PUSHING THE BOUNDARIES OF LARGE-SCALE TENSOR COMPUTATIONS

Allocation: Illinois/20 Knh
PI: Edgar Solomonik[1]

[1]University of Illinois at Urbana–Champaign

EXECUTIVE SUMMARY

This project seeks to develop new parallel algorithms and scalable productive software for matrix and tensor computations. Over the past year, the research team has made significant advances in software infrastructure of the Cyclops library for tensor computations. This library provides a productive algebraic programming interface in Python and C++ that performs numerical and combinatorial operations in a data-distributed manner. The team has developed support for hypersparse matrix representations, automatic optimization of contraction order, parallel tensor-times-tensor-product kernels, and has significantly extended capabilities at the Python level. The team currently is benchmarking tensor completion algorithms on Blue Waters using the new Python interface layer and has obtained preliminary benchmarking results for a new Cyclops application in genomic analysis. Separately, the researchers have developed a new practical communication-avoiding parallel algorithm for QR factorization, evaluating its performance via large-scale runs on Blue Waters.

RESEARCH CHALLENGE

Tensor computations are a growing area in computational science, with applications in quantum chemistry and physics, quantum circuit simulation, machine learning, optimization, and numerical PDEs (partial differential equations). They also push the boundaries of numerical linear algebra technologies, requiring algebra on large, extremely sparse, and in some cases unstructured matrices, as well as redistribution of their data. These applications create a demand for better algorithms and software for distributed-memory tensor computations.

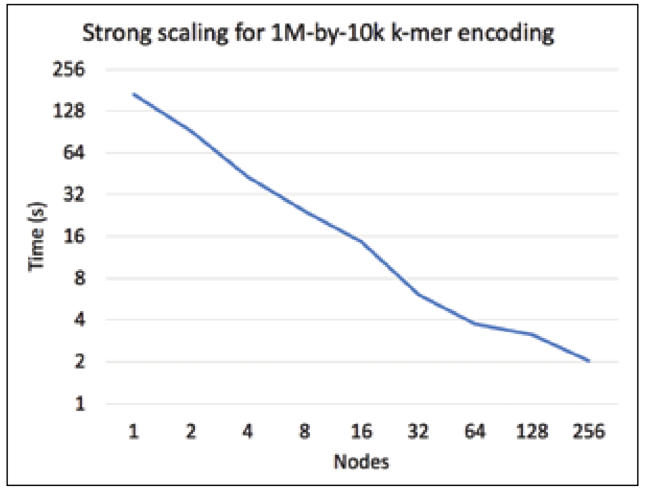

Figure 1: Strong scaling on Blue Waters of a computational biology application using specialized parallel sparse matrix multiplication with Cyclops.

METHODS & CODES

The research group led the development of Cyclops, a distributed-memory library for tensor computations, which is likely the most widely used distributed tensor algebra library. Cyclops uses MPI, OpenMP, CUDA, and HPTT (High-Performance Tensor Transpose). It interoperates with ScaLAPACK and makes use of advanced sparse matrix routines in MKL. Cyclops provides a much simpler and more intuitive interface for both that is almost entirely agnostic to sparsity. It also provides a wide range of further functionality, including sparse and dense tensor contractions and generalized elementwise operations that enable graph and combinatorial algorithms. Additionally, it provides an interface to Python and a wide variety of functionality in a style similar to the NumPy library of mathematical functions.

The group has also developed standalone codes for parallel numerical linear algebra kernels. The researchers have developed and maintained suites of algorithms for tensor completion and tensor decomposition, which employ variants of alternating least squares, coordinate descent, and stochastic gradient descent methods, all parallelized using Cyclops. The team planned to benchmark the newly developed tensor decomposition and tensor completion kernels on Blue Waters in the summer of 2019.

Aside from tensor computations, the group also works on improving basic numerical linear algebra operations, such as computation of dense QR and eigenvalue decomposition of symmetric matrices. Edward Hutter, a Ph.D. student at the University of Illinois at Urbana–Champaign (Illinois), developed and maintains a new QR algorithm and library that achieves asymptotic improvements in communication efficiency by a new parallelization for the Cholesky–QR2 algorithm. One application of such large rectangular QR factorizations is in computing the Tucker decomposition of tensors.

