Communication-Optimal Algorithms

Grey Ballard, Mark Hoemmen

{ballard,mhoemmen}@cs.berkeley.edu

Motivation

"Communication" means

- Parallel: Data movement between processors
- Sequential: Data movement between levels of memory hierarchy
- # words (inverse bandwidth) and # messages (latency)

Communication matters because:

- Much slower than flops, and getting *exponentially slower* over time
- Moving data much more *energy-intensive* than computing on it
- Sparse linear algebra kernels already communication-bound
- Dense linear algebra: strong scaling demands increase relative comm. cost

Direct Methods

Summary

- New communication lower bounds for (nearly) all dense or sparse, sequential or parallel, direct linear algebra problems
- New algorithms that attain lower bounds (dense only, sequential and parallel)
- Measured and modeled speedups, not just asymptotics

Dense Matrix Multiplication

Lower bound on:	Lower bound
# words	$\Omega\left(\text{\# flops} / (\text{local/fast memory size})^{1/2}\right)$
# messages	$\Omega\left(\text{\# flops} / (\text{local/fast memory size})^{3/2}\right)$

- Results due to Hong-Kung [HK81], Irony/Tishkin/Toledo [ITT04]
- Attained by usual block algorithm (sequential) and Cannon's algorithm (parallel)

Extensions to (nearly) all direct problems

- Theorem: same lower bounds hold for LU, Cholesky, QR, eigenproblems, and SVD
  - Sequential or parallel, dense or sparse
  - See [BDHS09b] for details and proof
- Existing (Sca)LAPACK routines not both bandwidth and latency optimal
  - ScaLAPACK: only Cholesky is optimal; LAPACK: Cholesky bandwidth only
  - See [BDHS09a] for details on Cholesky algorithms
- New algorithms to attain lower bounds (up to polylog factors)
  - CAQR (QR factorization): new panel factorization & representation of Q
  - CALU (LU factorization): new pivoting scheme (still stable)
  - Eigenproblems and SVD: constant factor more flops, and randomization (see [DDH07])

New algorithm - Communication-Avoiding LU (CALU)

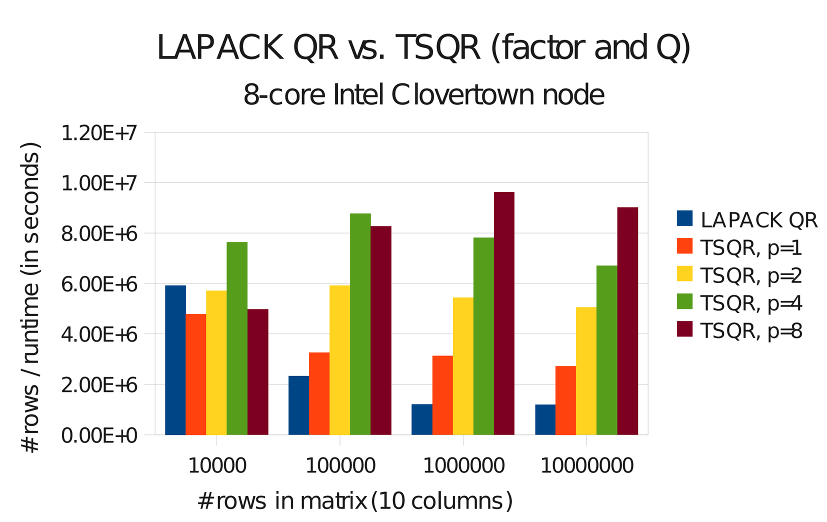
- Factor panel once with "Tall Skinny LU" (like a block reduction) to choose pivots
- Swap pivot rows to top and factor *again* without pivoting – $O(n^2)$ extra computation
- Measured speedup of parallel TSLU: up to $5.58\times$ on Cray XT4
- Measured speedup of parallel CALU (size $10^4 \times 10^4$): $1.31\times$ on Cray XT4
- See [DGX08] for details, models, and more performance results

New algorithm - Communication-Avoiding QR (CAQR)

- Panel factorization: "Tall Skinny QR" (TSQR) – block reduction with QR as operator
- Measured speedup of parallel TSQR: up to $6.7\times$ on 16 procs of a Pentium III cluster
- Modeled speedup of parallel CAQR: up to $9.7\times$ on an IBM Power5 system
- See [DGHL08] for details, models, and more performance results
- Standalone TSQR useful for iterative methods (orthogonalize basis vectors)

TSQR performance results

- Single node of 8-core Intel Clovertown (we have cluster and out-of-core versions too)
- Includes factorization and assembling explicit Q factor
- Best number of threads for LAPACK QR (Intel MKL and stock LAPACK): 1
- Even better measured and modeled speedups on clusters



Iterative Methods

◇ Subject of Mark Hoemmen's 11:30 talk

Based on 2 (or 3) communication-bound kernels

1. Sparse matrix-vector multiplication (SpMV)
2. (Possibly also preconditioning)
3. Orthogonalization (explicit, like Gram-Schmidt, or implicit, like in CG)

