Small dots, big challenging? ☆

Shengxin Zhu

Oxford Centre for Collaborative and Applied Mathematics, Mathematical Institute, 24-29 St Giles', Oxford, OX1 3LB, U.K.

Abstract

It is tremendously difficult for sparse iterative solvers to fully utilize the computing capability of a supercomputer because of communications, especially those global ones. The global/collective communications are usually necessary when computing the small `dots`—inner products—on distributed supercomputers, and it is easy to dominate the entire solution time if it is unavoidable. Communications can be partially hidden by some computations, but it is difficult for one to hide all parts of a huge object by a tinny one. When each processor is assigned to the same amount of computations, we observe that on a Dawning 5000A, the global communication time increases in the order of $P^{4/5}$ against the number of processors, P, while the local communication for some structured sparse matrix vector multiplications increases in the order $P^{1/2.5}$. The order of $P^{4/5}$ depends on the complexity of the MPI reduce algorithm for collective communications, while the order $P^{1/2.5}$ depends on the computer architecture, especially how the processors connect with each others. This brings immediately two challenging problems, (a) how to design an optimal algorithms for collective communications, and (b) how to get an optimal assembly of CPUs/GPUs with multiple constraints. There may be fertile opportunities to (c) develop inner product free iterative methods (not necessary Krylov type) without losing accuracy. When taking preconditioning and hierarchical techniques into account, an unconventional thinking may lead to better direction: using Krylov methods as local sub-solvers and classical iterative methods as main solvers, (d) which may redefine the role of some classical iterative methods in a modern time. In words, the small dots bring great challenges as well as potential collaborative opportunities for numerical analysts, computer scientists, software engineers and vendors.

Keywords: sparse iterative solvers, high performance computing, synchronizations, inner product

Problems related to the small dots computations

1. Accurate and fast MPI reduce algorithms for collective communications.
2. Optimal assembly of CPUs/GPUs with multiple constraints.
3. Inner product free iterative solvers for high performance distributed computing.
4. Accurate and fast inner product computations on distributed computers.
5. Synchronization avoiding/hiding algorithms.
6. New framework of preconditioning with classical iterative methods.

References

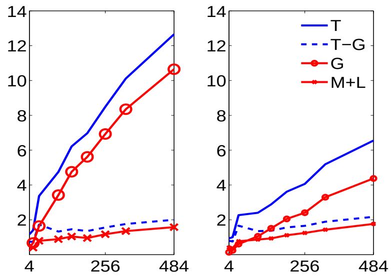
[1] T. Ma, G. Bosilca, A. Bouteiller and J. D. Dongarra, Kernel-assisted and topology-aware MPI collective communications on multicore/many-cor platforms *J. Parallel and Distributed Computing*, 2013 http://dx.doi.org/10.1016/j.jpdc.2013.01.015

[2] A.Castaldo and R. Whaley and A. Chronopoulos. Reducing floating point error in DOT product using the superblock Family of Algorithms. *SIAM. J. Sci. Comput.* 31(2),1156-1174, 2008.

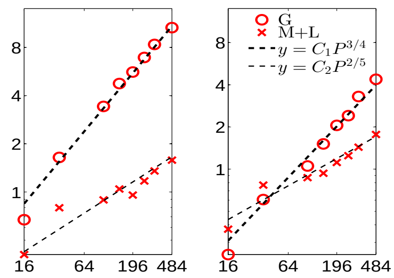
Email address: `shengxin.zhu@maths.ox.ac.uk` (Shengxin Zhu)

URL: `https://www.maths.ox.ac.uk/contact/details/zhus` (Shengxin Zhu)

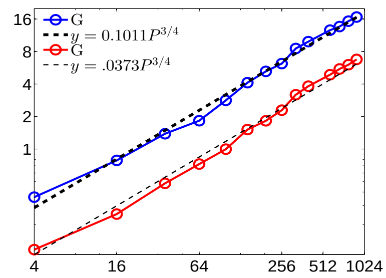
[1] The authors' research is supported by Award no KUK-C1-013-04, made by King Abdullah University of Science of Technology.



(a) Timing for a Krylov method method(left) and corresponding parallel version(right). T: total time, G: Global communication time, T-G: time excluding global communication, M+L: matrix vector multiplication computation and (local) communication time.



(b) The role of matrix vector multiplications and global communication time in a Krylov method(left) and corresponding parallel version. G: global communication times. M+L: matrix vector multiplication computation and (local) communication time.



