GA

TN

A CRITICAL EVALUATION OF THE OP2/OPS PARALLEL MESHING AND CODE GENERATION SOFTWARE

Allocation: Industry/8 Knh
PI: Kevin Olson[1]
Co-PIs: Reid Atcheson[1], Edward Smyth[1], Mike Dewar[1]

[1]Numerical Algorithms Group

EXECUTIVE SUMMARY

OPS/OP2 is a software package designed to construct algorithms that are easier to implement, maintain, and perform well on different computer architectures, including symmetric multiprocessing (SMP), clusters and clusters of graphics processing units (GPUs). This is accomplished by a programming model where the user defines stencils for performing operations, defining data structures, defining the data and how it is stored on the data structures, and constructing parallel kernels that operate on the data. Once these steps are completed, OPS and OP2 will construct underlying code for different parallel programming paradigms that include sequential, MPI, OpenMP, CUDA, MPI+OpenMP, and MPI+CUDA. Both C/C++ and Fortran programming languages are currently supported by OPS and OP2.

RESEARCH CHALLENGE

The challenge of this research was to test the OPS/OP2 code generation software to determine how easy it is to use and if it can generate code that scales well on large, parallel computer architectures.

METHODS & CODES

OPS/OP2 is software for constructing parallel computer programs [1,2]. OPS focuses on algorithms using structured meshes while OP2 is used for unstructured meshes. Using OPS/OP2 code involves writing either Fortran or C/C++ source code that embeds OPS/OP2 function calls that abstract out the parallelism in the algorithm. A code generation script is then run on the user-written code that generates source code for different parallel programming paradigms that include MPI, OpenMP, and CUDA as well as any combination of them.

RESULTS & IMPACT

OPS was used to construct a finite volume, compressible hydrodynamics code. OP2 was used to construct a Laplace equation solver using an unstructured mesh. The main result was that OPS and OP2 can produce code that runs using all the parallel programming paradigms listed above. Scaling was good in all cases, but MPI-“only” code gave the best performance. Using OPS and OP2 requires an investment in learning its syntax and use.

OPS and OP2 could be very useful tools for developing parallel algorithms since it would ease the amount of maintenance work for developing parallel algorithms. We note that OPS and OP2 are continuing to be developed and have evolved several versions ahead of those used for this study.

Whether OPS/OP2 can be used in further work at Numerical Algorithms Group is still under evaluation.

WHY BLUE WATERS

Blue Waters was chosen because Numerical Algorithms Group is one of its corporate partners and Blue Waters has the right combination of hardware available: a large number of SMP and GPU nodes that allowed us to perform the required scaling studies to evaluate OPS and OP2.