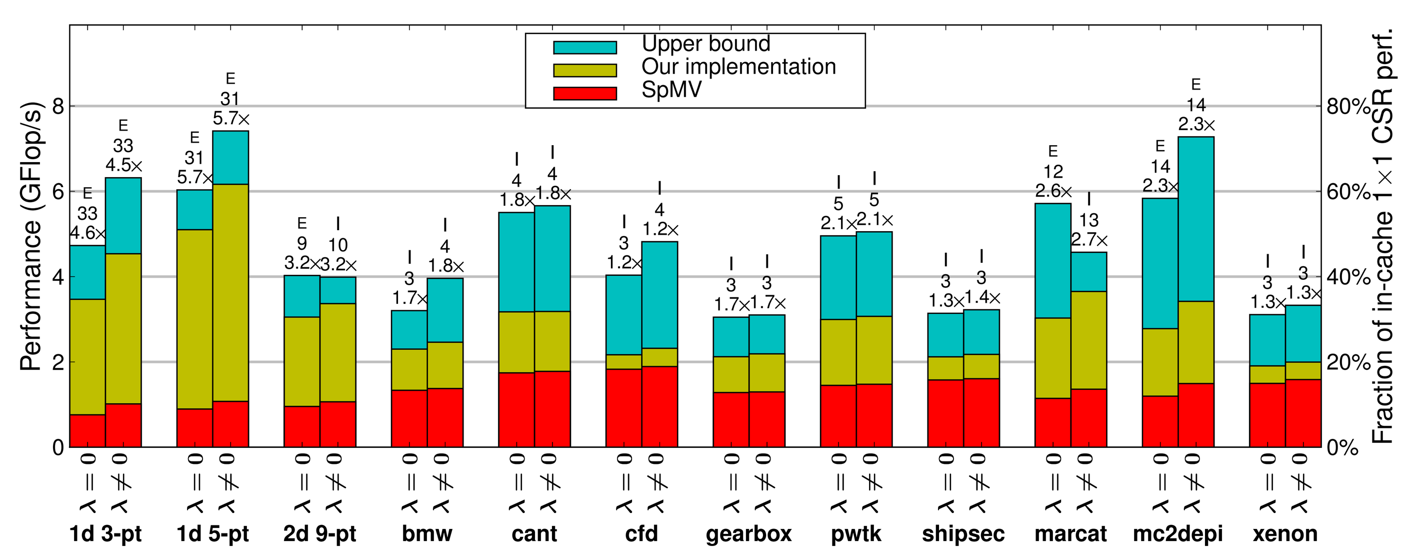
Our new algorithms

- Communicate factor of s times less than existing iterative methods (this is optimal)
- Work as long as sparse matrix structure partitions well (true for structured and unstructured meshes, as well as other matrices)
- Mathematically equivalent to existing methods and stable in practice

Based on two new kernels

- Matrix powers kernel: use (possibly) redundant computation to compute a basis of $\text{span}\{v, Av, A^2v, \dots, A^sv\}$
- TSQR: orthogonalize this basis accurately in 1 reduction
- See [MHDY09] for implementation and performance details of a GMRES algorithm using these two kernels

Matrix powers kernel performance results



- Matrix powers kernel on a variety of sparse matrices, symmetric and nonsymmetric
- Red (on bottom) is *tuned* $A \cdot x$, green (next) is matrix powers kernel, blue (top) is upper bound
- Performance relative to *in L2 cache* $A \cdot x$ ("instruction throughput measured peak")
- Two different bases computed:
  1. $\lambda = 0$ is "power basis" $v, Av, A^2v, \dots$
  2. $\lambda \neq 0$ is "Newton basis" $v, (A - \lambda_1 I)v, (A - \lambda_2 I)(A - \lambda_1 I)v, \dots$

Credits

- Research supported by Microsoft (Award #024263) and Intel (Award #024894) funding and by matching funding by U.C. Discovery (Award #DIG07-10227).
- Joint work with J. Demmel, L. Grigori, O. Holtz, J. Langou, M. Mohiyuddin, O. Schwartz, H. Xiang

References

[BDHS09a] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. Communication-optimal parallel and sequential Cholesky-decomposition. Technical Report EECS-2009-29, University of California Berkeley EECS, 2009. Accepted by ACM SPAA.

[BDHS09b] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz. A general communication lower bound for linear algebra, 2009. Submitted to FOCS.

[DDH07] J. Demmel, I. Dumitriu, and O. Holtz. Fast linear algebra is stable. *Numer. Math.*, 108(1):59–91, 2007.

[DGHL08] J. Demmel, L. Grigori, M. Hoemmen, and J. Langou. Communication-optimal parallel and sequential QR and LU factorizations. Technical Report EECS-2008-89, University of California Berkeley EECS, August 2008. Submitted to SIAM J. Sci. Comp.

[DGX08] J. Demmel, L. Grigori, and H. Xiang. Communication-avoiding Gaussian elimination. In *Supercomputing 2008*, 2008.

[HK81] Jia-Wei 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.

[ITT04] D. Irony, S. Toledo, and A. Tiskin. Communication lower bounds for distributed-memory matrix multiplication. *J. Parallel Distrib. Comput.*, 64(9):1017–1026, 2004.

[MHDY09] M. Mohiyuddin, M. Hoemmen, J. Demmel, and K. Yelick. Minimizing communication in sparse matrix solvers, 2009. Submitted to Supercomputing '09.