Brief Announcement: Lower Bounds on Communication for Sparse Cholesky Factorization of a Model Problem

Laura Grigori
INRIA Saclay - Ile de France
Université Paris-Sud 11
France
Laura.Grigori@inria.fr

Pierre-Yves David
INRIA Saclay - Ile de France
Pierre-yves.david@ens-lyon.org

James W. Demmel
Departments of Mathematics and CS, University of California Berkeley, USA
demmel@cs.berkeley.edu

Sylvain Peyronnet
LRI, Université Paris-Sud 11
France
syp@lri.fr

ABSTRACT

Previous work has shown that a lower bound on the number of words moved between large, slow memory and small, fast memory of size M by any conventional (non-Strassen like) direct linear algebra algorithm (matrix multiply, the LU, Cholesky, QR factorizations, ...) is $\Omega(\#flops/\sqrt(M))$. This holds for dense or sparse matrices. There are analogous lower bounds for the number of messages, and for parallel algorithms instead of sequential algorithms.

Our goal here is to find algorithms that attain these lower bounds on interesting classes of sparse matrices. We focus on matrices for which there is a lower bound on the number of flops of their Cholesky factorization. Our Cholesky lower bounds on communication hold for any possible ordering of the rows and columns of the matrix, and so are globally optimal in this sense. For matrices arising from discretization on two dimensional and three dimensional regular grids, we discuss sequential and parallel algorithms that are optimal in terms of communication. The algorithms turn out to require combining previously known sparse and dense Cholesky algorithms in simple ways.

Categories and Subject Descriptors: F.2.1 [Theory of Computation]: Analysis of Algorithms and Problem Complexity - Numerical Algorithms and Problems, Computations on matrices

General Terms: Algorithms

Keywords: communication bounds, sparse Cholesky

1. INTRODUCTION

Recent research has raised an increasing interest on identifying lower bounds on communication for operations in linear algebra and algorithms that attain them. This research started with results from [10, 11] that show that a lower bound on the volume of communication (bandwidth) for computing the product of two dense matrices is $\Omega(W/M^{1/2})$ and a lower bound on the number of messages transferred (latency) is $\Omega(W/M^{3/2})$, where W is the number of flops and M is the fast memory size in the case of a sequential algorithm and the local memory size in the case of a parallel algorithm. Here communication refers to the data transferred between large, slow memory and fast, small memory for a sequential algorithm, and to the data transferred between processors for a parallel algorithm. It has been shown that the same bound applies to LU factorization [4], Cholesky factorization [2], and other operations in linear algebra and their sparse implementations in [3]. Some of the algorithms in the literature attain the bounds, as the block algorithm and Cannon algorithm for sequential and parallel matrix multiplication. For QR and LU factorizations, new optimal algorithms have been designed [4, 8] that show significant speedups in practice.

In this paper we derive bounds on communication for sparse Cholesky factorization $A = LL^T$. We focus our analysis on matrices whose graphs satisfy a property, from which a lower bound on the number of flops of the Cholesky factorization can be derived [12]. This includes matrices arising from the discretization of PDEs (in particular using a finite difference operator) on regular grids of dimension k^s. The graphs of these matrices have good separators, and in this case the Cholesky factors L are sparse and the Cholesky factorization can be performed efficiently. In contrast, the Cholesky factors of matrices whose graphs don't have good separators are almost dense, and so the Cholesky factorization costs almost as much as the dense case. Lipton et al. show that almost all graphs don't have good separators [12].

For two dimensional (2D) and three dimensional (3D) regular grids, we describe a sequential algorithm and we identify that the parallel algorithm implemented in PSPASES [9] (when using an optimal layout) attain the lower bounds on communication. Both algorithms use nested dissection to order the input matrix [6].

2. SPARSE CHOLESKY FACTORIZATION

In this section we derive lower bounds on communication for matrices whose graphs satisfy a property that we describe in the following. Let A be a symmetric matrix of size $n \times n$. Its undirected graph, denoted $G = (V, E)$, has a vertex $i \in V$ for each row and column of A, and an edge $(i, j) \in E$ for each nonzero symmetric off-diagonal element $A_{ij} = A_{ji}$. We consider in this paper a matrix whose graph has the following property for some l: every set of vertices

SPAA'10, June 13–15, 2010, Thira, Santorini, Greece.
ACM 978-1-4503-0079-7/10/06.

