SANDIA REPORT

SAND2016-6204
Unlimited Release
Printed June, 2016

Reducing Computation and Communication in Scientific Computing: Connecting Theory to Practice

Grey Ballard

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2016-6204
Unlimited Release
Printed June, 2016

Reducing Computation and Communication in Scientific Computing: Connecting Theory to Practice

Grey Ballard

Abstract

This report summarizes the work produced as part of a Truman Fellowship appointment and its associated LDRD project. The overall goal of the project was to develop better algorithms and implementations for key computational kernels within the field of scientific computing by designing them to be communication efficient, moving as little data as possible. The primary problem of interest was dense matrix multiplication; other computations that were addressed include sparse matrix-matrix multiplication, QR factorization, solving symmetric linear systems, and the symmetric eigendecomposition. The project also involved the study of computations at the intersection of scientific computing and data analysis, including nonnegative matrix factorization for discovering latent factors, Tucker tensor decomposition for data compression, and sampling methods for similarity search.

Acknowledgment

Grey Ballard gratefully acknowledges the Truman selection committee for granting him the unique opportunity and resources to lead a three-year research project. Grey also thanks Tammy Kolda for providing many years of invaluable mentorship, dating back to his summer internship experiences. Grey also acknowledges Susanna Gordon, Karim Mahrous, Jim Costa, Heidi Ammerlahn, and Len Napolitano for their managerial support during his tenure as a Truman Fellow.

Finally, Grey would like to thank the many coauthors with whom he had the pleasure of working over the past three years: Woody Austin, Ariful Azad, Dulceneia Becker, Austin Benson, Aydın Buluç, Erin Carson, Jim Demmel, Jack Dongarra, Alex Druinsky, Laura Grigori, Mark Hoemmen, Jonathan Hu, Mathias Jacquelin, Ramakrishnan Kannan, Nicholas Knight, Tammy Kolda, Benjamin Lipshitz, Hong Diep Nguyen, Haesun Park, Inon Peled, Ali Pinar, Oded Schwartz, C. Seshadri ("Sesh"), Chris Siefert, Edgar Solomonik, Sivan Toledo, Samuel Williams, and Ichitaro Yamazaki.

This research was supported in part by an appointment to the Sandia National Laboratories Truman Fellowship in National Security Science and Engineering, sponsored by Sandia Corporation (a wholly owned subsidiary of Lockheed Martin Corporation) as Operator of Sandia National Laboratories under its U.S. Department of Energy Contract No. DE-AC04-94AL85000.

1 Introduction

The gap between the peak capabilities of computer hardware and the achieved performance of numerical computations is caused in large part by the high cost of communication, i.e., the movement of data between processors and throughout the memory hierarchy of a single processor. "Standard" $O(n^3)$ matrix multiplication is the most fundamental dense matrix computation, and communication-optimal algorithms exist that have been heavily tuned on most architectures to attain high performance. "Fast" $O(n^{2.81})$ matrix multiplication algorithms have been identified for over 40 years and are starting to become practical as communication costs dominate. The primary objective of this research has been to use computer-aided search to find a matrix multiplication algorithm that is both theoretically and experimentally faster than current implementations (i.e., $O(n^p)$ with $p < 2.81$). The secondary objective is to pursue several complementary projects that involve developing other communication-optimal algorithms. This work was completed as part of a Truman Fellowship appointment.

This report summarizes the work associated with the project. Papers related to the primary objective (fast matrix multiplication) are described in Sections 2 and 3. Overall, the goal of finding a practical algorithm with better exponent than 2.81 was achieved, but the performance improvement of that particular algorithm was minimal for reasonably sized matrices; however, other fast algorithms with comparable exponents were shown to give practical improvements for non-square matrix multiplications. In particular, much of the early work of the project focused on techniques for searching for new algorithms; a major conclusion from that effort was that combinatorial methods can be successful for very small base cases but could not scale up to base cases that would yield new algorithms. Numerical techniques have been much more effective; in fact, the publication of external work by Alexey Smirnov [15] during the first year of the project presented the best known method to date. Using his approach, we were able to reproduce many of the results, and we developed code generation tools to streamline the process from algorithm discovery to high-performance implementation. We have also analyzed the numerical properties of the algorithms and devised techniques to improve the errors inherent to fast algorithms. The search for fast, practical algorithms continues, and the most promising avenue for future progress seems to be exploiting the special structure of the matrix multiplication tensor.

