Vivek Kale, PhD

phone: +01 1-217-369-7996 . *e-mail*: vivek.lkale@gmail.com . *web*: `https://vivek112.googlepages.com`

U.S. Citizen

Education

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

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

Awards

- 2018 Argonne Training Program on Exascale Computing Invitee and Participant
- SC 2017 Early Career Program Invitee and Participant
- 2015 Heidelberg Laureate Forum Invitee and Participant.
- Lawrence Scholar at Lawrence Livermore National Laboratory.
- Recipient of 2015 Heidelberg Laureate Forum Travel

Courses Taken

- Algorithms
- Numerical Methods
- Designing and Building Scientific Applications
- Artificial Intelligence
- Software Engineering
- Program Optimization
- Parallel Software Design Patterns
- Parallel Computer Architecture
- Advanced Distributed Systems
- Advanced Computer Networking

Technical Skills

Languages: C, C++, python, Fortran, bash, csh, VHDL, Matlab, Java

Tools: LaTeX, gnuplot, emacs, autoconf, cmake, svn, git, Globus Toolkit

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

Performance Profiling Tools: OpenSpeedShop, hpcToolkit, PMPI, Intel VTune

Platforms: Intel Xeon, IBM Power7, AMD Opteron, NVIDIA Keplar, Intel Xeon Phi

Experience

Charmworks, Inc. Research Software Developer Jun '18 - present

- Integrating a shared memory library for sophisticated loop scheduling strategies, with some strategies being based on strategies I've developed for my dissertation, into the current version of Charm++.
- Comparing performance of a loop scheduling strategy available in the integrated shared memory library with the performance of the corresponding loop scheduling strategy available in LLVM's OpenMP library.
- Providing feedback for tutorials on Charm++ to improve them.

University of Southern California Computer Scientist Dec. '16 - present

- Working with postdoc from LLNL on a proposal to study techniques that combine loop scheduling and load balancing to improve performance of scientific applications.
- Working with OpenMP Language Committee to support user-defined schedules in OpenMP.
- Translating an x-ray tomography code written in Matlab code to C code and then parallelizing it to run on a supercomputer having nodes with GPGPUs.
- Working on modifications to LLVM compiler to support new OpenMP loop schedules.
- Worked on ensuring external network infrastructure to support transfer of application code's input data files were adequate for an application code's efficient execution using the Globus Toolkit.
- Working in team to manage computational performance aspects of running an application program involving Fast Fourier Transformation and image reconstruction algorithms.

Charmworks, Inc. Developer Jan. '16 - Nov. '16

- Implemented mixed static/dynamic loop scheduling strategies within Charm++'s thread scheduling library.
- Helped to improve portability of Charm++ to a variety of platforms.
- Assisted with business aspects of a high-tech startup.

University of Illinois Postdoctoral Associate Jul. '15 – Dec. '15

- Developed library that allows application programmers to use strategies from dissertation.
- Adapted a plasma physics application code to work on a GPGPU processor and Intel Xeon Phi.
- Incorporated over-decomposition and locality-aware scheduling into strategies from dissertation.

Lawrence Livermore Nat'l Lab Lawrence Scholar Feb. '12 – Jun. '15

- 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 and Berkeley Nat'l Lab Scholar Jun. '10 - Sep. '10 / Jun. '11 - Sep. '11

- Experimented with different OpenMP parameters of implemented MPI+OpenMP application code to understand performance optimizations on LLNL supercomputers.
- Analyzed results for the performance tests developed on NERSC machines, and compared with collectives in reference to MPI (mpich2) runtime system.
- Modified OpenMP gomp runtime system in order to integrate low-overhead schedulers within it.

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.

Research Overview

My work focuses on improving performance of scientific codes run on supercomputers with multi-core nodes, in particular on developing strategies to efficiently schedule iterations of a computational loop of an application code on cores of a multi-core node, which was the focus of my PhD work.

Prior to my PhD work, I was involved in the effort on Our Parallel Patterns developed by University of California at Berkeley at University of Illinois at Urbana-Champaign, resulting in publications 14-15 and, later, 16-17. I also was involved in work on the MPI shared memory extensions model for MPI-3, also known as the MPI+MPI model, resulting in publication 12. Additionally, I worked on performance optimizations a code involving a Lattice-Boltzmann computation that simulated blood flow of a human's heart run on a cluster of multi-core nodes, resulting in publication 13. My dissertation work was studying and developing lightweight loop schedulingstrategies to improve the scalability of bulk-synchronous MPI+OpenMP application codes run on supercomputers with multi-core nodes. The work resulted in publications 2-11 and 17. The scheduling strategies developed can be beneficial to mitigate the amplification problem, a problem shown to cause serious performance bottlenecks for bulk-synchronous and loosely synchronous MPI applications running on next-generation supercomputers having on the order of 1,000,000 nodes, as discussed in publications 8 and 11. The strategies have been applied to Communication-Avoiding dense matrix factorization codes, studied in publications 6 and 9, regular mesh computations, studied in publication 11, and n-body simulations, studied in publications 5 and 7. Additionally, a general development methodology for the strategies was designed for use in a variety of real-world numerical simulations. The effort to do so was a study of composing different types of low-overhead loop scheduling strategies to form new loop scheduling strategies and resulted in publication 6.

My current work involves performance optimizations that include techniques of auto-tuning and loop scheduling of ptychography solvers intended to run on supercomputers having nodes with GPUs and/or MICs. I'm working towards a publication that describes performance optimizations done to the application code. I'm involved in the development of user-defined loop schedules for OpenMP, resulting in publication 3. Additionally, I've done research on a technique that combines OpenMP-style loop scheduling and Charm++ load balancing, resulting in publication 2.

List of Publications

1. 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.
2. Vivek Kale, Harshitha Menon and Karthik Senthil. ***Adaptive Loop Scheduling with Charm++ to Improve Performance of Scientific Applications***. SC '17. November 2017. Denver, USA. *(Selected as a Candidate for Best Poster)*
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, Harshitha Menon and Karthik Senthil. ***Adaptive Loop Scheduling with Charm++ to Improve Performance of Scientific Applications***. Technical Report. University of Southern California. May 2017.
5. Vivek Kale and William D. Gropp. ***Composing Low-Overhead Scheduling Strategies for Improving Performance of Scientific Applications***. IWOMP 2015. October 2015. Aachen, Germany.

6. Vivek Kale, Simplice Donfack, Laura Grigori and William D. Gropp. ***Balancing the Trade-off Between Load Balancing and Locality to Improve Performance of Scientific Applications***. SC '14. November 2014. New Orleans, USA.
7. 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.
8. Vivek Kale, Todd Gamblin, Torsten Hoefler, Bronis R. de Supinski and William D. Gropp. ***Slack-conscious Lightweight Loop Scheduling for Scaling Past the Noise Amplification Problem***. SC '12. November 2012. Salt Lake City, USA.
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.
17. Vivek Kale. ***A Pattern Language for Dynamic Scheduling***. ParaPLoP 2011. May 2011. Carefree, USA.

Membership of Organizations

- Society for Industrial and Applied Mathematics
- Association for Computing Machinery
- Institute of Electrical and Electronics Engineers
- Association for Computing Machinery's SIGHPC

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.

Services

- 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