Programming Models for Petascale to Exascale

Katherine Yelick

Electrical Engineering and Computer Sciences Department, UC Berkeley
National Energy Research Scientific Computing Center, Lawrence Berkeley National Lab

Abstract:

Multiple petascale systems will soon be available to the computational science community and will represent a variety of architectural models. These high-end systems, like all computing platforms, will have an increasing reliance on software-managed on-chip parallelism. These architectural trends bring into question the message-passing programming model that has dominated high-end programming for the past decade. In this talk I will describe some of the technology challenges that will drive the design of future systems and their implications for software tools, algorithm design, and application programming. In particular, I will show a need to consider models other than message passing as we move towards massive on-chip parallelism. I will talk about a class of Partitioned Global Address Space (PGAS) languages, which are an alternative to both message passing models like MPI and shared memory models like OpenMP. PGAS languages offer the possibility of a programming model that will work well across a wide range of shared memory, distributed memory, and hybrid platforms. Some of these languages, including UPC, CAF and Titanium, are based on a static model of parallelism, which gives programmers direct control over the underlying processor resources. The restricted nature of the static parallelism model in these languages has advantages in terms of implementation simplicity, analyzability, and performance transparency, but some applications demand a more dynamic execution model, similar to that of Charm++ or the recently developed HPCS languages (X10, Chapel, and Fortress). I will describe some of our experience working with both static and dynamically managed applications and some of the research challenges that I believe will be critical in developing viable programming techniques for future systems.

Bio:

Katherine Yelick is the Director of the National Energy Research Scientific Computing Center (NERSC) at Lawrence Berkeley National Laboratory and a Professor of Electrical Engineering and Computer Sciences at the University of California at Berkeley. She is the author or co-author of two books and more than 85 refereed technical papers on parallel languages, compilers, algorithms, libraries, architecture, and storage. She co-invented the UPC and Titanium languages and demonstrated their applicability across architectures through the use of novel runtime and compilation methods. She also developed techniques for self-tuning numerical libraries, such as the OSKI sparse matrix library, which automatically adapt the code to machine properties. Her work includes performance analysis and modeling as well as optimization techniques for memory hierarchies, multicore processors, communication libraries, and processor accelerators. She has worked with interdisciplinary teams on application scaling, and her own applications work includes parallelization of a model for blood flow in the heart. She earned her Ph.D. in Electrical Engineering and Computer Science from MIT and has been a professor of Electrical Engineering and Computer Sciences at UC Berkeley since 1991 with a joint research appointment at Berkeley Lab since 1996. She has received multiple research awards, as well as teaching awards from both UC Berkeley and MIT.