The remaining sections describe publications related to the secondary objective. In particular, Section 4 describes a survey article written at the beginning of the project, and Sections 5, 6, 7, and 8 describe communication-avoiding algorithms for various dense linear algebra computations, including QR factorization, solving symmetric linear systems, and the symmetric eigendecomposition. Sections 9, 10, and 11 focus on the multiplication of two sparse matrices on parallel machines, developing the theory of communication efficiency of algorithms, studying its use within a particular application, and producing a parallel implementation for general inputs. Finally, Sections 12, 13, and 14 address problems in the field of data analysis; each approach uses techniques from communication-avoiding linear algebra and applies them to particular computations for analyzing big data sets, including nonnegative matrix factorization, Tucker decompositions of tensors, and similarity search.

2 A Framework for Practical Parallel Fast Matrix Multiplication

Matrix multiplication is a fundamental computation in many scientific disciplines. In a paper presented at PPoPP 2015 [13], we show that novel fast matrix multiplication algorithms can significantly outperform vendor implementations of the classical algorithm and Strassen's fast algorithm on modest problem sizes and shapes. Furthermore, we show that the best choice of fast algorithm depends not only on the size of the matrices but also the shape. We develop a code generation tool to automatically implement multiple sequential and shared-memory parallel variants of each fast algorithm, including our novel parallelization scheme. This allows us to rapidly benchmark over 20 fast algorithms on several problem sizes. Furthermore, we discuss a number of practical implementation issues for these algorithms on shared-memory machines that can direct further research on making fast algorithms practical.

3 Improving the Numerical Stability of Fast Matrix Multiplication Algorithms

Fast algorithms for matrix multiplication, namely those that perform asymptotically fewer scalar operations than the classical algorithm, have been considered primarily of theoretical interest. Apart from Strassen's original algorithm, few fast algorithms have been efficiently implemented or used in practical applications. However, there exist many practical alternatives to Strassen's algorithm with varying performance and numerical properties. Fast algorithms are known to be numerically stable, but because their error bounds are slightly weaker than the classical algorithm, they are not used even in cases where they provide a performance benefit. We argue in a paper submitted to *SIMAX* [4] that the numerical sacrifice of fast algorithms, particularly for the typical use cases of practical algorithms, is not prohibitive, and we explore ways to improve the accuracy both theoretically and empirically. The numerical accuracy of fast matrix multiplication depends on properties of the algorithm and of the input matrices, and we consider both contributions independently. We generalize and tighten previous error analyses of fast algorithms and compare their properties. We discuss algorithmic techniques for improving the error guarantees from two perspectives: manipulating the algorithms, and reducing input anomalies by various forms of diagonal scaling. Finally, we benchmark performance and demonstrate our improved numerical accuracy.

4 Communication Lower Bounds and Optimal Algorithms for Numerical Linear Algebra

The traditional metric for the efficiency of a numerical algorithm has been the number of arithmetic operations it performs. Technological trends have long been reducing the time to perform an arithmetic operation, so it is no longer the bottleneck in many algorithms; rather, communication, or moving data, is the bottleneck. This motivates us to seek algorithms that move as little data as possible, either between levels of a memory hierarchy or between parallel processors over a network. In a survey paper appearing in *Acta Numerica* [5], we summarize recent progress in three aspects of this problem. First we describe lower bounds on communication. Some of these generalize known lower bounds for dense classical ($O(n^3)$) matrix multiplication to all direct methods of linear algebra, to sequential and parallel algorithms, and to dense and sparse matrices. We also present lower bounds for Strassen-like algorithms, and for iterative methods, in particular Krylov subspace methods applied to sparse matrices. Second, we compare these lower bounds to widely used versions of these algorithms, and note that these widely used algorithms usually communicate asymptotically more than is necessary. Third, we identify or invent new algorithms for most linear algebra problems that do attain these lower bounds, and demonstrate large speed-ups in theory and practice.

5 Communication-Avoiding Symmetric-Indefinite Factorization

In a paper appearing in *SIMAX* [3], We describe and analyze a novel symmetric triangular factorization algorithm. The algorithm is essentially a block version of Aasen's triangular tridiagonalization. It factors a dense symmetric matrix A as the product $A = PLTL^TP^T$, where P is a permutation matrix, L is lower triangular, and T is block tridiagonal and banded. The algorithm is the first symmetric-indefinite communication-avoiding factorization: it performs an asymptotically optimal amount of communication in a two-level memory hierarchy for almost any cache-line size. Adaptations of the algorithm to parallel computers are likely to be communication efficient as well; one such adaptation has been recently published. The current paper describes the algorithm, proves that it is numerically stable, and proves that it is communication optimal.

