The Impact of Multicore on Math Software

Jack Dongarra University of
Tennessee Oak Ridge National Laboratory

For at least two decades, HPC programmers have taken for granted that each successive generation of microprocessors would, either immediately or after minor adjustments; make their old software run substantially faster. But three main factors are converging to bring this "free ride" to an end. First, system builders have encountered intractable physical barriers – too much heat, too much power consumption, and too much leaking voltage – to further increases in clock speeds. Second, physical limits on the number and bandwidth of pins on a single chip means that the gap between processor performance and memory performance, which was already bad, will get increasingly worse. Finally, the design trade-offs being made to address the previous two factors will render commodity processors, absent any further augmentation, inadequate for the purposes of tera- and peta-scale systems for advanced applications. But despite the rapidly approaching obsolescence of familiar programming paradigms, there is currently no well understood alternative in whose viability the community can be confident. The essence of the problem is the dramatic increase in complexity that software developers will have to confront. But contrary to the assumptions of the old model, programmers will not be able to consider these cores independently (i.e. multi-core is not "the new SMP") because they share on-chip resources in ways that separate processors do not. This situation is made even more complicated by the other non-standard components that future architectures are expected to deploy, including mixing different types of cores, hardware accelerators, and memory systems. When combined, these changes produce a picture of a future in which programmers must overcome software design problems that are vastly more complex and challenging than in the past in order to take advantage of the much higher degrees of concurrency and greater computing power that new architectures will offer.

****

Jack Dongarra; Innovative Computing Laboratory; Computer Science Dept;
1122 Volunteer Blvd; University of Tennessee; Knoxville TN 37996-3450;
+1-865-974-8295; dongarra@cs.utk.edu; http://www.cs.utk.edu/~dongarra/