AsHES 2017 Keynote Talk

Accelerators and Exascale Systems: A programmer’s perspective

Tim Mattson, Intel

Abstract: Life as an HPC researcher used to be so easy. MPI between nodes and OpenMP on a node and you were done. GPUs came along and complicated things, but OpenCL and later the OpenMP device constructs gave us a portable way to write software for nodes composed of CPUs and GPUs. It was awkward but at least you could still build a base of software around portable, open standards.

Now with FPGAs and fixed function accelerators added to the mix, the ability to maintain an application from a single code base has become prohibitively difficult. On top of this, there is no consensus on how we will scale between nodes on an exascale system. Sure, there is always MPI but it’s not clear how the reliability needs of an exascale system will be addressed by MPI. Many new programming models have been suggested, but none of them have proven themselves to be “ready for prime time”. Things are quite confusing.

In this talk we will explore these issues from a programmer’s perspective. We will present a path forward and a workable solution, but in all honesty, there are still too many uncertainties to know the safest approach for an application programmer to take. Which ultimately is the real goal of this talk. We need to get the issues out on the table so we can effectively push the people who build our hardware to work with us to create software solutions that support the needs of our exascale-class applications.

Bio: Tim Mattson is an old-fashioned application programmer. Starting in the sciences (Ph.D. Chemistry, UCSC, 1985) he progressed to high performance computing where he worked with programming models (Linda, MPI, OpenMP, OpenCL, MPI+Fenix) and contributed to influential hardware/software co-design efforts (The first teraflop computing in 1997 and ten years later, the first teraflop chip). Tim is the PI for Intel’s Science and Technology center for Big Data. Among the projects at the center, he has been directly involved with an innovative array storage engine (TileDB), Linear algebra primitives for Graph algorithms (GraphBLAS), and Polystore Database management systems (BigDAWG).