# Graph Algorithms for Highly Parallel Sparse Linear Solvers

Proposal for Postdoctoral Research Project
Mentor: Anshul Gupta, IBM T.J. Watson Research Center, Yorktown Heights, NY, 10595

April 1, 2009

## Introduction

The core computation in a large number of applications in science, engineering, and optimization involves solving large sparse systems of linear equations. With the rapidly increasing number and complexity of scientific applications that model time dependent physical phenomena, there is an urgent need for robust, high-performance, highly-parallel, practical general purpose sparse linear solvers because of the limitations of almost all iterative solver softwares [7] and the asymptotically superlinear time and memory demands of direct solvers [10]. We have had some recent success [1, 9] in developing general purpose iterative solvers for symmetric positive definite (SPD) systems with improved robustness and performance. This proposal briefly describes some of our plans to extend this success to unsymmetric and indefinite systems and to develop highly parallel solvers for SPD systems.

## Parallel Matching Algorithms

Matching is a fundamental combinatorial problem that has applications in many contexts. One particular formulation of the matching problem, computing the maximum weight edge matching in a bipartite graph (MWEM-BP), is of particular importance in the context of solving sparse linear systems of equations, where the coefficient matrix is unsymmetric or indefinite [6, 15, 19, 8, 11]. MWEM-BP can be used to generate permutations and scaling vectors to maximize the diagonal dominance of the coefficient matrix, which typically improves the convergence properties of iterative solvers and the performance of direct solvers by minimizing pivoting. Our preliminary research shows that applying MWEM-BP permutation and scaling can be crucial to the robustness of incomplete factorization based preconditioners for indefinite and unsymmetric sparse matrices.

Conventional exact MWEM-BP algorithms have the drawback of being hard to parallelize [16] due to their inherent sequential nature. This poses a serious challenge to their application for solving extremely large linear systems that are becoming increasingly common. Typically, these systems are solved in large distributed-memory parallel environments, and currently, cannot benefit from MWEM-BP preprocessing. The goal of this research is to develop highly parallel algorithms for the MWEM-BP problem. We plan to explore parallelization of exact algorithms, as well as new approximation algorithms that permit efficient and scalable parallel implementations.

Even though matching problems can be solved in polynomial time, approximation algorithms are in demand due to many practical needs [18]. Approximation algorithms that are less expensive than exact matching algorithms are important even in moderately parallel shared-memory situations in which the matching algorithm has access to the entire coefficient matrix. The reason is that many practical applications require solution to a large number of linear systems with varying values of entries in the coefficient matrix. As a result, unlike some other symbolic preprocessing steps, the matching algorithm needs to be reapplied frequently and it is desirable to minimize its total cost.

While linear-time $\frac{1}{2}$-approx algorithms have been proposed for the MWEM-BP problem, the same has eluded for approximation ratios better than half [17, 20]. Halappanavar et al. [12] have recently developed a new parallel half-approximation algorithm for MWEM-BP and have demonstrated its scalability on up to tens of thousands of processors. We have experimented with Halappanavar et al.'s algorithms and found them to be only partially effective in the context of sparse linear solvers. We suspect that the problem is that while the approximation algorithm generates matchings of reasonably good quality, unlike the exact algorithms, it does not not generate the scaling vectors. Therefore, we plan to focus some of our research efforts into either developing approximation algorithms that also address scaling, or find a way of generating approximate scaling vectors within Halappanavar et al.'s framework.

Another research direction that we plan to pursue is to study auction based algorithms for the MWEM-BP problem, such as the one proposed by Bertsekas and Castañon [2]. Auction algorithms are exact algorithms, but appear to be more amenable to parallelization [3] than the more traditional algorithms based on the Hangarian method (Gupta and Ying [11] have done a detailed survey of such algorithms). Other appealing properties of the auction-based algorithms are that it may be possible to truncate their iterations to derive faster approximation versions, and that they generate scaling vectors for the coefficient matrix.

## Parallel Graph Partitioning Algorithms

Effective parallel graph partitioning and reordering is vital to balancing computation and minimizing communication while solving PDE's on massively parallel computers. There are a handful of free and commercial software packages for parallel graph partitioning and repartitioning (to support adaptive mesh refinement) [14, 21, 4] and their viability on massively parallel machines with tens or hundreds of thousands of nodes of massively parallel computers such as IBM's BlueGene and Cray's XT-4/5 is unclear, at best.

In the first phase of our proposed research on parallel graph partitioning, we plan to study (empirically and analytically) the algorithms and existing s/w packages (e.g. ParMetis, Jostle, Scotch, and others) for parallel mesh partitioning and repartitioning, look at the advantages and shortcomings of each scheme, and determine which, if any, have the potential for scaling to tens of thousands of nodes or more. After this study, we plan to come up with recommendations for the next phase of the work. These recommendations could be of the form: (1) method/software A can be modified/enhanced, or (2) a combination of B+C will result in a satisfactory partitioner, or (3) all existing solutions have fundamental limitations and we need to build a new solution from scratch using the best ideas from A, B, and C and using new algorithms to overcome the limitations, etc.

