Antisocial Parallelism:
Avoiding, Hiding and Managing Communication

Katherine Yelick
University of California at Berkeley and Lawrence Berkeley National Laboratory

Future computing system designs will be constrained by power density and total system energy, and will require new programming models and implementation strategies. Data movement in the memory system and interconnect will dominate running time and energy costs, making communication cost reduction the primary optimization criteria for compilers and programmers. Communication cost can be divided into latency costs, which are per communication event, and bandwidth costs, which grow with total communication volume. The trends show growing gaps for both of these relative to computation, with the additional problem that communication congestion can conspire to worsen both in practice

In this talk I will describe some of the main techniques for reducing the impact of communication, starting with latency hiding techniques, including the use of one-sided communication in Partitioned Global Address Space languages. I will describe some of the performance benefits from overlapped and pipelined communication but also note case where there is “too much of a good thing” that causes congestion in network internals. I will also discuss some of the open problems that arise from increasingly hierarchical computing systems, with multiple levels of memory spaces and communication layers.

Bandwidth reduction often requires more substantial algorithmic transformations, although some techniques, such as loop tiling, are well known. These can be applied as hand-optimizations, through code generation strategies in autotuned libraries, or as fully automatic compiler transformations. Less obvious techniques for communication avoidance have arisen in the so-called “2.5D” parallel algorithms, which I will describe more generally as “.5D” algorithms. These ideas are applicable to many domains, from scientific computations to database operations. In addition to having provable optimality properties, these algorithms also perform well on large-scale parallel machines. I will end by describing some recent work that lays the foundation for automating transformations to produce communication optimal code for arbitrary loop nests.

Biography: Katherine Yelick is the Associate Laboratory Director for Computing Sciences at Lawrence Berkeley National Laboratory. She is also a Professor of Electrical Engineering and Computer Sciences at the University of California at Berkeley. She co-invented the UPC and Titanium languages as well as techniques for self-tuning sparse matrix kernels, and has published over 100 technical papers. She earned her Ph.D. in EECS from MIT and has been a professor at UC Berkeley since 1991 with a joint appointment at LBNL since 1996. She has received multiple research and teaching awards, is an ACM Fellow and serves on numerous advising committee, including the California Council on Science and Technology and the National Academies Computer Science and Telecommunications Board.