6 Avoiding Communication in Successive Band Reduction

The running time of an algorithm depends on both arithmetic and communication (i.e., data movement) costs, and the relative costs of communication are growing over time. In a paper appearing in *ACM TOPC* [8], we present sequential and distributed-memory parallel algorithms for tridiagonalizing full symmetric and symmetric band matrices that asymptotically reduce communication compared to previous approaches. The tridiagonalization of a symmetric band matrix is a key kernel in solving the symmetric eigenvalue problem for both full and band matrices. In order to preserve structure, tridiagonalization routines use annihilate-and-chase procedures that previously have suffered from poor data locality and high parallel latency cost. We improve both by reorganizing the computation and obtain asymptotic improvements. We also propose new algorithms for reducing a full symmetric matrix to band form in a communication-efficient manner. In this article, we consider the cases of computing eigenvalues only and of computing eigenvalues and all eigenvectors.

7 Reconstructing Householder Vectors from Tall-Skinny QR

The Tall-Skinny QR (TSQR) algorithm is more communication efficient than the standard Householder algorithm for QR decomposition of matrices with many more rows than columns. However, TSQR produces a different representation of the orthogonal factor and therefore requires more software development to support the new representation. Further, implicitly applying the orthogonal factor to the trailing matrix in the context of factoring a square matrix is more complicated and costly than with the Householder representation. In a conference paper presented at IPDPS 2014 [6] and a journal version appearing in *JPDC* [7], we show how to perform TSQR and then reconstruct the Householder vector representation with the same asymptotic communication efficiency and little extra computational cost. We demonstrate the high performance and numerical stability of this algorithm both theoretically and empirically. The new Householder reconstruction algorithm allows us to design more efficient parallel QR algorithms, with significantly lower latency cost compared to Householder QR and lower bandwidth and latency costs compared with Communication-Avoiding QR (CAQR) algorithm. Experiments on supercomputers demonstrate the benefits of the communication cost improvements: in particular, our experiments show substantial improvements over tuned library implementations for tall-and-skinny matrices. We also provide algorithmic improvements to the Householder QR and CAQR algorithms, and we investigate several alternatives to the Householder reconstruction algorithm that sacrifice guarantees on numerical stability in some cases in order to obtain higher performance.

8 A Communication-Avoiding Parallel Algorithm for the Symmetric Eigenvalue Problem

Many large-scale scientific computations require eigenvalue solvers in a scaling regime where efficiency is limited by data movement. In a technical report [16], we introduce a parallel algorithm for computing the eigenvalues of a dense symmetric matrix, which performs asymptotically less communication than previously known approaches. We provide analysis in the Bulk Synchronous Parallel (BSP) model with additional consideration for communication between a local memory and cache. Given sufficient memory to store c copies of the symmetric matrix, our algorithm requires $\Theta(\sqrt{c})$ less interprocessor communication than previously known algorithms, for any $c \leq p^{1/3}$ when using p processors. The algorithm first reduces the dense symmetric matrix to a banded matrix with the same eigenvalues. Subsequently, the algorithm employs successive reduction to $O(\log p)$ thinner banded matrices. We employ two new parallel algorithms that achieve lower communication costs for the full-to-band and band-to-band reductions. Both of these algorithms leverage a novel QR factorization algorithm for rectangular matrices.

9 Hypergraph Partitioning for Parallel Sparse Matrix-Matrix Multiplication

In a conference paper presented at SPAA 2015 [9] and a journal version submitted to *ACM TOPC* [10], we propose a fine-grained hypergraph model for sparse matrix-matrix multiplication (SpGEMM), a key computational kernel in scientific computing and data analysis whose performance is often communication bound. This model correctly describes both the interprocessor communication volume along a critical path in a parallel computation and also the volume of data moving through the memory hierarchy in a sequential computation. We show that identifying a communication-optimal algorithm for particular input matrices is equivalent to solving a hypergraph partitioning problem. Our approach is sparsity dependent, meaning that we seek the best algorithm for the given input matrices. In addition to our (3D) fine-grained model, we also propose coarse-grained 1D and 2D models that correspond to simpler SpGEMM algorithms. We explore the relations between our models theoretically, and we study their performance experimentally in the context of three applications that use SpGEMM as a key computation. For each application, we find that at least one coarse-grained model is as communication efficient as the fine-grained model. We also observe that different applications have affinities for different algorithms. Our results demonstrate that hypergraphs are an accurate model for reasoning about the communication costs of SpGEMM as well as a practical tool for exploring the SpGEMM algorithm design space.

