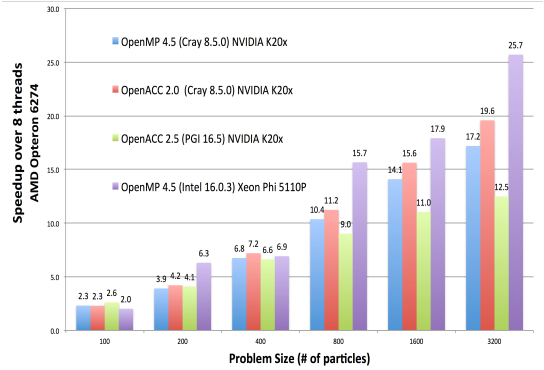
Evaluation of Directive-based Performance Portable Programming Models

Achievement: Developed an understanding of the performance portability of current directives-based (OpenACC and OpenMP) programming models on current HPC architectures.

Significance and Impact: This work demonstrates the current performance portability capabilities of OpenMP and OpenACC for a variety of algorithms on current HPC architectures, and suggests strategies for application developers to enhance the performance portability of their codes.

Research Details:

- Developed performance portable versions of both compute- and memory-bound kernels for GPUs and Intel manycore platforms using OpenMP and OpenACC
- Demonstrated successful strategies for performance portability, as well as shortcomings of current directive-based programming models.



(above) Performance of a CORAL benchmark kernel using various programming models and compilers on both GPUs and Intel Xeon PHI platforms.

Sponsor/Facility: Work was performed at ORNL, CSMD, and OLCF

PI and affiliation: M. Graham Lopez from CSMD– Oak Ridge National Laboratory

Team: M. Graham Lopez, Wayne Joubert, Veronica G. Vergara Larrea, Oscar Hernandez, Azzam Haidar, Stanimire Tomov, and Jack Dongarra

Publication: M. Graham Lopez, Wayne Joubert, Veronica G. Vergara Larrea, Oscar Hernandez, Azzam Haidar, Stanimire Tomov, and Jack Dongarra, *"Evaluation of Directive-based Performance Portable Programming Models"* Int. J. High Performance Computing and Networking (2017) (accepted)

Overview:

We present an extended exploration of the performance portability of directives provided by OpenMP 4 and OpenACC to program various types of node architectures with attached accelerators, both self-hosted multicore and offload multicore/GPU. Our goal is to examine how successful OpenACC and the newer offload features of OpenMP 4.5 are for moving codes between architectures, and we document how much tuning might be required and what lessons we can learn from these experiences. To do this, we use examples of algorithms with varying computational intensities for our evaluation, as both compute and data access efficiency are important considerations for overall application performance. To better understand fundamental compute vs. bandwidth bound characteristics, we add the compute-bound Level 3 BLAS GEMM kernel to our linear algebra evaluation. We implement the kernels of interest using various methods provided by newer OpenACC and OpenMP implementations, and we evaluate their performance on various platforms including both x86 64 and Power8 with attached NVIDIA GPUs, X86 64 multicores, self-hosted Intel Xeon Phi KNL, as well as an X86 64 host system with Intel Xeon Phi coprocessors. We update these evaluations with the newest version of the NVIDIA Pascal architecture (P100), Intel KNL 7230, Power8+, and the newest supporting compiler implementations. Furthermore, we present in detail what factors affected the performance portability, including how to pick the right programming model, its programming style, its availability on different platforms, and how well compilers can optimize and target multiple platforms.