$W \subset V$ such that $n/3 \leq |W| \leq 2n/3$, is adjacent to at least l vertices in $V - W$. Here $|W|$ denotes the cardinality of W. Then Lemma 2 and Theorem 10 in [12] show that for any ordering of A, its Cholesky factor contains a dense lower triangular matrix of size $l \times l$. This property can be used to compute a lower bound on the number of flops performed in the Cholesky factorization, independent of any reordering of the input matrix. These results can be applied to matrices resulting from a finite difference operator on regular grids [12], that is matrices whose graphs are defined on a $k \times k \times \ldots \times k$ (s times) mesh of k^s points, where each point is connected to its nearest neighbours (points on the boundary have fewer neighbours). In this particular case, similar results are also derived in [5].

Theorem 1. *Consider the Cholesky factorization LL^T of an $n \times n$ symmetric matrix A whose undirected graph $G = (V, E)$ has the following property for some l: every set of vertices $W \subset V$ with $n/3 \leq |W| \leq 2n/3$ is adjacent to at least l vertices in $V - W$. A lower bound on communication for computing the Cholesky factorization of A is*

$$\#words \geq \Omega\left(\frac{W}{\sqrt{M}}\right), \qquad \#messages \geq \Omega\left(\frac{W}{M^{3/2}}\right)$$

For a sequential algorithm, $W = l^3$ and M is the fast memory size. For a parallel algorithm executed on P processors that is work-balanced, $W = \frac{l^3}{P}$. We assume that the matrix and the L factor are distributed evenly over all the processors and the local memory size used is estimated to be $M = \Theta(nnz(L)/P)$.

Proof. Lemma 2 in [12] says that the graph of the Cholesky factor L contains a clique of at least l vertices. This means that L contains a dense lower triangular matrix L_s of size $l \times l$. Theorem 10 in [12] uses this result to derive a lower bound on the number of floating point operations of the Cholesky factorization of $l(l-1)(l-4)/6$.

The lower bounds on communication developed in [2, 3] provide a lower bound on communication for the computation of L_s, and hence a lower bound for the Cholesky factorization of the entire matrix A. This leads to the communication bounds in the theorem. □

For a regular grid of dimension k^s, with $n = k^s$, $l = \Theta(n^{(s-1)/s})$. For 2D and 3D regular grids, the lower bounds derived from Theorem 1 are presented in Table 1 for the parallel case and Table 2 for the sequential case. We note that [3] presents also the lower bound for the 2D case. However our results apply to a larger class of graphs and in particular to regular grids of higher dimension. Nested dissection is an optimal ordering for the grid [6], and consists of partitioning the associated undirected graph of the sparse symmetric matrix using a divide-and-conquer paradigm. The nested dissection method is based on finding a small vertex separator, S, that partitions the graph into two disconnected subgraphs. The rows and columns associated with the vertices of the disconnected subgraphs are ordered first, followed by those corresponding to the vertices of the separator S. The permuted matrix PAP^T has the form

$$\begin{pmatrix} A_{11} & 0 & A_{13} \\ 0 & A_{22} & A_{23} \\ A_{13}^T & A_{23}^T & A_{33} \end{pmatrix}.$$

The partitioning can then be applied recursively on the subgraphs corresponding to the submatrices A_{11} and A_{22}. The partitioning defines a binary tree structure, called the separator tree. Each node of this tree corresponds to a separator. The root of the tree corresponds to the separator from the first level partitioning.

In the following we briefly analyze sequential and parallel algorithms that attain the communication bounds. Detailed performance counts for these algorithms will be presented in an extended version of this paper. Our analysis considers that nested dissection uses + separators. That is, in the case of 2D grids, a separator partitions a square grid into four square subgrids. In the case of 3D grids, a separator formed by three orthogonal planes partitions the grid into eight subgrids. We discuss multifrontal methods, that compute the Cholesky factorization by using the separator tree. We give a brief description here. Each node in the separator has associated a frontal matrix. This matrix is formed by the vertices of the separator and the vertices that correspond to columns modified by the vertices of the separator. The Cholesky factorization is performed during a bottom-up traversal of the separator tree. At each node of the tree, a number of steps equal to the size of the separator of Cholesky factorization is performed on the associated frontal matrix. Then, the update matrix is transmitted to the parent node. At the parent node, the update matrices are merged through extend-add operations to form a frontal matrix. And then the factorization continues on this new frontal matrix.

