Office of Science
U.S. DEPARTMENT OF ENERGY

Advanced Scientific Computing Research

The Landscape of Parallel Computing Research: A View from Berkeley (2006)

Krste Asanovic,[1,2,3] Ras Bodik,[1] Bryan Christopher Catanzaro,[1] Joseph James Gebis,[1] Parry Husbands,[3] Kurt Keutzer,[1] David A. Patterson,[1,3] William Lester Plishker,[1] John Shalf,[3] Samuel Webb Williams,[1,3] Katherine A. Yelick;[1,3]
[1]University of California, Berkeley, [2]International Computer Science Institute, [3]Lawrence Berkeley National Laboratory

The path to petascale computing will be paved with new system architectures featuring hundreds of thousands of manycore processors. Such systems will require scientists to completely rethink programming models. A frequently cited white paper called "The Landscape of Parallel Computing Research: A View from Berkeley," addressed the challenge of finding ways to make it easy to write programs that run efficiently on manycore systems.

The recent switch to parallel microprocessors is a milestone in the history of computing. Industry has laid out a roadmap for multicore designs that preserves the programming paradigm of the past via binary compatibility and cache coherence. Conventional wisdom is now to double the number of cores on a chip with each silicon generation.

A multidisciplinary group of Berkeley researchers met nearly two years to discuss this change. In their view, this evolutionary approach to parallel hardware and software may work from 2 or 8 processor systems, but is likely to face diminishing returns as 16 and 32 processor systems are realized, just as returns fell with greater instruction-level parallelism.

They believe that much can be learned by examining the success of parallelism at the extremes of the computing spectrum, namely embedded computing and high performance computing. This led the researchers to frame the parallel landscape with seven questions (Fig. 1), and to recommend the following:

- The overarching goal should be to make it easy to write programs that execute efficiently on highly parallel computing systems.
- The target should be thousands of cores per chip, as these chips are built from processing elements that are the most efficient in MIPS (million instructions per second) per watt, MIPS per area of silicon, and MIPS per development dollar.
- Instead of traditional benchmarks, use 13 (originally 7) "Dwarfs" to design and evaluate parallel programming models and architectures. A dwarf is an algorithmic method that captures a pattern of computation and communication. The 13 Dwarfs are:

  1. Dense linear algebra (e.g., BLAS or MATLAB)

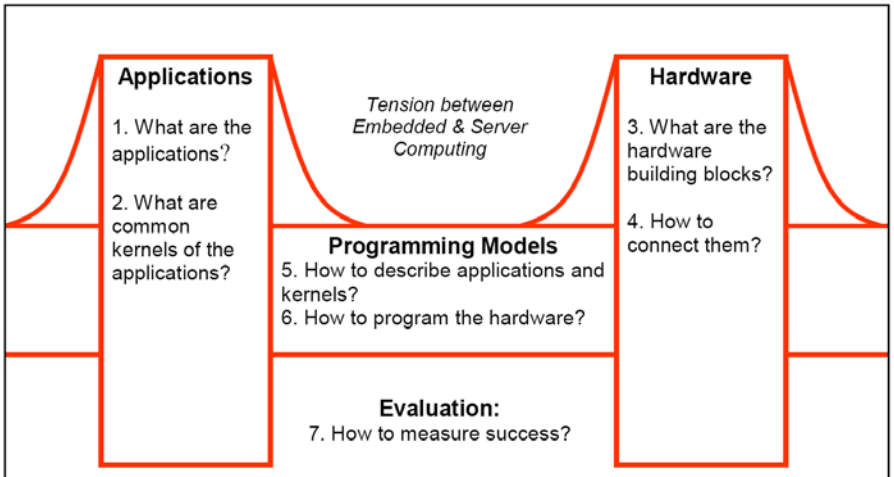

Figure 1. A view from Berkeley: seven critical questions for 21st Century parallel computing. (This figure is inspired by a view of the Golden Gate Bridge from Berkeley.)

2. Sparse linear algebra (e.g., SpMV, OSKI, or SuperLU)
3. Spectral methods (e.g., FFT)
4. N-body methods (e.g., Barnes-Hut, Fast Multipole Method)
5. Structured grids (e.g., Cactus or lattice-Boltzmann magnetohydrodynamics)
6. Unstructured grids (e.g., ABAQUS or FIDAP)
7. MapReduce (e.g., Monte Carlo)
8. Combinational logic
9. Graph traversal
10. Dynamic programming
11. Backtrack and branch-and-bound
12. Graphical models
13. Finite state machine

- "Autotuners" should play a larger role than conventional compilers in translating parallel programs.

- To maximize programmer productivity, future programming models must be more human-centric than the conventional focus on hardware or applications.

- To be successful, programming models should be independent of the number of processors.

- To maximize application efficiency, programming models should support a wide range of data types and successful models of parallelism: task-level parallelism, word-level parallelism, and bit-level parallelism.

- Architects should not include features that significantly affect performance or energy if programmers cannot accurately measure their impact via performance counters and energy counters.

- Traditional operating systems will be deconstructed and operating system functionality will be orchestrated using libraries and virtual machines.

- To explore the design space rapidly, use system emulators based on field programmable gate arrays (FPGAs) that are highly scalable and low cost.

Since real world applications are naturally parallel and hardware is naturally parallel, what the computational science community needs is a programming model, system software, and a supporting architecture that are naturally parallel. Researchers have the rare opportunity to re-invent these cornerstones of computing, provided they simplify the efficient programming of highly parallel systems.

Fig. 1 shows the seven critical questions the authors used to frame the landscape of parallel computing research. They do not claim to have the answers in this report, but they do offer non-conventional and provocative perspectives on some questions and state seemingly obvious but sometimes-neglected perspectives on others.

When it was first published, the "Berkeley View" paper resulted in numerous commentaries and interviews in high performance computing publications, and it received 125 scholarly citations in the first two years after its publication.

Two ongoing research projects have resulted from this paper:

- The Parallel Computing Laboratory, or Par Lab, a multidisciplinary research project exploring the future of parallel processing, funded primarily by Intel and Microsoft (parlab.eecs.berkeley.edu/).

- The RAMP Project (Research Accelerator for Multiple Processors), which focuses on how to build low cost, highly scalable hardware/ software prototypes using field programmable gate arrays (FPGAs) (ramp.eecs.berkeley.edu/).

Publications:

K. Asanovic et al., "The Landscape of Parallel Computing Research: A View from Berkeley," Electrical Engineering and Computer Sciences, University of California, Berkeley, Technical Report No. UCB/EECS-2006-183, Dec. 18, 2006; www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html.

K. Asanovic et al., "The Parallel Computing Laboratory at U.C. Berkeley: A Research Agenda Based on the Berkeley View," Electrical Engineering and Computer Sciences, University of California, Berkeley, Technical Report No. UCB/EECS-2008-23, March 21, 2008; www.eecs.berkeley.edu/Pubs/TechRpts/2008/EECS-2008-23.pdf.

Funding:

University of California, Berkeley; DOE Office of Advanced Scientific Computing Research.