Hierarchical Algorithms with Reduced Communication/Synchronization

Xiaoye S. Li (xsli@lbl.gov)
Computational Research Division
Lawrence Berkeley National Laboratory, Berkeley, CA 94720.

It is anticipated that large partial differential equations (PDEs) will underpin many computer simulation and modeling codes to be executed on the exascale machines. For certain PDEs, most notably the elliptic ones, there have been considerable successes in developing fast and scalable solution methods, such as multigrid and fast multipole (FMM) based methods. The progress has been less satisfactory for those non-selfadjoint and indefinite PDEs. These kinds of systems occur more often now when we model complex physical systems that involve multiple time and length scales or multiple physical components. These systems are often too difficult for current state-of-the-art methods, particularly so at scale because of high resolution and high fidelity demands. We propose to develop two new classes of algorithms that are targeted for broad classes of PDEs and make effective use of the exascale systems.

1. Randomized sparse factorization with low-rank compression

It was long discovered that for structured matrices, such as Hilbert matrix, Toeplitz matrix and the matrices from the BEM methods for integral equations, the off-diagonal blocks are numerically rank deficient. Several compression techniques using hierarchically truncated SVD representation have been develoeped. These compact forms can be used to design asymptotically faster and low-memory linear algebra algorithms [1, 5, 6, 2, 10]. Only recently, these structured matrix ideas were introduced in the sparse solvers for solving discretized PDEs [14, 3]. In particular, we have pioneered the work of using *hierarchically semi-separable* (HSS) structure to develop superfast sparse direct solver. We are the first group to have developed parallel algorithms for the key HSS operations (e.g., construction, ULV factorization & solution) [12]. Moreover, we integrated these parallel HSS kernels into the parallel sparse multifrontal method, and demonstrated that our HSS-embedded sparse solver can solve the discretized Helmholtz equations with 6.4 billion unknowns using 4096 processing cores. This is beyond reach of both direct solvers and multigrid solvers [11].

Despite the initial success, we encountered one major road block: the compression kernel based on traditional rank-revealing QR (RRQR) is difficult to scale up. Worse yet, it leads to non-compatible HSS structures, which prevents fast Schur complement updates (*extend-add*) at each step of Gaussian elimination. We discovered that the *randomized sampling* (RS) method can replace RRQR to obtain the desired truncated SVD compression. The following procedure computes an orthonormal basis (ON-basis) U for an $m \times n$ matrix B such that $\|B - UU^*B\|$ is small *with high probability*.

1. Choose a Gaussian random matrix $\Omega_{n\times(r+p)}$, where r is the expected rank and p is a small over-sampling constant of $O(1)$.
2. Form the sample matrix $S = B\,\Omega$ whose columns span the column space of B.
3. Construct an ON-basis U for S using a strong RRQR.

It can be shown that U is a good ON-basis for B's column space. The key saving comes from the fact that the column dimension of the sampling matrix S is significantly smaller. This procedure can be used in each step of the HSS-embedded multifrontal factorization as follows: 1) using the RS procedure to construt HSS forms for the frontal and update matrices [7]; 2) form a sample update matrix by multiplying the update matrix with a random matrix; 3) perform extend-add of the *sampled* update matrix to the parent. The advantages are:

- Extend-add involves tall-skinny dense sample matrices instead of HSS structures.
- The kernel operation is dense matrix-matrix multiplication, which is well studied for good scalability and resilience (ABFT).
- It is especially attractive when the matrix from the application admits FMM structure which leads to very fast matrix-vector product, or when the application cannot afford to store the matrix but only has matrix-vector operator; then we can develop the *matrix-free* sparse factorization algorithms.

This randomized sparse factorization framework can be used to construct nearly linear-time sparse direct solver and preconditioner [13], sparse eigensolver, selected inversion, among others. Although this type of solver/preconditioner is not as black-box as a direct solver, it is applicable to broad classes of PDEs, including some of those for which multigrid does not work [8]. The new research includes:

- For this stochastic process, derive reliable quantifications of error and confidence interval, and develop resilience mechanism in response to both soft and hardware failures.
- Develop adaptive algorithms to accommodate rank variations at different parts of the matrix.
- Perform theoretical analysis on memory access and communication complexity for certain model PDEs.

