I am writing this letter in support of Vivek Kale for his application for invitation to the Heidelberg Laureate Forum. I have known Vivek Kale since I moved to University of Illinois at Urbana Champaign as a visiting professor in March 2009 to establish the INRIA-Illinois joint-laboratory on peta-scale computing at NCSA, and have had a continued relationship with Vivek throughout his PhD. Before moving to Illinois, I led a team of 30+ people with up to 7 permanent researchers, working on High Performance Parallel and Distributed Computing. In particular, I published at the ACM/IEEE Supercomputing conference in 2000 one of the first papers about hybrid MPI+OpenMP programming of cluster of multi-core processors, a novel method of implementing a large number of numerical simulations and scientific applications. I remain engaged in this area.

The MPI+OpenMP programming model is currently one of the most popular ones to use for scientific simulations intended to run at large-scale on modern supercomputers. However, although MPI+OpenMP implementations have made outstanding progress since 2000 on a large number of current supercomputers of 1000 processors (or, nodes), managing the OpenMP threads and their workload efficiently still remains complex as scientists scale applications to 10,000 nodes or more. The problem comes from the fact that bulk synchronous execution is used by many HPC applications, due to the domain-specific constraints in the application, which cannot relax the synchrony of communication. Given this, if there are small performance fluctuations due to operating system events (which we refer to as noise, or system interference) in an MPI process within a particular timestep of a simulation, that noise can amplify, producing a major performance slowdown at large scale. This problem was first established in 2003 by Petrini et al. Much work has been done to quantify and understand the impact of system noise (Tsasfir ICS05, Hoefler SC10), showing the continued and increasing importance of this problem. A second related problem is that within-node application-level imbalances are a significant factor to across-node (application-level) load imbalance, and its contribution can be larger as more cores per node exist on next-generation supercomputers.

Some low-level strategies such as co-scheduling, as suggested by Petrini, have been used to mitigate impact of system interference, but have been limited in their performance improvements, and also do not handle within-node imbalances induced by non-OS events such as transient hardware variations. For handling within-node application-level imbalances, measurement-based (across-node) load balancing strategies from Zoltan or Charm++ runtime systems could be adapted to work within a node, but using these strategies is fundamentally problematic: measurement-based load balancing relies on persistence of load imbalance patterns based on prior simulation timesteps, but the nature of within-node imbalance is impersistent because of the combination of load imbalance due to system interference and load imbalance due to the application. To address both of these problems together, Vivek followed a new strategy: develop an intelligent mixture of OpenMP static+dynamic scheduling so that the scheduler overheads could be reduced and self-adaptivity could be increased, both of which can help to significantly increase performance of scientific applications. The proposed approach consists of experimentally tuning various mixtures of static and dynamic scheduling, so that within-node imbalances can be mitigated without allowing the overhead of dynamic scheduling itself to extend the critical path of application execution. The resulting paper that he wrote on this basic scheduling strategy was his first published paper and received a best paper award at the EuroMPI 2010 conference held in Stuttgart.

Vivek then continued this work towards a Ph.D. under the supervision of Bill Gropp. He first focused on applying his mixed static/dynamic scheduling strategies to dense matrix factorizations. Dense matrix factorizations are a class of numerical linear algebra computations that are highly time consuming in many engineering simulations, e.g., simulation of air flow across an airplane wing. Vivek worked with Simplice Donfack of INRIA to improve the performance of an implementation of a communication-avoiding dense LU matrix factorization (CALU) for multi-core architectures, written by Jim Demmel at UC-Berkeley and Simplice Donfack. Using the mixed static/dynamic scheduling strategy gave a 30% improvement over the original CALU code. Using the strategy also gave a 38% gain over an Intel MKL implementation, a widely-used industry standard. Vivek then published a paper with Simplice Donfack to discuss these results in IPDPS 2012. Shortly after this, he visited INRIA-Saclay as a visiting researcher for a month to work with Simplice to extend work to show its benefits of his scheduling scheme to dense matrix factorization implementations in the context of future extreme-scale supercomputers. His visiting research assistantship was done under supervision of Laura Grigori, a colleague of mine. This collaboration shows Vivek’s ability to work together with other researchers at an international level and to contribute his strengths for a greater success of the team.

