**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.

**Highlights**: (add highlights here)

**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.

**Awards:**

1. *Lawrence Scholar Fellowship:* This is a fellowship sponsored by the Lawrence Livermore National Laboratory and Department of Energy providing up to 4 years of funding for graduate research. Each year a selection from a pool of candidates consisting of PhD students is made from areas including Physics, Chemistry, Biology, and Computer Science. This fellowship award allows a PhD student to conduct their PhD research at Lawrence Livermore National Laboratory in Livermore, California, with guidance for research given by their advisor and one or more mentors at Lawrence Livermore National Laboratory. This fellowship was awarded to me in November of 2011.

**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.

**Coding Projects:**

(top 3 listed):

1. A large coding project that I worked on was runtime software system for our scheduling strategy, which automatically adjusted the parameters of the low-overhead scheduling strategies implemented in the scheduling library described above. This size of the runtime software is about 7500 lines of code. The runtime is written in C++. Additionally, supporting application profiling was done using python. My advisor and LLNL supervisor led the project for this runtime. I wrote about 80% of the code for the runtime, and the other 20% was written in conjunction with software engineers at Lawrence Livermore National Laboratory. The purpose of the project was to provide a runtime to automatically adjust scheduler parameters for obtaining further performance improvements of application codes, and to substantially reduce application programmer effort to use our technique through automatic transformation of the application programmer’s code. The functionality is as follows. Given a threaded scientific application code using MPI, the application programmer compiles the code with our provided compiler. Our provided compiler modifies the code by placing our runtime’s function calls at the beginning and end of the threaded computation regions of the code. Then, the application programmer runs the code using our runtime software. On every invocation of the threaded computation region of the code, the function at the beginning of the region decides the best parameter values for the scheduling strategy applied in the immediately subsequent computation region. The parameter values are chosen based on history gathered through the function call invoked at the end of the computation region. The expected result is reduced application code execution time compared to that of the original application code.

2. Low-overhead scheduling and locality-aware scheduling for LU/CALU and QR/CAQR:

3. Low-overhead scheduling library with macros for scientific application codes:

**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

**Additional:**

* *Publications Read*: Papers from SC conference series, IEEE Transactions of Parallel and Distributed Systems Journal, SIAM Journal of Computing.
* *Conferences Attended*: SC’14, SC ‘13, SC ‘12, IPDPS 2010, IPDPS 2013, SIAM PP ‘10

**10. Are there any technical skills, experiences or interests not on your resume that you would like to highlight?**

Below are technical skills, experiences, or interests not on my resume that I would like to highlight.   
  
1. Lawrence Scholar Fellowship: This is a fellowship sponsored by the Lawrence Livermore National Laboratory and the Department of Energy providing up to 4 years of funding for a student’s PhD thesis research. Each year a selection is made from a pool of candidates consisting of PhD students from areas including Physics, Chemistry, Biology, and Computer Science. This fellowship award allows a PhD student to conduct their PhD research at the Lawrence Livermore National Laboratory in Livermore, California, with guidance for their research given by both their advisor and one or more Senior Technical Staff at the Lawrence Livermore National Laboratory. This fellowship was awarded to me in November 2011. I would like to highlight this fellowship experience because of the valuable connections to scientists and engineers I gained within the lab as well as outside the lab during the fellowship period, with whom I plan to maintain contact with throughout my career.

I developed connections with software engineers and scientists, a skill I believe will be important throughout my career. The work done during the fellowship is in the 1st bullet of the experience section.

This increased my ability to communicate and collaborate.

This fellowship particularly contained a component of the development of software for improving performance of application codes used to do computational simulation of renewable energy technology developed by the Department of Energy. I worked with the laboratory’s scientists and senior software engineers on the development of the software, so as to integrate it with other software in the lab. I also discussed the software with scientists who would use the software in the future (once it is ready for production). Through the development of the software, I gained software engineering experience, as well as experience in team collaboration and project management. I would like to highlight the development of the software because I believe the skills I acquired from it will be relevant to the software engineering aspect of the position at Google that I have applied to.

– happiness.

INRIA-Saclay Visiting Research Position: As part of my PhD research, I was invited to a research laboratory in INRIA-Saclay (France) to continue and enrich collaborative work published in my IPDPS 2012 paper on my low-overhead multi-core scheduling in the context of numerical linear algebra computations. Specifically, I did additional experimentation and further theoretical analysis. I would like to highlight this experience because I believe the performance modeling to formulate performance expectations of numerical linear algebra algorithms and the theoretical analysis to determine the best performing scheduling parameter values of my technique I did during this experience is relevant to the algorithms aspect of the position at Google that I have applied to.

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.

It highlights performance tuning techniques I developed. It highlights the importance of the intersection between mathematics and computer science. Of its relevance to theoretical analysis. This is relevant for the Google software engineering position I’m applying to because. 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.

Relevance to Google’s position. Having seen Google’s software development positions emphasis on software engineering, I feel that this.

I worked with the laboratory’s scientists and senior software engineers on the development of the software, and discussed the software with two scientists currently using my software, and to a few other scientists who might use the software in the future (once it is ready for production).

Vivek

1. Lawrence Scholar Fellowship: This is a fellowship sponsored by the Lawrence Livermore National Laboratory and the Department of Energy providing up to 4 years of funding for student’s PhD thesis research. Each year a selection is made from a pool of candidates consisting of PhD students from areas including Physics, Chemistry, Biology, and Computer Science. This fellowship award allows a PhD student to conduct their PhD research at the Lawrence Livermore National Laboratory in Livermore, California, with guidance for their research given by both their advisor and one or more Senior Technical Staff at the Lawrence Livermore National Laboratory. This fellowship was awarded to me in November 2011. I would like to highlight this fellowship experience because it gave me the ability to