TN

ALGORITHMS FOR EXTREME-SCALE SYTEMS

Allocation: Blue Waters Professor/40 Knh
PI: William Gropp[1]
Students: Paul Eller[1], Ed Karrels[1]

[1]University of Illinois at Urbana–Champaign

EXECUTIVE SUMMARY

Continued increases in the performance of large-scale systems will come from greater parallelism at all levels. At the node level, this is seen both in the increasing number of cores per processor and the use of large numbers of simpler computing elements in general-purpose computing on GPUs. The largest systems must network tens of thousands of nodes together to achieve the performance required for the most challenging computations.

Successfully using these systems requires new algorithms. In the last year, the research team developed a simple yet complete implementation of the Message-Passing Interface Cartesian topology routines that significantly outperforms the available implementations, as well as further developing performance models and implementations for highly scalable Krylov methods for the solution of large linear systems.

RESEARCH CHALLENGE

This work directly targets current barriers to effective use of extreme-scale systems by applications. For example, Krylov methods such as Conjugate Gradient are used in many applications currently being run on Blue Waters and other large-scale systems. These methods depend both on high-performance matrix–vector products, which are communication intensive, and on collective all-reduce operations, which introduce synchronizations that can limit scalability. Developing and demonstrating a more scalable version of this algorithm would immediately benefit those applications.

The research team's approach begins with developing a performance model that captures the key aspects of the intra- and internode communication costs and uses that model to inform the development of new algorithms. This approach has also yielded improved parallel I/O routines and a better implementation of a process placement operation that can improve the performance of applications.

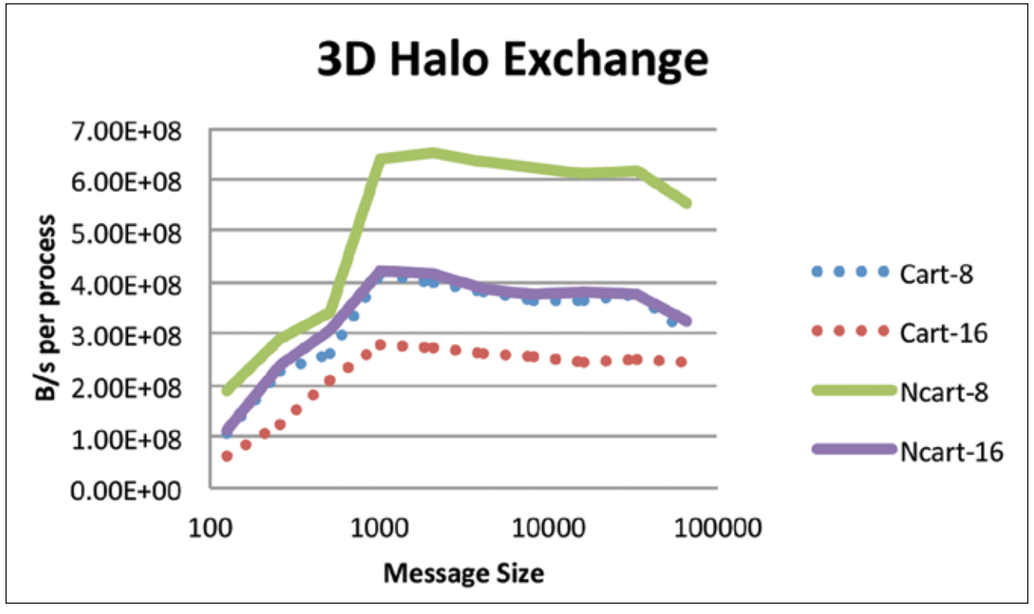

Figure 1: Communication performance for a 3D halo exchange on Blue Waters using the vendor-provided implementation of MPI_Cart_create (dotted lines) and the research group's node-aware version (solid lines). The "-8" versions use an 8 x 8 x 8 process mesh (512 processes); the "-16" versions use a 16 x 16 x 16 process mesh (4,096 processes).

METHODS & CODES

To address these challenges, the research team has developed several performance models that address limitations in off-node communication bandwidth, message matching costs, network contention, and the effect of system "noise." Benchmarks to test these performance models have been developed, and experiments have been conducted with some applications. These performance models have led to new algorithms for providing good and efficient implementation of process mapping for regular meshes. Some of the codes are open source and available; the routines for process mapping in regular Cartesian topologies are under consideration for inclusion in the MPICH implementation of the Message-Passing Interface (MPI).

RESULTS & IMPACT

Paul Eller, working with PI William Gropp, has been using Blue Waters over the last year for an investigation into performance modeling of scalable Krylov solvers for structured grid problems. This includes developing code for measuring and processing parallel runtimes and network performance counters, developing a collection of kernels relevant to structured grid problems, exploring their use in several applications, and developing performance models with penalty terms that accurately model parallel performance at scale. He has run experiments to determine parameters for the performance models and performed scaling studies for the various parallel communication kernels and scalable conjugate gradient solvers. These results have led to a paper submitted to the 2019 Association for Computing Machinery International Conference on Supercomputing. These experiments have helped to better understand Krylov solver performance at scale, to develop more accurate performance models, and to optimize the solvers to obtain better performance.

Ed Karrels, also working with William Gropp, has continued to explore parallel I/O performance, including exploring the sensitivity of several parameters for I/O operations, such as stripe count and size, and the number of processes per node performing I/O operations. The goal is to provide better guidance for users in choosing I/O parameters and to develop more automatic methods for selecting these parameters within parallel I/O libraries. The libraries that he developed in the previous year—MeshIO, zlines, and zchunk—remain available for use by applications.

In addition, William Gropp has developed a new algorithm for implementing process mapping for Cartesian grids, which is needed for many applications that use structured grids. MPI provides a convenient routine for this operation, but few MPI libraries provide a good implementation of this operation. As a result, applications must either forgo the performance or use ad hoc, nonportable techniques to achieve a good mapping. Such tools do exist for Blue Waters, but these are not portable to other systems and often fail to provide good process mappings in practice. Applications should be able to rely on the features in MPI and not need to use nonstandard, nonportable methods. Finally, by using insight gained from the team's new performance model, they developed an alternative implementation of MPI_Cart_create that provides a significant performance benefit compared to the vendor-supplied implementation, as shown in Fig. 1.

WHY BLUE WATERS

Scalability research relies on the ability to run experiments at large scale, requiring tens of thousands of nodes and hundreds of thousands of processes and cores. Blue Waters provides one of the few available environments where such large-scale experiments can be run. In addition, Blue Waters provides a highly capable I/O system, which the research team used in developing improved approaches to extreme-scale I/O.

PUBLICATIONS & DATA SETS

W. D. Gropp, "Using node and socket information to implement MPI Cartesian topologies," *Parallel Comput.*, vol. 85, pp. 98–108, July 2019.

A. Bienz, W. D. Gropp, and L. N. Olson, "Node aware sparse matrix–vector multiplication," *J. Parallel Distrib. Comput.*, vol. 130, pp. 166–178, Aug. 2019.

W. D. Gropp, "Using node information to implement MPI Cartesian topologies," in *Proc. 25th Eur. MPI Users' Group Meeting*, Barcelona, Spain, Sept. 23–26, 2018, pp. 1–9.

Both zchunk and zlines are available at https://github.com/oshkosher/bioio.

MeshIO is available at https://github.com/oshkosher/meshio.

The improved implementation of MPI_Cart_create is part of baseenv and is available from wgropp@illinois.edu.