Concurrently, Vivek worked on enhancing his strategy on using the measurement-based load balancing techniques used in Charm++ RTS to further guide his schedulers. This allowed performance gains on production machines such as ORNL's Cray and TACC's Ranger. This is a fundamental enhancement because it allows his scheduler using knowledge of the previous iterations to make scheduling decisions. Through the use of these techniques, Vivek's scheduler incorporated an aspect of intelligence during the application's execution, which allowed for increasing performance of an application. These enhancements are particularly beneficial for bulk-synchronous simulations involving several application timesteps. This work led to a publication of a paper in HiPC2011. This was done in collaboration with Abhinav Bhatele, at that time PhD student/postdoc working within the Charm++ group at Illinois, and now a post-doc at LLNL. This conference is highly competitive (~19% acceptance rate). Vivek presented this paper in Bangalore, India in December 2011.

In January 2012, Vivek was awarded a Lawrence Scholar Fellowship, which awarded him funding from the United States Department of Energy to do research on his scheduling strategies for mitigating the amplification on DOE supercomputers. The award funding is for up to 4 years. He has worked from Lawrence Livermore Laboratory in California while doing his PhD at Illinois. His collaboration team involves Bronis de Supinski (mentor), Todd Gamblin, Torsten Hoefler, and Bill Gropp (advisor). The five of them meet for weekly conference calls, discussing ongoing progress on the implementation of his scheduling techniques. During his time at LLNL up to this point, Vivek has investigated how to make his scheduling approach usable for real-world applications. He has developed, under the supervision of Bronis de Supinski, a dynamic scheduler library integrated within OpenMP, where OpenMP uses information of the MPI runtime to possibly further guide the scheduling strategies. Specifically, each process tunes the scheduler parameters based on the process properties. This work resulted in contributions of a source-to-source transformation of OpenMP codes, along with new slack prediction techniques. He also addressed the problem of building an interface between two programming models within a hybrid programming model. He has developed other techniques such as using a scheduler variable task sizes, and integrated them into his software infrastructure. He along with his co-authors Bronis de Supinki, Todd Gamblin, Torsten Hoefler, and Bill Gropp has published a poster in the Supercomputing Conference in 2012 and a journal paper on dynamic scheduling techniques. They recently submitted to a top-tier high-performance computing conference the work on experimentation and development of schedulers done at LLNL, with a particular emphasis on the enhancements to the scheduler strategies done through systematic software implementation. With this, Vivek developed a scheduling library to attain additional performance benefits of pure static scheduling in his mixed static/dynamic technique, focusing on load imbalanced applications which use partitioning of work (e.g., Morton Orders for Molecular Dynamics Simulations) to improve spatial locality, which is beneficial only when using static scheduling. Vivek made this pre-determined partitioning of work still be beneficial in the context of his mixed static/dynamic scheduling strategy. This led to a paper in EuroMPI 2014 presented in September of last year.

Vivek also concluded his collaboration with Simplice in November 2014, in which he presented a poster with Simplice at SC ’14 in New Orleans. Within this published work, he discusses advanced theoretical analysis of the mixed static/dynamic scheduling techniques in the context of numerical linear algebra computations, and shows how the lessons learned from applying mixed static/dynamic scheduling to numerical linear algebra computations can be used towards improving performance of scientific applications.

Finally, Vivek defended his PhD last week, with his thesis dissertation discussing low-overhead scheduling for improving performance of synchronous MPI programs. He is looking for post-doc positions to continue his research on low-overhead schedulers by integrating his schedulers with classical across-node load balancers, and to use techniques for any hybrid MPI+X programming models, not just MPI+OpenMP programming models.

The above elements and the global consistency of his research projects clearly demonstrate that Vivek Kale has the maturity of an excellent researcher, developing a timely and important research topic for the HPC community and exploring many of its theoretical and practical aspects, including the best way for others to use his approach.

In conclusion, Vivek Kale is an outstanding Ph. D. student. His research topic concerns a critical problem that needs to be solved in the near future. This is one of the fundamental problems in parallel computing and in the general field of computer science. This problem will be relevant as long as extreme-scale computing is necessary for the advancements in science and engineering. He already has established strong contacts and developed collaborations with several top researchers in the HPC domain. His work has shown to be directly beneficial and relevant in both a theoretical as well as practical setting. Based on Vivek's current work, along with his recently defined long-term goals as discussed in his letter of motivation, Vivek will benefit greatly through interactions with top scientists at the Heidelberg Laureate Forum, and also benefit from networking with other young aspiring computer scientists and mathematicians. Most importantly, I believe he will contribute to lively discussions and will go with motivation and great interest if he is invited to attend. Vivek has my highest recommendation to be a part of the Heidelberg Laureate Forum in August of this year.