TN

CI

PERFORMANCE EVALUATION OF NEW ALGEBRAIC ALGORITHMS AND LIBRARIES

Allocation: Illinois/50 Knh
PI: Edgar Solomonik[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

High-performance parallel algorithms for numerical linear algebra play a crucial role in most large-scale computational science problems. We have made progress in two aspects of advancing these types of algorithms: (1) introduction and tuning of distributed-memory functional abstractions for tensor operations, and (2) development of new communication-avoiding algorithms for matrix factorizations. Blue Waters has allowed us to deploy and evaluate these new methods on a leadership-class computing platform. In particular, we have done performance studies on the use of distributed symmetric tensor contractions for the atomic-to-molecular orbital transformation (a key kernel in quantum chemistry computations), of parallel sparse matrix multiplication routines and their use in graph analytics (betweenness centrality), and are currently evaluating a novel communication-avoiding algorithm for QR factorization of rectangular matrices.

RESEARCH CHALLENGE

The biggest challenge facing the parallel scalability of methods in computational science is the overhead of moving data between processors. Our goal is to develop algorithms and libraries that minimize communication in the number of messages as well as in the amount of data moved. To do so in the most useful way, we leverage the ubiquity of numerical linear algebra in scientific computing, targeting the development of algebraic algorithms and libraries. Matrices and tensors (multidimensional matrices) provide high-level abstractions for data sets and transformations thereof. Our aim is to push to the limit the parallel scalability of fundamental computational kernels on matrices and tensors by formulating algorithms that have a provably minimal communication complexity, and evaluating them in large-scale execution on Blue Waters.

METHODS & CODES

Our work does not only study hypothetical algorithms but contributes directly to libraries that are available to application developers using Blue Waters or other supercomputing platforms. In particular, our research has focused on extending and tuning Cyclops Tensor Framework (CTF). This library provides distributed-memory support for sparse and dense tensors, automatically mapping contractions and other functions on these distributed data sets. CTF uses performance models to make runtime mapping decisions, using autotuning to train the model parameters. After performing tuning at scale on a training suite of CTF driver-routines, we studied performance of two kernels from radically different domains: quantum chemistry and graph analysis.

RESULTS & IMPACT

We conducted a performance study of in-memory and out-of-core CTF versions of an atomic-to-molecular orbital (AO-MO) transformation, immediately showing scalability on problems that are comparable in scale to the largest previously executed. This transformation appears in many high-accuracy quantum chemistry methods and is the most expensive step in some newly proposed methods. This work served to provide preliminary results for a many-principal investigator interdisciplinary proposal focusing on catalysis in chemical reactions.

Additionally, we evaluated the performance of a betweenness centrality code—MFBC—that leverages sparse matrix multiplication functionality in CTF as well as its support for user-defined tensor element-types and functions. We were able to calculate centrality scores for some of the largest graphs publicly available, including the Friendster graph, which has 1.8 billion edges, leading to a paper in *Supercomputing '17*. Fig. 1 displays the parallel scalability of the CTF MFBC code in terms of millions of edge traversals per second.

Over the past year, we have also developed a new parallel algorithm for QR factorization as well as a parallel implementation thereof. The algorithm aims to realize a QR code that achieves optimal communication and synchronization complexity in theory and is efficient in practice. While the new algorithm is not asymptotically more efficient than the state of the art, it is substantially more simple and easier to implement (no algorithm with the same communication complexity has been implemented previously). The basic idea is to use the Cholesky–QR2 algorithm and leverage communication-optimal parallel Cholesky and matrix multiplication routines. The Cholesky–QR2 algorithm is numerically stable, so long as the matrix is reasonably well-conditioned. We plan to compare the performance of our implementation to the QR routine in ScaLAPACK and test its stability on very large matrices.

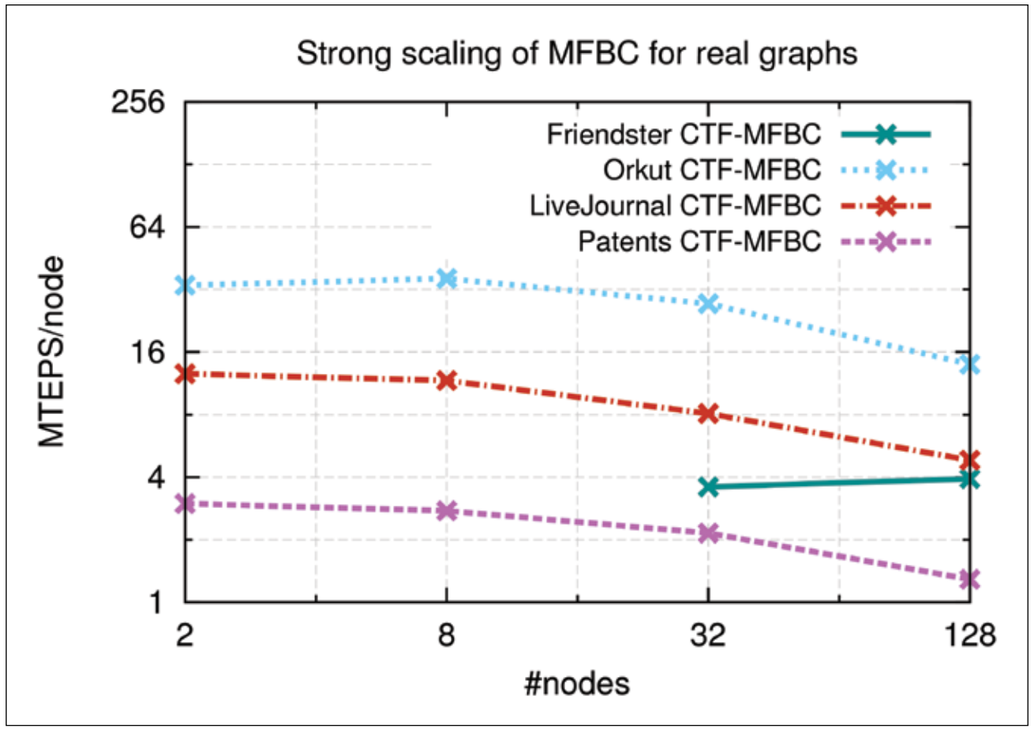

Figure 1: Parallel scalability of betweenness centrality using sparse matrix multiplication for massive graphs.

WHY BLUE WATERS

As our goal is to design software and algorithms for applications running on supercomputing systems, access to Blue Waters is essential for testing and evaluation. While all of the codes developed are designed to be portable, demonstrating performance on Blue Waters will help foster local collaborations and deployment of parallel numerical library software.

PUBLICATIONS AND DATA SETS

Solomonik, E., M. Besta, F. Vella, and T. Hoefler, Scaling betweenness centrality using communication-efficient sparse matrix multiplication. *ACM/IEEE Supercomputing Conference*, Denver, Colo., November 12–17, 2017.

Cyclops Tensor Framework: distributed-memory tensor algebra data sets, https://github.com/solomonik/ctf.