The parallel algorithm implemented in PSPASES is based on a multifrontal method and uses the separator tree to distribute the input matrix over the processors using a cyclic approach and a subtree to subcube mapping [7]. This algorithm maps nodes to processors during a top-down traversal of the separator tree. It starts by assigning all the P processors to the root. Then it assigns (recursively) $P/4$ processors to each of the four subtrees of the root. The frontal matrix is distributed among those processors using a 2D cyclic distribution. The communication complexity of the algorithm is analyzed in [9], and we display it in Table 1. With an appropriate layout as described in [9], the merge of the update matrices has a low communication cost. Hence the communication in the Cholesky factorization of the frontal matrices associated with nodes in the separator tree dominates the overall communication. PSPASES attains the lower bound on bandwidth, but not on latency, as displayed in Table 1.

	PSPASES	PSPASES with optimal layout	Lower bound Thm. 1
2D grids			
# flops	$O\left(\frac{n^{3/2}}{P}\right)$	$O\left(\frac{n^{3/2}}{P}\right)$	$O\left(\frac{n^{3/2}}{P}\right)$
# words	$O(\frac{n}{\sqrt{P}})$	$O\left(\frac{n}{\sqrt{P}} \log P\right)$	$\Omega\left(\frac{n}{\sqrt{P \log n}}\right)$
# messages	$O(\sqrt{n})$	$O\left(\sqrt{P} \log^3 P\right)$	$\Omega\left(\frac{\sqrt{P}}{(\log n)^{3/2}}\right)$
3D grids			
# flops	$O\left(\frac{n^2}{P}\right)$	$O\left(\frac{n^2}{P}\right)$	$O\left(\frac{n^2}{P}\right)$
# words	$O(\frac{n^{4/3}}{\sqrt{P}})$	$O\left(\frac{n^{4/3}}{\sqrt{P}} \log P\right)$	$\Omega\left(\frac{n^{4/3}}{\sqrt{P}}\right)$
# messages	$O(n^{2/3})$	$O\left(\sqrt{P} \log^3 P\right)$	$\Omega\left(\sqrt{P}\right)$

Table 1: Performance of PSPASES, PSPASES with optimal layout and lower bounds on communication when factoring an $n \times n$ matrix resulting from 2D and 3D regular grids. Some lower order terms are omitted. The analysis assumes the local memory of each processor is $M = O(n \log n/P)$ in the 2D case and $M = O(n^{4/3}/P)$ in the 3D case.

This is due to the fact that the analysis of PSPASES uses a cyclic distribution, and this involves the exchange of a message for each step of Cholesky factorization. A block cyclic distribution will decrease the latency but will still not allow to attain the lower bound. To attain the lower bound on latency an optimal layout needs to be used. We use the same approach as in [4] in which the matrix is distributed in a two dimensional block cyclic layout using square blocks of size $b \times b$ and letting b be close to its maximal value. In other words, we consider that the factorization of each frontal matrix is performed using a ScaLAPACK-like algorithm with an optimal layout. This leads to performance results presented in Table 1, which show that with an optimal layout, PSPASES attains the latency and bandwidth lower bounds, modulo polylog factors. The number of floating point operations is optimal, modulo constant factors.

We discuss now an optimal algorithm for sequentially computing the Cholesky factorization of matrices arising from 2D and 3D regular grids. The analysis is performed in a big O sense. The algorithm considers that the input matrix has been ordered using nested dissection based on + separators and uses a multifrontal Cholesky factorization. The algorithm computes the factorization during a postorder traversal of the separator tree. At each node of the tree, the factorization consists of two main steps. The update matrices of its child nodes are read from slow memory and merged through an extend-add operation to form the frontal matrix of this node. The postorder traversal ensures that the update matrices can be stored on a stack. Then a number of steps of Cholesky factorization are performed on this frontal matrix. The update matrix is then stored on slow memory, such that the parent node can use it. For the partial Cholesky factorization of each frontal matrix, the algorithm uses a recursive Cholesky factorization algorithm. This algorithm presented in [1] has been shown to be optimal through multiple levels of memory hierarchy with an appropriate recursive block storage [2], where each block fits in the fast memory of size M. The communication necessary to copy a dense matrix of size $n \times n$ stored in column major or row major order into a block format is asymptotically equal to the communication necessary to perform the Cholesky factorization of this matrix given $M = O(n)$ [2]. We assume this or a smaller bound in our analysis, depending on the size of the frontal matrix. The reads and writes between different levels of memory occur at two different phases of the algorithm, when the partial Cholesky factorization of a frontal matrix is computed, and when the update matrices are merged to form a frontal matrix. The upper bounds on communication of this optimal Cholesky algorithm, presented in Table 2, attain the lower bounds of Theorem 1.