During the next phase of the research, we plan to implement the recommendations from the first phase, with particular attention to the fact that a $K$-way partitioning among $K$ nodes must be performed in parallel by $K \times m$ cores, with $m$ cores per node. None of the existing graph partitioning packages are designed for hierarchical parallelism. In particular, we plan to develop and use highly parallel algorithms for vertex matching for coarsen graphs to implement multi-level algorithms [13]. This may require either developing new parallel exact or approximation algorithms, or parallel versions of recently developed serial approximation algorithms [5].

## References


[1] Haim Avron, Anshul Gupta, and Sivan Toledo. New Krylov-subspace solvers for Hermitian positive definite matrices with indefinite preconditioners. Technical Report RC 24698, IBM T. J. Watson Research Center, Yorktown Heights, NY, December 2008. Submitted to *SIAM Journal on Matrix Analysis and Applications.*

[2] Dimitri P. Bertsekas and David A. Castañon. An auction algorithm for the transportation problem. Technical Report LIDS-P-1850, Massachusetts Institute of Technology, 1989.

[3] Dimitri P. Bertsekas and David A. Castañon. Parallel synchronous and asynchronous implementations of the auction algorithm. *Parallel Computing*, 17(6-7):707–732, 1991.

[4] C. Chevalier and F. Pellegrini. Pt-scotch: A tool for efficient parallel graph ordering. *Parallel Computing*, 34(6-8):318–331, 2008.

[5] Florin Dobrian, Mahantesh Halappanavar, and Alex Pothen. Algorithms for vertex weighted matching in graphs. In *To be submitted to ACM Transactions on Algorithms.*

[6] I. S. Duff and J. Koster. On algorithms for permuting large entries to the diagonal of a sparse matrix. *SIAM J. Matrix Anal. Appl.*, 22(4):973–996, 2000.

[7] Thomas George, Anshul Gupta, and Vivek Sarin. An empirical analysis of iterative solver performance for SPD systems. Technical Report RC 24737, IBM T. J. Watson Research Center, Yorktown Heights, NY, January 2009. A short version submitted to *ACM Transactions on Mathematical Software*.

[8] Anshul Gupta. Recent advances in direct methods for solving unsymmetric sparse systems of linear equations. *ACM Transactions on Mathematical Software*, 28(3):301–324, September 2002.

[9] Anshul Gupta and Thomas George. Adaptive techniques for improving the performance of incomplete factorization preconditioning. Technical Report RC 24598, IBM T. J. Watson Research Center, Yorktown Heights, NY, July 2008. To appear in *SIAM Journal on Scientific Computing.*

[10] Anshul Gupta, Seid Koric, and Thomas George. Sparse matrix factorization on massively parallel computers. In *Submitted to Supercomputing 2009.*

[11] Anshul Gupta and Lexing Ying. On algorithms for finding maximum matchings in bipartite graphs. Technical Report RC 21576, IBM T. J. Watson Research Center, Yorktown Heights, NY, October 1999.

[12] Mahantesh Halappanavar, Florin Dobrian, and Alex Pothen. A scalable approximate matching algorithm for the distributed parallel computers. In *Submitted to Supercomputing 2009.*

[13] George Karypis and Vipin Kumar. Analysis of multilevel graph partitioning. In *Supercomputing '95: Proceedings of the 1995 ACM/IEEE conference on Supercomputing (CDROM)*, page 29, New York, NY, USA, 1995. ACM.

[14] George Karypis and Vipin Kumar. ParMETIS: Parallel graph partitioning and sparse matrix ordering library. Technical Report TR 97-060, Department of Computer Science, University of Minnesota, 1997.

[15] Xiaoye S. Li and James W. Demmel. Making sparse gaussian elimination scalable by static pivoting. In *Supercomputing '98: Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)*, pages 1–17, Washington, DC, USA, 1998. IEEE Computer Society.

[16] Andrew Lumsdaine, Douglas Gregor, Bruce Hendrickson, and Jonathan W. Berry. Challenges in parallel graph processing. *Parallel Processing Letters*, 17(1):5–20, 2007.

[17] Seth Pettie and Peter Sanders. A simpler linear time $\frac{2}{3} - \epsilon$ approximation for maximum weight matching. *Inf. Process. Lett.*, 91(6):271–276, 2004.

[18] Robert Preis. Linear time $\frac{1}{2}$-approximation algorithm for maximum weighted matching in general graphs. In *16th Ann. Symp. on Theoretical Aspects of Computer Science (STACS)*, pages 259–269, 1999.

[19] Olaf Schenk, Andreas Wächter, and Michael Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Comput. Optim. Appl.*, 36(2-3):321–341, 2007.

[20] Doratha E. Drake Vinkemeier and Stefan Hougardy. A linear-time approximation algorithm for weighted matchings in graphs. *ACM Trans. Algorithms*, 1(1):107–122, 2005.

[21] Chris Walshaw and M. Cross. JOSTLE: Parallel Multilevel Graph-Partitioning Software – An Overview. In F. Magoules, editor, *Mesh Partitioning Techniques and Domain Decomposition Techniques*, pages 27–58. Civil-Comp Ltd., 2007.