10 Reducing Communication Costs for Sparse Matrix Multiplication within Algebraic Multigrid

In a paper appearing in *SISC* [12], we consider the sequence of sparse matrix-matrix multiplications performed during the setup phase of algebraic multigrid. In particular, we show that the most commonly used parallel algorithm is often not the most communication-efficient one for all of the matrix-matrix multiplications involved. By using an alternative algorithm, we show that the communication costs are reduced (in theory and practice), and we demonstrate the performance benefit for both model (structured) and more realistic unstructured problems on large-scale distributed-memory parallel systems. Our theoretical analysis shows that we can reduce communication by a factor of up to 5.4 for a model problem, and we observe in our empirical evaluation communication reductions of factors up to 4.7 for structured problems and 3.7 for unstructured problems. These reductions in communication translate to run-time speedups of up to factors of 2.3 and 2.5, respectively.

11 Exploiting Multiple Levels of Parallelism in Sparse Matrix-Matrix Multiplication

Sparse matrix-matrix multiplication (or SpGEMM) is a key primitive for many high-performance graph algorithms as well as for some linear solvers, such as algebraic multigrid. The scaling of existing parallel implementations of SpGEMM is heavily bound by communication. Even though 3D (or 2.5D) algorithms have been proposed and theoretically analyzed in the flat MPI model on Erdos-Renyi matrices, those algorithms had not been implemented in practice and their complexities had not been analyzed for the general case. In a paper submitted to *SISC* [2], we present the first ever implementation of the 3D SpGEMM formulation that also exploits multiple (intra-node and inter-node) levels of parallelism, achieving significant speedups over the state-of-the-art publicly available codes at all levels of concurrencies. We extensively evaluate our implementation and identify bottlenecks that should be subject to further research.

12 A High-Performance Parallel Algorithm for Nonnegative Matrix Factorization

Non-negative matrix factorization (NMF) is the problem of determining two non-negative low rank factors W and H, for the given input matrix A, such that $A \approx WH$. NMF is a useful tool for many applications in different domains such as topic modeling in text mining, background separation in video analysis, and community detection in social networks. Despite its popularity in the data mining community, there is a lack of efficient distributed algorithm to solve the problem for big datasets. Existing distributed-memory algorithms are limited in terms of performance and applicability, as they are implemented using Hadoop and are designed only for sparse matrices. In a paper presented at PPoPP 2016 [14], we propose a distributed-memory parallel algorithm that computes the factorization by iteratively solving alternating non-negative least squares (NLS) subproblems for W and H. To our knowledge, our algorithm is the first high-performance parallel algorithm for NMF. It maintains the data and factor matrices in memory (distributed across processors), uses MPI for interprocessor communication, and, in the dense case, provably minimizes communication costs (under mild assumptions). As opposed to previous implementations, our algorithm is also flexible: (1) it performs well for both dense and sparse matrices, and (2) it allows the user to choose from among multiple algorithms for solving local NLS subproblems within the alternating iterations. We demonstrate the scalability of our algorithm and compare it with baseline implementations, showing significant performance improvements.

13 Parallel Tensor Compression for Large-Scale Scientific Data

As parallel computing trends towards the exascale, scientific data produced by high-fidelity simulations are growing increasingly massive. For instance, a simulation on a three-dimensional spatial grid with 512 points per dimension that tracks 64 variables per grid point for 128 time steps yields 8 TB of data, assuming double precision. By viewing the data as a dense five-way tensor, we can compute a Tucker decomposition to find inherent low-dimensional multilinear structure, achieving compression ratios of up to 5000 on real-world data sets with negligible loss in accuracy. So that we can operate on such massive data, in a paper presented at IPDPS 2016 [1], we present the first-ever distributed-memory parallel implementation for the Tucker decomposition, whose key computations correspond to parallel linear algebra operations, albeit with nonstandard data layouts. Our approach specifies a data distribution for tensors that avoids any tensor data redistribution, either locally or in parallel. We provide accompanying analysis of the computation and communication costs of the algorithms. To demonstrate the compression and accuracy of the method, we apply our approach to real-world data sets from combustion science simulations. We also provide detailed performance results, including parallel performance in both weak and strong scaling experiments.

14 Diamond Sampling for Approximate Maximum All-pairs Dot-product (MAD) Search

Given two sets of vectors A and B, our problem is to find the top-t dot products among all possible pairs where one vector is taken from each set. This is a fundamental mathematical problem that appears in numerous data applications involving similarity search, link prediction, and collaborative filtering. In a paper presented at ICDM 2015 [11], we propose a sampling-based approach that avoids direct computation of all dot products. We select diamonds (i.e., four-cycles) from the weighted tripartite representation of A and B. The probability of selecting a diamond corresponding to pair (i,j) is proportional to the square of the dot product between vectors i (from A) and j (from B), amplifying the focus on the largest-magnitude entries. Experimental results indicate that diamond sampling is orders of magnitude faster than direct computation and requires far fewer samples than any competing approach. We also apply diamond sampling to the special case of maximum inner product search, and get significantly better results than the state-of-the-art hashing methods. This paper was awarded the Best Paper prize at ICDM.

