TN

ALGORITHMS FOR EXTREME SCALE SYSTEMS

Allocation: Blue Waters Professor/80 Knh
PI: William Gropp[1]
Collaborator: Luke Olson[1]
Students: Amanda Bienz[1], 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, we see this both in the increasing number of cores per processor and the use of large numbers of simpler computing elements in general purpose 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, we have further improved a new communication model that better fits the performance of multicore nodes to develop new algorithms for sparse matrix–vector products and better understand the behavior of nonblocking algorithms for the Conjugate Gradient method. We also developed a simple implementation of the MPI Cartesian topology routines that significantly outperforms the available implementations and several parallel I/O libraries.

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. 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. Our 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.

METHODS & CODES

To address these challenges, we have developed several performance models that address limitations in off-node communication bandwidth, message matching costs, network contention, and the effect of system "noise." We have developed benchmarks to test these performance models and conducted experiments with some applications. Some of the codes are open source and freely available.

RESULTS & IMPACT

Over the last year, we made progress in four areas.

Amanda Bienz, working with Luke Olson and William Gropp, has looked at the communication cost of irregular point-to-point communications in sparse matrix operations, with a focus on the operations needed for algebraic multigrid (AMG) methods. They are investigating methods for improving the postal and max-rate models to account for the main costs associated with point-to-point communication. They added node-aware parameters to the max-rate model, distinguishing the large differences in cost among intrasocket, intersocket but intranode, and internode messages. Furthermore, they added a quadratic queue search penalty to accurately model the cost of sending multiple messages and a network contention penalty to account for contention of links in the network when bytes are communicated across a large number of links. Fig. 1 shows the measured versus modeled cost of communication throughout sparse matrix operations on various levels of algebraic multigrid hierarchies. The results show the importance of including queue search times as well as contention effects in the network.

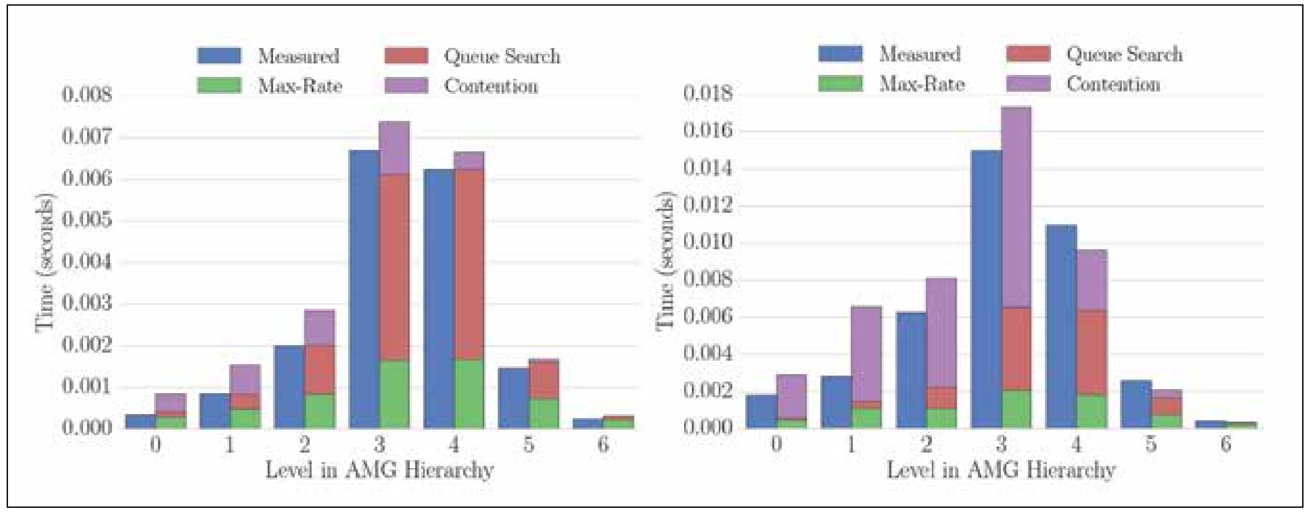

Figure 1: Measured versus modeled costs for communication during sparse matrix-vector multiplication (left) and sparse matrix-matrix multiplication (right) on each level of an algebraic multigrid hierarchy for unstructured linear elasticity. From Improving Performance Models for Irregular Point-to-Point Communication, submitted to *EuroMPI* (2018).

Paul Eller, working with 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, and developing performance models with penalty terms that accurately model parallel performance at scale. Eller 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. He is currently designing and running experiments with optimizations designed to improve the performance of these kernels on Blue Waters. He has also done some initial work on studies to investigate the impact of network noise on solver performance and to investigate the performance of scalable Krylov solvers within a quantum chromodynamics application. 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, working with William Gropp, has been testing a variety of input/output access patterns and developing tools for improving input/output performance. These tools include:

- MeshIO—a tool for reading and writing N-dimensional meshes in parallel. This is being used by the XPACC project, sponsored by the U.S. Department of Energy, to accelerate job start-up and shut-down, and is being evaluated by two other science teams.
- Zlines—a tool for reading compressed text data at arbitrary line offsets efficiently. This is being used by bioinformatics researchers on genomic data.
- Zchunk—a tool for reading compressed binary data at arbitrary offsets efficiently. This is not currently being used by other projects.

Karrels also prepared a tutorial on input/output best practices for Blue Waters.

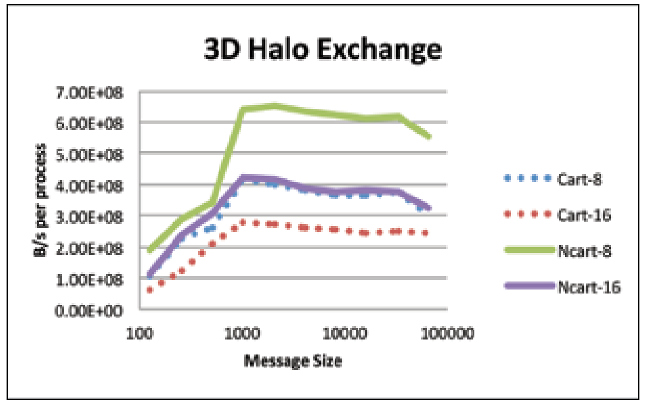

Figure 2: Results for simple halo exchange for a 3D mesh on up to 4,000 processes on Blue Waters. "Cart-8" shows the performance for the Cray MPI for an 8×8×8 process mesh; "Ncart-8" shows the performance of our replacement for the same mesh. From Using Node Information to Implement MPI Cartesian Topologies, submitted to *EuroMPI* (2018).

Finally, 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 they do not provide the right solution to the problem. Applications should be able to rely on the features in MPI and not need to use nonstandard, nonportable methods.

In addition, by using insight gained from our new performance model, we developed an alternative implementation of MPI_Cart_create that provides a significant performance benefit, as shown in Fig. 2.

WHY BLUE WATERS

Scalability research relies on the ability to run experiments at large scale and requires 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 we used in developing improved approaches to extreme-scale I/O.

PUBLICATIONS & DATA SETS

Bienz, A., W. Gropp, and L. Olson, Improving Performance Models for Irregular Point-to-Point Communication. Submitted to *EuroMPI* (2018).

Gropp, W., Using Node Information to Implement MPI Cartesian Topologies. Submitted to *EuroMPI* (2018).

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.