RESULTS & IMPACT

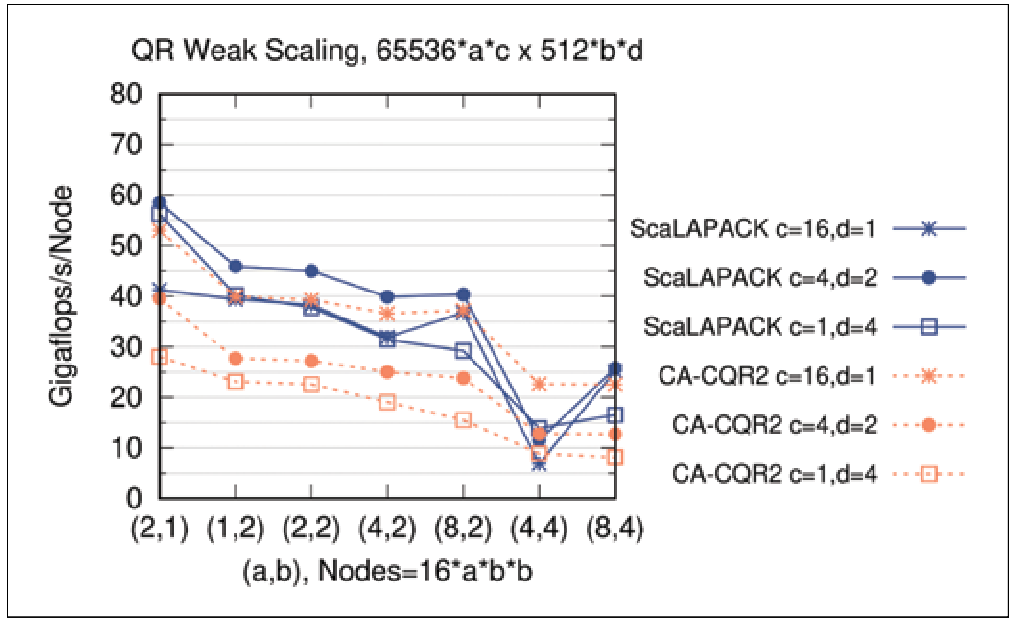

Figure 2: Weak scaling on Blue Waters of a new parallel QR factorization for various matrix sizes compared to ScaLAPACK.

The team's work on Cyclops has a leading role in parallel infrastructure for numerical tensor algebra. Their library enables distributed-memory parallelism in leading quantum chemistry codes, including QChem, PySCF, and CC4S. The library has been used to execute methods for quantum chemistry at near one petaflop/s and was used by a group of IBM and Lawrence Livermore National Laboratory researchers to perform a 49-qubit quantum circuit simulation, a central result in the field of quantum supremacy testing.

The research team is currently involved in collaborations with researchers in the physics department at Illinois and chemistry department at Caltech to develop the first massively parallel versions of tensor network codes. One-dimensional and two-dimensional tensor networks (namely DMRG and PEPS, two methods for computing properties such as ground state energy of quantum systems, which represent the quantum many-body state via 1-D and 2D tensor networks, respectively) provide highly accurate results for ground and excited state properties of highly correlated quantum systems. Researchers in the Illinois physics group (led by Bryan Clark) are currently prototyping initial versions of DMRG on Blue Waters.

A major impact on applications provided by Cyclops is the ability to rapidly develop massively parallel code via tensor algebra. A very recent example of this is the research group's collaboration with a team of bioinformatics researchers who are interested in computing a Jaccard similarity matrix from the data of a large set of genomes, the first calculation of this type. Using Cyclops primitives for general sparse matrix multiplication, the team implemented the necessary kernels for logical (bitwise) operations within a week and are seeing good weak and strong scaling in small-scale initial tests on Blue Waters (Fig. 1). These results should pave the way for the bioinformatics team to apply for resources to perform a full-scale computation.

The team has obtained large-scale results for the new Cholesky–QR2 parallel algorithm, which achieves better parallel scaling trends than ScaLAPACK's QR owing to needing less communication, but it is slightly behind in absolute performance because it requires more FLOPS (Fig. 2). The team has also obtained results on more compute-intensive architectures, where the new Cholesky–QR2 algorithm outperforms ScaLAPACK significantly on large node-counts. The researchers are currently developing a GPU-accelerated version of the Cholesky–QR2 algorithm that they believe will be effective on the GPU nodes of Blue Waters as well as on future architectures. QR is one of the most widely used dense linear algebra primitives, so this work has the potential to impact many applications. These results will appear in the proceedings of the 2019 International Parallel and Distributed Processing Symposium.

WHY BLUE WATERS

As Illinois researchers, the team takes pride in using and showcasing results from the Blue Waters high-performance computing infrastructure. They also generally aim to test new algorithms and software on multiple supercomputing architectures, and Blue Waters is both unique and is itself diverse (providing large infrastructure for both GPU and pure-CPU runs). Using Blue Waters provided the team with a better understanding of the dependency of the performance of Cholesky–QR2 for architectures with different ratios of bandwidth and compute rate. In a number of application areas of interest, including quantum chemistry, quantum circuit simulation, tensor decomposition, and bioinformatics, the capability of calculations is often bounded by memory, making Blue Waters the architecture of choice.

PUBLICATIONS & DATA SETS

E. Hutter and E. Solomonik, "Communication-avoiding Cholesky-QR2 for rectangular matrices," in *Proc. of the IEEE Int. Parallel and Distributed Processing Symp.*, Rio de Janeiro, Brazil, May 20–24, 2019.