An in-depth evaluation of GCC’s OpenACC implementation on Cray systems

Achievement: First comprehensive evaluation of OpenACC support in the GCC compiler suite

Significance and Impact: Wide support for the OpenACC directives in HPC compiler is crucial for the continued and expanded adoption of the directive to offload computations to the GPU. The widely used, open source GCC compiler suite has recently gained support for kernel offloading using version 2.0 of the OpenACC 2.0 specification. A deeper understanding of the capabilities and limitations of this support is crucial to guide the use of the compiler suite, and guide future development to address any performance and/or functionality shortcomings in the current implementation.

Research Details:
• This work was done in collaboration with the main developers of OpenACC support in GCC (MentorGraphics)
• The compiler suite was applied to various computational kernels and performance was compared against more mature implementations of the specification.
• The work presented details on the implementation strategy of OpenACC support in GCC, which informs the developers as to the best strategy to extract maximum performance from their computational kernels.

Sponsor/Facility: DOE-OLCF

PI and affiliation: Verónica G. Vergara Larrea (ORNL)
Team: Wael R. Elwasif (ORNL), Oscar Hernandez (ORNL), Cesar Philippidis (Mentor Graphics), Randy Allen (Mentor Graphics)

Overview:
OpenACC is a directive-based API that extends the C/C++ and Fortran base languages to program accelerators and multicores. Several commercial implementations are available that support OpenACC including PGI, Cray, and PathScale. More recently, GCC started adding support for OpenACC and is expected to fully support the OpenACC 2.0 specification in the upcoming GCC 7 release. However, to our knowledge, the quality and performance of GCC’s OpenACC implementation have not been studied in detail. In this work, we perform and in-depth evaluation of GCC’s OpenACC implementation on Titan, ORNL’s Cray XK7 supercomputer, and compare it to other commercially available compiler implementations. We first provide a description of the OpenACC implementation design in GCC, its runtime, as well as provide an overview of the current state of OpenACC supported features as described in GCC 6.x. Then, we evaluate the quality and performance of the GCC 6.x implementation by using the OpenACC Verification and Validation suite to test the accuracy and correctness of the implementation, the EPCC OpenACC benchmark suite to measure performance, and the SPEC ACCEL benchmark OpenACC suite to exercise the implementation. We believe that the results presented in this study will be useful for the larger community interested in using and evaluating new OpenACC implementations.

Papers: