**Vivek Kale**

217-369-7996.[vivek@illinois.edu](mailto:vivek@illinois.edu). <http://vivek112.googlepages.com>.

**U.S. Citizen**

**Education:**

PhD in Computer Science at University of Illinois at Urbana-Champaign

Lawrence Scholar at Lawrence Livermore National Laboratory

GPA: 3.95/4.00

**Advisor:**William Gropp

**Focus:** Dynamic scheduling for multi-core architectures

**General Area of Interest:** High-Performance Scientific Computing, Computer Architecture, Parallel Computation, work stealing, auto-tuning, dense linear algebra, computational physics simulations

***Overview of Work:*** I have worked on low-overhead dynamic scheduling strategies for performance tuning of MPI+OpenMP codes on next-generation clusters of SMPs. These low-overhead scheduling strategies aim to achieve the best balance between load balance and locality, two important characteristics of MPI+OpenMP code that help to obtain good performance for next-generation clusters of SMPs. The techniques have been applied to dense matrix factorization codes, specifically Communication-avoiding LU and Communication-avoiding QR. I have applied the strategies to regular mesh computations and Lattice-Boltzmann simulations, and most recently, n-body simulations. 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 exa-scale machines or the cloud.

**Publications:**

1. Vivek Kale, Simplice Donfack, Laura Grigori, William D. Gropp. ***Balancing the Tradeoff Between Load Balancing and Locality to Improve Performance of Scientific Applications.*** SC' 14. New Orleans, LA.

1. Vivek Kale, Amanda Peters Randles, William D. Gropp. ***Locality-Optimized Mixed Static/Dynamic Scheduling for Load Balancing on SMPs****.* EuroMPI/ASIA 2014. Kyoto, Japan.
2. Vivek Kale, Todd Gamblin, Torsten Hoefler, Bronis R. de Supinski, William D. Gropp. ***Slack-conscious Lightweight Loop Scheduling for Scaling Past the Noise Amplification Problem****.* SC ‘12 Poster. Salt Lake City, Utah.
3. Simplice Donfack, Vivek Kale, Laura Grigori, William D. Gropp. ***Hybrid Static/Dynamic Scheduling for Already Optimized Dense Matrix Factorizations***. IPDPS 2011. Shanghai, China.
4. Vivek Kale, Abhinav Bhatele, William D. Gropp. ***Weighted Locality-Sensitive scheduling for Noise Mitigation on Multi-core Clusters.***HiPC 2011. Bangalore, India.
5. Vivek Kale, William D. Gropp. ***Load Balancing for Regular Meshes on a Cluster of SMPs with MPI.*** EuroMPI ’10. Stuttgart, Germany. *(Selected as a Best Paper).*
6. Torsten Hoefler, James Dinan, Darius Buntinas, Pavan Balaji, Brian Barrett, Ron Brightwell, William Gropp, Vivek Kale, Rajeev Thakur *.* ***MPI+MPI: A New Hybrid Approach to Parallel Programming with MPI Plus Shared Memory*.** EuroMPI ’12. Madrid, Spain.
7. Amanda Peters Randles, Vivek Kale, Jeff Hammond, William D. Gropp, Efthimios Kaxiras. ***Performance Analysis of the Lattice Boltzmann Model Beyond Navier-Stokes.*** IPDPS 2013. Boston, MA.

**Experience:**

**Lawrence Livermore National Laboratory  Lawrence Scholar                     Feb 2012 – Jun 2014**

* Obtained measurements for within-node imbalances on LLNL supercomputers, and developed a cost model for these imbalances.
* Created a software system for automated performance optimization and application programmer usability of low-overhead scheduling strategies.
* Developed a theoretical analysis for obtaining best performing scheduling parameters for a particular application and architecture.
* Developed a ROSE-based custom compiler for automatically transforming MPI+OpenMP applications to use my low-overhead scheduling technique and runtime.
* Assessed further opportunities for performance improvement of low-overhead schedulers, including improvement of spatial locality of low-overhead schedulers.

**INRIA-Saclay, France Visiting Research Position Jan 2012 - Feb 2012**

* Carried out experimentation that projected performance slowdowns of Communication-Avoiding LU and QR factorization at extreme-scale.
* Developed theoretical analysis for Communication-Avoiding LU and QR factorization with low-overhead scheduling techniques applied, and used analysis to show mitigation of the performance slowdowns at extreme-scale using our techniques.

**Lawrence Livermore National Laboratory              Scholar                    Jun 2011 - Sept 2011**

* Modified OpenMP gomp runtime system in order to integrate low-overhead schedulers within it.
* Designed a multi-level low-overhead scheduling strategy, where scheduling could occur across nodes.
* Profiled overheads of OpenMP gomp runtime system’s schedulers using profiling tool hpcToolkit.

**Lawrence Berkeley National Laboratory Intern Aug 2010 – Sept 2010**

* Wrote a performance testing suite for the collectives in Berkeley’s UPC (bupc) runtime system.
* Analyzed results for the performance tests on NERSC machines, and compared with collectives in reference MPI (mpich2) runtime system.

**Lawrence Livermore National Laboratory           Scholar          Jun 2010 - Aug 2010**

* Helped to add OpenMP threading to an MPI application code simulating laser-plasma interactions.
* Experimented with different OpenMP parameters of resulting MPI+OpenMP application code to understand how to improve its performance on LLNL supercomputers.

**Goldman-Sachs     Summer Analyst Jun 2009 – Sept 2009**

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

**Technical Skill Highlights:**

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

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

*Libraries for Parallelism:* POSIX threads (Pthreads), MPI (mpich2 and mpich3), OpenMP (gomp), UPC (bupc)

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

*Platforms*: clusters of NUMA multi-core nodes, clusters of SMP nodes, GPUs, desktop multi-core processors