The X-Scale Challenge

George Bosilca

Innovative Computing Laboratory
Electrical Engineering and Computer Science Department,
University of Tennessee, Knoxville, TN, USA
bosilca@cs.utk.edu

The last few years have seen fast changes in the high performance computing field. Some of these changes are related to the computer hardware and architecture. Multi/many core architecture have become prevalent, with architectures more or less exotic and heterogeneous. The overall theoretical computational power of the new generation processors increased greatly, but their programmability still lacks confidence. The changes in shape of the newest architectures has come so rapidly, that we are still deficient in terms of high performance libraries and applications in order to take advantage of all these new features.

At same time, application requirements grow at least at the same pace. Obviously, more computations require more data in order to feed the deepest processor pipelines. More data means either a faster access to the memory, or a faster access to the network. The access speed to all types of memory (network included) lags behind. As a result, extracting the right performance of the current and next generation architectures is still (and will remain for a while) a challenge. These two fields are coming along, providing some very interesting advances over the last few years, but not at the speed of FLOPS. Moreover, the current approach indicates a higher degree of memory hierarchies (Non Uniform Memory Accesses) that have already become another limiting factor for the application performance increase.

Simultaneously, increasing the size of the parallel machines triggers an increase in fault tolerance requirements. While the fault management and recovery topic was thoughtfully studied over the last decade, the recent changes in the number and distribution of the processor's cores have raised some interesting questions. While the question of which fault tolerant approach fits best to the peta-scale environments is still debated, few of these approaches show interesting performances at scale or a low degree of intrusion in the application code. Eventually, the right answer might be somewhere in between a dynamic combination of several of these methods, strictly based on the application's properties and the hardware environment.

As expected, all these changes guarantee a highly dynamic (and exciting from a research point of view) high performance field over the next years. New mathematical algorithms will have to emerge in order to take advantage of these unbalanced architectures. In addition, a tight collaboration between the applications and the high performance libraries developers is/will be *critical* to the future of HPC fields.

F. Cappello et al. (Eds.): EuroPVM/MPI 2007, LNCS 4757, pp. 1–2, 2007.
© Springer-Verlag Berlin Heidelberg 2007

How MPI will adapt to fit into this conflicting environment is still an open question. Over the last few years, MPI has been a very successful parallel programming paradigm, partially due to its apparent simplicity to express basic message exchange patterns and partially to the fact that it increases the productivity of the programmers and the parallel machines. Whatever the future of MPI will be, these two features should stay an indispensable part of its new direction of development.