Problem		Optimal Cholesky	Lower bound
2D grids	# flops	$O\left(n^{3/2}\right)$	$O\left(n^{3/2}\right)$
	# words	$O\left(\frac{n^{3/2}}{\sqrt{M}} + n\log n\right)$	$\Omega\left(\frac{n^{3/2}}{\sqrt{M}}\right)$
	# messages	$O\left(\frac{n^{3/2}}{M^{3/2}} + \frac{n\log n}{M}\right)$	$\Omega\left(\frac{n^{3/2}}{M^{3/2}}\right)$
3D grids	# flops	$O\left(n^{2}\right)$	$O\left(n^{2}\right)$
	# words	$O\left(\frac{n^{2}}{\sqrt{M}} + n^{4/3}\right)$	$\Omega\left(\frac{n^{2}}{\sqrt{M}}\right)$
	# messages	$O\left(\frac{n^{2}}{M^{3/2}} + \frac{n^{4/3}}{M}\right)$	$\Omega\left(\frac{n^{2}}{M^{3/2}}\right)$

Table 2: Performance of optimal sequential multifrontal Cholesky factorization when factoring an $n \times n$ matrix resulting from 2D and 3D regular grids. The lower bounds on communication are also presented, and M is the fast memory size. The analysis assumes $M = O(\sqrt{n})$ in the 2D case and $M = O(n^{2/3})$ in the 3D case.

3. CONCLUSIONS

In this paper we have discussed bounds on communication for sparse Cholesky factorization of a certain class of matrices, that includes matrices resulting from regular grids. The approach used here to derive optimal algorithms can be used for other classes of graphs with good separators as well. Consider the case when the computation and communication to factor the submatrix formed by the vertices of the first separator dominates the overall communication and computation. Then an optimal algorithm can be derived by using an optimal dense algorithm to factor the submatrix formed by the vertices of the separator.

We do not discuss other approaches as right-looking or left-looking. It is possible that a right looking factorization with an appropriate optimal layout that takes into account the sparsity of the input matrix might be optimal. This remains as future work.

4. ACKNOWLEDGMENTS

The work of L. Grigori is supported in part by French National Research Agency (ANR) through COSINUS program (project PETAL no ANR-08-COSI-009). The research of J. W. Demmel is supported by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG07-10227).

5. REFERENCES

[1] N. Ahmed and K. Pingali. Automatic generation of block-recursive codes. In Springer-Verlag, editor, *Euro-Par*, 2000, pages 368–378, 2000.

[2] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Communication-optimal parallel and sequential Cholesky decomposition. *ACM SPAA*, 2009.

[3] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Minimizing communication in linear algebra. Technical Report UCB/EECS-2009-62, UC Berkeley, 2009.

[4] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou. Communication-optimal parallel and sequential QR and LU factorizations. Technical Report UCB/EECS-2008-89, UC Berkeley, 2008. LAPACK Working Note 204.

[5] S. C. Eisenstat, M. H. Schultz, and A. H. Sherman. Applications of an element model for Gaussian elimination. In J. Bunch and D. Rose, editors, *Sparse Matrix Computations*, pages 85–96. Academic Press, New York, 1976.

[6] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10:345–363, 1973.

[7] A. George, J. W.-H. Liu, and E. G. Ng. Communication results for parallel sparse Cholesky factorization on a hypercube. *Parallel Computing*, 10(3):287–298, 1989.

[8] L. Grigori, J. W. Demmel, and H. Xiang. Communication avoiding Gaussian elimination. *Proceedings of the ACM/IEEE SC08 Conference*, 2008.

[9] A. Gupta, G. Karypis, and V. Kumar. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed Systems*, 8(5), 1995.

[10] J.-W. Hong and H. T. Kung. I/O complexity: The Red-Blue Pebble Game. In *STOC '81: Proceedings of the Thirteenth Annual ACM Symposium on Theory of Computing*, pages 326–333, New York, NY, USA, 1981. ACM.

[11] D. Irony, S. Toledo, and A. Tiskin. Communication lower bounds for distributed-memory matrix multiplication. *Journal of Parallel and Distribed Computing*, 64(9):1017–1026, 2004.

[12] R. J. Lipton, D. J. Rose, and R. E. Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16:346–358, 1979.