Position Paper HPCD brainstorm workshop

Pieter Maris

Department of Physics and Astronomy, Iowa State University, Ames, IA 50011

Need for software development support

Over the past decades we have seen an enormous increase in computational resources available for scientific research. Yet, there never seems to be enough resources for any specific research project. I believe part of the issue is that it is extremely hard for domain scientists to keep up-to-date on hardware developments.

Parallel computing has been around for decades, but until recently, it was either shared memory (using e.g. OpenMP) or distributed memory (using e.g. MPI). Current systems all have multi-core nodes (sometimes with accelerators), and the codes I use are mostly hybrid MPI/OpenMP (developed from MPI-only codes). Right now, I am investigating how to efficiently use many-cores on a node (Knights Landing) – with support from NERSC, Intel, and Cray. We as domain scientists have to find and exploit additional layers of parallelism, worry about data-locality, and/or adept to new programming models (GPUs). The lifetime of a new supercomputer is relatively short – of the order of three to four years in full production mode. Domain scientists are pressed for time, and want to focus most of their efforts to the actual science, not to coding and performance improvements. One cannot expect domain scientists to rewrite their codes every three to four years in order to obtain the highest performance on the latest supercomputer; furthermore, often the same code is used on more than one platform, so one faces a new system every year or two. Legacy codes are often kept up-to-date just enough so that they do compile and run on current systems, but that does not mean they all run highly efficiently on those systems.

We as domain scientist need not only hardware computer resources, but we also need help to increase performance of existing as well as new codes under development. Domain scientists are supposed to be experts in their domain, and while they should also be good at scientific computing, they are not necessarily experts in supercomputing. Support scientists (either pure computer scientists or domain scientists that have switched to become supercomputing experts), employed by the supercomputing center and with expert knowledge of the current hardware (and ideally with knowledge of future hardware), willing and able to work with domain scientists on improving efficiency and performance of (legacy) codes are essential in order to maximize the benefits from current and future supercomputers.

Whether this is a task of NSF to do is an interesting question. DOE has at their leadership class facilities support staff asigned to each project (at least for all INCITE projects) for this purpose. NERSC has just started support for preparing and porting 20 projects to their next platform (Cori). I am not familiar with the NSF efforts in this direction, but as the hardware gets more and more complicated, and as one has to worry about multi-level parallellism, data-locality, vectorization, etc., it is essential that not only the hardware is state-of-the-art, but also the software that runs on it. More support for domain scientists – whether from NSF or other sources – to efficiently use the new hardware is essential. Of course, domain scientists have to also realize that collaboration with computer scientists is critical in order to fully benefit from future supercomputing systems.

Need for diverse hardware for medium-size scale capacity computing

In addition to capability computing on the largest supercomputers available for open science (e.g. Blue Waters and Stampede (both NSF), or Mira and Titan (both DOE)) there is a clear need for more capacity computing of medium-size jobs. This can be done at petascale and future exascale facilities, but at least Mira and Titan are intended for capability computing – for jobs that need at least 20% of the machine. Capacity computing can be done more efficiently on supercomputers that fit those jobs better, without pushing both the technology and the software to the limits. Thus several few-petascale machines could very well serve the scientific community better than one 10 to 20 petascale machine.

Futhermore, different scientific problems require different types of supercomputers: some need a very high flop-rate, but not a lot of memory; other problems require a lot of memory, but not an extremely high flop-rate. By having several different systems, with different specs, different memory per core, different number of cores per node, etc., domain scientists are more likely to find access to a computer suitable for their problem at hand, in particular if the allocations are shared between different resources (like it is within XSEDE).

Finally, flash memory / SSDs attached to each node of a petascale system would be an excellent platform not only for big-data, but also for problems that require large amount of memory. Using big-data techniques, and flash memory at each node, we have been able to do 'out-of-core' calculations on relatively small systems, that would otherwise have required a much bigger system, with larger aggregate RAM. Extending the amount of flash memory per node may be more cost-efficient than increasing the system size and serving both the 'big-data' science community and science problems that require a lot of memory.