References

[1] W. Austin, G. Ballard, and T. G. Kolda. Parallel tensor compression for large-scale scientific data. In *30th IEEE International Parallel and Distributed Processing Symposium*, pages 912–922, May 2016.

[2] A. Azad, G. Ballard, A. Buluc, J. Demmel, L. Grigori, O. Schwartz, S. Toledo, and S. Williams. Exploiting multiple levels of parallelism in sparse matrix-matrix multiplication. Technical Report 1510.00844, arXiv, October 2015.

[3] G. Ballard, D. Becker, J. Demmel, J. Dongarra, A. Druinsky, I. Peled, O. Schwartz, S. Toledo, and I. Yamazaki. Communication-avoiding symmetric-indefinite factorization. *SIAM Journal on Matrix Analysis and Applications*, 35(4):1364–1406, November 2014.

[4] G. Ballard, A. R. Benson, A. Druinsky, B. Lipshitz, and O. Schwartz. Improving the numerical stability of fast matrix multiplication algorithms. Technical Report 1507.00687, arXiv, July 2015.

[5] G. Ballard, E. Carson, J. Demmel, M. Hoemmen, N. Knight, and O. Schwartz. Communication lower bounds and optimal algorithms for numerical linear algebra. *Acta Numerica*, 23:1–155, May 2014.

[6] G. Ballard, J. Demmel, L. Grigori, M. Jacquelin, H. D. Nguyen, and E. Solomonik. Reconstructing Householder vectors from tall-skinny QR. In *IEEE 28th International Parallel and Distributed Processing Symposium*, pages 1159–1170, May 2014.

[7] G. Ballard, J. Demmel, L. Grigori, N. Knight, M. Jacquelin, and H. D. Nguyen. Reconstructing Householder vectors from tall-skinny QR. *Journal of Parallel and Distributed Computing*, 85:3–31, August 2015.

[8] G. Ballard, J. Demmel, and N. Knight. Avoiding communication in successive band reduction. *ACM Transactions on Parallel Computing*, 1(2):11:1–11:37, February 2015.

[9] G. Ballard, A. Druinsky, N. Knight, and O. Schwartz. Brief announcement: Hypergraph partitioning for parallel sparse matrix-matrix multiplication. In *Proceedings of the 27th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '15, pages 86–88, New York, NY, USA, June 2015. ACM.

[10] G. Ballard, A. Druinsky, N. Knight, and O. Schwartz. Hypergraph partitioning for sparse matrix-matrix multiplication. Technical Report 1603.05627, arXiv, March 2016.

[11] G. Ballard, T. G. Kolda, A. Pinar, and C. Seshadhri. Diamond sampling for approximate maximum all-pairs dot-product (MAD) search. In *15th IEEE International Conference on Data Mining*, ICDM '15, pages 11–20. IEEE Computer Society, November 2015.

[12] G. Ballard, C. Siefert, and J. Hu. Reducing communication costs for sparse matrix multiplication within algebraic multigrid. *SIAM Journal on Scientific Computing*, 38(3):C203–C231, June 2016.

[13] A. R. Benson and G. Ballard. A framework for practical parallel fast matrix multiplication. In *Proceedings of the 20th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP 2015, pages 42–53, New York, NY, USA, February 2015. ACM.

[14] R. Kannan, G. Ballard, and H. Park. A high-performance parallel algorithm for nonnegative matrix factorization. In *Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '16, pages 9:1–9:11, New York, NY, USA, February 2016. ACM.

[15] A. Smirnov. The bilinear complexity and practical algorithms for matrix multiplication. *Computational Mathematics and Mathematical Physics*, 53(12):1781–1795, 2013.

[16] E. Solomonik, G. Ballard, J. Demmel, and T. Hoefler. A communication-avoiding parallel algorithm for the symmetric eigenvalue problem. Technical Report 1604.03703, arXiv, Apr. 2016.

DISTRIBUTION:

1	MS 9159	Grey Ballard, 8966
1	MS 9159	Tamara Kolda, 8966
1	MS 9159	Karim Mahrous, 8966
1	MS 0899	Technical Library, 8944 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911


Sandia National Laboratories