Potholes on the Road to Exascale

John Barr

Sometimes the HPC industry gets excited by the hype surrounding progress towards Exascale systems. But, we need to remember that building an Exascale system is not an end in itself. The purpose of such a machine is to run applications, and to deliver insights that could not be achieved with less powerful systems.

In order to build an Exascale system that draws an acceptable amount of power, it is likely that compute accelerators (such as NVIDIA GPUs or Intel Xeon Phi) will be used to maximise the compute power delivered per watt.

But, relatively few applications today actually make use of such accelerators. Ninety percent of systems in the TOP500 list do not use accelerators and, of those that do, the majority of the users of these systems do not use the accelerators. So, in order to develop real applications that can exploit accelerated systems more applications must be ported to this programming model, which means in turn that more programmers must be trained in this style of programming.

Exactly what style of programming is required? While some people feel that a new program development approach is required to meet the demands of Exascale systems, many people think that MPI plus OpenMP plus an accelerator component is sufficient. This glosses over the changes required both to MPI and to program design to build in resilience at the application layer rather that in the hardware — but it's a good first approximation.

Programs are being developed for accelerators today using a mix of OpenMP, OpenACC, OpenCL and CUDA. Many developers want to have a single approach that will work for the more popular hardware choices. This is similar to the situation that arose when distributed memory HPC systems first appeared. Every vendor had their own communications library, and many users and ISVs refused to consider this style of machine until there was a standard way to program them all — and the standard that was developed was MPI.

When OpenACC first appeared it made sense to use this forum to experiment with new approaches while the use of GPUs in HPC was evolving rapidly, with the expectation that the best ideas would then be reintroduced into OpenMP. But, OpenMP and OpenACC now seem to be diverging. Indeed, a comparison of OpenACC and OpenMP on the OpenACC Web site says "efforts so far to include support for GPUs in the OpenMP specification are — in the opinions of many involved — at best insufficient, and at worst misguided."

In order to advance the take up of accelerator programming (with an end game of making more applications potentially ready for Exascale systems), we need to have a programming environment where a single code base can support multiple distinct accelerated hardware platforms. The bifurcation of OpenMP and OpenACC is bad for users, bad for the HPC industry and ultimately, bad for both Intel and Nvidia, the two vendors with the most to gain or lose from accelerator sales.

John Barr is a technology analyst who covers the HPC market. His focus areas include Exascale technologies; processors and accelerators; clusters, grid and cloud computing; networking; storage and software. Formerly he was the Research Director for The 451 Group's work in High Performance Computing.

This blog was originally published on ISC Events *[1] and is reproduced here with permission.*

Source URL (retrieved on *09/02/2015 - 10:53am*): http://www.scientificcomputing.com/blogs/2014/05/potholes-road-exascale

Links:
[1] http://www.isc-events.com/isc14/blogs.html