static/Dynamic Scheduling for Load Balancing on SMPs***. EuroMPI/ASIA 2014. September 2014. Kyoto, Japan.

Invited Talks

1. Vivek Kale. ***Low-overhead Loop Scheduling in LLVM's OpenMP to Improve Performance of Scientific Applications on Heterogeneous Nodes***. OpenMP Booth Talk at Supercomputing 2021. November 8, 2021.

   Vivek Kale. ***A Software Ecosystem for the DoE Exascale Computing Program's LLVM OpenMP Implementation.***. Talk in SIAM CSE 2021 mini-symposium on Programming Environments. March 1, 2021.
2. Vivek Kale. ***Locality-sensitive Loop Scheduling in SOLLVE's OpenMP***. OpenMP Booth Talk at Supercomputing 2020. November 6, 2020.
3. Vivek Kale. ***Low-overhead Loop Scheduling in OpenMP***. OpenMP Booth Talk at Supercomputing 2019. November 20, 2019.
4. Vivek Kale. ***Loop Scheduling in OpenMP***. OpenMP Booth Talk at Supercomputing 2018. November 15, 2018.

Services

- Technical Program Committee and reviewer for 2021 ACM International Conference on Parallel Processing (ICPP)
- Technical Program Committee and reviewer for 2021 IEEE/ACM Hierarchical Parallelism for Exascale Computing (HiPar) workshop at Supercomputing 2021
- Technical Program Committee and reviewer for 2020 ACM International Conference on Parallel Processing (ICPP)
- Technical Program Committee and reviewer for 2020 IEEE/ACM Hierarchical Parallelism for Exascale Computing (HiPar) workshop at Supercomputing 2020
- Reviewer for 2017 Elsevier Parallel Computing Journal

- Member of Organization Committee for SC 2018's Early Career Program
- Member of Selection Committee for SC 2018's Experiencing HPC for Undergraduates Program

Membership of Organizations

Society for Industrial and Applied Mathematics, Association for Computing Machinery, Institute of Electrical and Electronics Engineers, Association for Computing Machinery's Special Interest Group for High-Performance Computing

Teaching Experience

1. Teaching Assistant for Programming Studio (CS 242) at University of Illinois at Urbana-Champaign for Spring 2009.
2. Teaching Assistant for Programming Studio (CS 242) at University of Illinois at Urbana-Champaign for Fall 2009.
3. Teaching Assistant for Programming Studio (CS 242) at University of Illinois at Urbana-Champaign for Spring 2010.