(c) Global communication time for a Krylov method (blue) and corresponding parallel version(red). The data shows the underlying MPI reduce operations have a complexity of $P^{4/5}$. As shown in (b), $G \gg M + L$, when P is large. Results from [27]

[3] A.T. Chronopoulos and C.W. Gear, s-step iterative methods for symmetric linear systems. *J. Comput. Appl. Math*, 25(2):153-168,1989.
[4] A.T. Chronopoulos and C.W. Gear, On the efficient implementation of preconditioned s-step iterative methods on multiprocessors with memory hierachy. *Parallel Computing*, 11(1):37-53,1989.
[5] A. T. Chronopoulos and A.B. Kucherov, Block s-step Krylov iterative methods, *Numer. Linear Algebra Appl.*, 17(3):3-15,2010.
[6] A.T. Chronopoulos and C.D. Swanson, Parallel iterative s-step methods for unsymmetric linear systems. *Parallel Computing*, 22(5):623-641,1996.
[7] T.P. Collignon and M.B. van Gijzen, Minimizing synchronization in IDR(s). *Numer. Linear Algebra Appl.*, 18:805-825, 2011.
[8] E. de Sturler and H.A. van der Vorst. Reducing the effect of global communication in GMRES(m) and CG on parallel distributed memory computers. *Applied Numerical Mathematics*, 18(4): 441–459, 1995.
[9] J. Demmel and Y. Hida. Accurate and Efficient Floating Point Summation *SIAM J. Sci. Comput.* (25)4, 1212-1248, 2003.
[10] Tong-Xiang Gu, Xing-Ping Liu, Ze-Yao Mo, and Xue-Bin Chi. Multiple search direction conjugate gradient method. I. Methods and their propositions. *Int. J. Comput. Math.*, 81(9): 1133–1143, 2004.
[11] Tong-Xiang Gu, Xing-Ping Liu, Ze-Yao Mo, and Xue-Bin Chi. Multiple search direction conjugate gradient method. II. Theory and numerical experiments. *Int. J. Comput. Math.*, 81(10): 1289–1307, 2004.
[12] T. Gu, X. Zuo, X. Liu, and P. Li. An improved parallel hybrid bi-conjugate gradient method suitable for distributed parallel computing. *J. Comput. Appl. Math.*, 226(1): 55–65, 2009.
[13] T. Gu, X. Zuo, L. Zhang, W. Zhang, and Z. Sheng. An improved bi-conjugate residual algorithm suitable for distributed parallel computing. *Appl. Math. Comput.*, 186(2): 1243–1253, 2007.
[14] M.H. Gutknecht and Z. Strakos, Accuracy of two three-term and three two-term recurrences for Krylov space solvers. *SIAM. J. Matrx Anal. Appl.*, 22(1):213-229, 2000
[15] L. C. McInnes, B. Smith, H. Zhang and R. T. Mills, Hierarchical and Nested Krylov Methods for Extreme-Scale Computing, 2012. *Preprint ANL/MCS-P2097-0612* www.mcs.anl.gov/uploads/cels/papers/P2097-0612.pdf
[16] R. Nishtala and K.A. Yelick. Optimizing collective communication on multicores. In *Proceedings of the First USENIX conference on Hot Topics in Parallelism*, pages 18–18. USENIX Association, 2009.
[17] T. Ogita, and S. Rump, and S. Oishi Accurate Sum and Dot Product. *SIAM J. Sci. Comput.* 26(6), 1955-1988, 2005.
[18] S. Rump, and S. Ogita, and S. Oishi. Accurate Floating-Point Summation Part I: Faithful Rounding *SIAM J. Sci. Comput.* 31(1),189–224, 2008.
[19] G.L.G. Sleijpen and H.A. van der Vorst. Reliable updated residuals in hybrid Bi-CG methods *Computing*, 56(2):141-163, 1996.
[20] A. Wathen, T. Rees, Chebshev semi-iteration in preconditioning for problems including the mass matrix. Electron. Trans. Numer. Anal., 34:125-135, 2008/09.
[21] H. Li, T. Huang, Y. Zhang, X. Liu and T. Gu, Chebyshev-type methods and preconditioning techniques, *Appl. Math. Comput.*, 218(2):260-270, 2011.
[22] L.T. Yang. The improved CGS method for large and sparse linear systems on bulk synchronous parallel architectures. In *Algorithms and Architectures for Parallel Processing, 2002. Proceedings. Fifth International Conference on*, pages 232–237. IEEE, 2002.
[23] T.R. Yang and H.X. Lin. The improved quasi-minimal residual method on massively distributed memory computers. In *High-Performance Computing and Networking*, pages 389–399. Springer, 1997.
[24] S. Zhu Parellel GPBiCG(m, ℓ) methods and preconditioning techneques. *Master Thesis, Graduate School, China Academy of Engineering Physics*, 2010
[25] S. Zhu Small dots, big challenging. *Oxford Conference on Challenges in Applied Mathematics*, July 2013.
[26] S. Zhu, T. Gu and X. Liu Inner product computations for sparse iterative solvers on distributed supercomputers. , *OCCAM report*, 2012
[27] S. Zhu, T. Gu and X. Liu Minimizing synchronizations in sparse iterative solvers for distributed supercomputers. *submitted*, 2013.