2. Communication-reducing methods for hybrid sparse solvers with hierarchical parallelism

For very difficult systems of PDEs arising from multiscale and multiphysics simulations, the current solver technologies are either less scalable or less robust at exascale. The MHD systems from plasma fusion simulation is such an example, where, the coupled equations are employed to model multiphysics phenomena, including equations for electro-magnetic field, momentum, Ohm's law, electron and ion energies. Recently we have been developing a direct/iterative hybrid solver PDSLin [15] targeting at these highly indefinite problems. PDSLin is based on a non-overlapping domain decomposition technique called the Schur complement method. We have successfully used PDSLin to model the magnetic reconnection process in the magnetized plasma using thousands of cores [16]. Despite its robustness, much research is needed to further it in scalability. In PDSLin, we use two-level parallelism to keep the number of subdomains small by assigning multiple processors to each subdomain. Hence, the numbers of subdomains can be far fewer than the number of cores, maintaining an excellent convergence rate. The scalability issue can be addressed in several components.

- For each subdomain on a subset of processors, we employ a parallel direct solver (e.g. SuperLU_DIST) to deal with a relatively smaller matrix. The critical path of the algorithm is panel factorization involving small tall-skinny, but distributed dense matrices. The recent communication-avoiding TSLU algorithm [4] could be used here to avoid communication dominance. Another idea is to develop the sparse version of the 2.5D dense LU factorization [9].
- We currently use (I)LU factor of approximate Schur complement as a preconditioner to GMRES for solving the Schur complement system. This requires parallel sparse matrix-matrix multiplication across the entire machine. It would be very beneficial to develop a sparse version of the 2.5D dense matrix multiplication [9].
- We can employ the HSS low-rank compression of the Schur complement as an alternative preconditioner, which incurs much less communication and memory requirement than (I)LU.

Combining the above communication-reducing techniques at both medium and large scale paralell levels would improve the overall scalabality substantially.

References

[1] M. Bebendorf. *Hierarchical Matrices*, volume 63 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin Heidelberg, 2008.

[2] S. Chandrasekaran, P. Dewilde, M. Gu, T. Pals, X. Sun, A.-J. van der Veen, and D. White. Some fast algorithms for sequentially semiseparable representations. *SIAM J. Matrix Anal. Appl.*, 27:341–364, 2005.

[3] B. Engquist and L. Ying. Sweeping preconditioner for the helmholtz equation: hierarchical matrix representation. *Commun. Pure Appl. Math.*, 64:697–735, 2011.

[4] L. Grigori, J. Demmel, and H. Xiang. Communication-avoiding Gaussian Elimination. In *Supercomputing 08*, Austin, Texas, November 15-21 2008.

[5] W. Hackbusch, L. Grasedyck, and S. Börm. An introduction to hierarchical matrices. *Math. Bohem.*, 127:229–241, 2002.

[6] W. Hackbusch, B. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. *Lectures on applied mathematics (Munich, 1999), Springer, Berlin*, pages 9–29, 2000.

[7] N. Halko, P.G. Martinsson, and J.A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011.

[8] A. Napov, X.S. Li, and M. Gu. An algebraic multifrontal preconditioner that exploits a low-rank property. Technical report, Lawrence Berkeley National Laboratory, 2013. In preparation.

[9] Edgar Solomonik and James Demmel. Communication-optimal parallel 2.5d matrix multiplication and lu factorization algorithms. In *Lecture Notes in Computer Science, Euro-Par*, Bordeaux, France, August 2011.

[10] R. Vandebril, M. Van Barel, G. Golub, and N. Mastronardi. A bibliography on semiseparable matrices. *Calcolo*, 42:249–270, 2005.

[11] S. Wang, X.S. Li, F.-H. Rouet, J. Xia, and M.V. de Hoop. A parallel fast geometric multifrontal solver using hierarchically semiseparable structure. *ACM Trans. Mathematical Software*, 2013. In preparation.

[12] S. Wang, X.S. Li, J. Xia, Y. Situ, and M.V. de Hoop. Efficient parallel algorithms for solving linear systems with hierarchically semiseparable structures. *SIAM J. Scientific Computing*, 2013. (revised).

[13] J. Xia. Randomized sparse direct solvers. *SIAM J. Matrix Analysis and Applications*, 34:197–227, 2013.

[14] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Superfast multifrontal method for large structured linear systems of equations. *SIAM J. Matrix Anal. Appl.*, 31:1382–1411, 2009.

[15] I. Yamazaki and X.S. Li. PDSLin: Parallel domain decomposition schur complement based linear solver. http://crd-legacy.lbl.gov/FASTMath-LBNL/Software/.

[16] X. Yuan, X.S. Li, I. Yamazaki, S.C. Jardin, A.E. Koniges, and D.E. Keyes. Application of PDSLin to the magnetic reconnection problem. *IOP Special Issue: Computational Science & Discovery*, 6(1), 2013.