Masha Sosonkina

Tuning a graph partitioner to improve parallel preconditioners

Ames Laboratory
Iowa State University
236C Wilhelm Hall
Ames IA 50011
masha@scl.ameslab.gov
Yousef Saad

[11pt]article

Tuning a graph partitioner to improve parallel preconditioners

Masha Sosonkina [1] Yousef Saad [2]

February 2, 2004

[1]Ames Laboratory DOE, 236 Wilhelm Hall, Ames, IA 50011. E-mail: masha@scl.ameslab.gov. Phone: (515) 294-6751. http://www.scl.ameslab.gov/∼masha

[2]Department of Computer Science and Engineering, University of Minnesota, 4-192 EE/CS Building, 200 Union Street S.E., Minneapolis, MN 55455. E-mail: saad@cs.umn.edu. Phone: (612) 624-7804. http://www.cs.umn.edu/∼saad

Abstract

Various graph partitioning algorithms are employed to partition a sparse matrix for parallel processing, which exploit a number of graph representations of a matrix. A standard approach is to represent rows (or unknowns) as graph vertices and the nonzero entries as the edges of the graph. The partitioning objective here has been to minimize the edge cut while retaining a balance between the sizes of the partitions. However, it is known [1] that such graph partitionings are not necessarily optimal for certain simple types of parallel computations, such as sparse matrix vector products, and for problems with irregular structure. Alternative approaches have been suggested to overcome this shortcoming. One of them is the hypergraph model, which generalizes the notion of a graph by permitting edges defined by more than two vertices. It has been shown that, for certain definitions of vertices and hyperedges, the hypergraph model minimizes correctly communication overhead for sparse matrix operations and represents adequately non-symmetric matrices. Typically the hypergraph approach, as well as other graph partitionings, disregards the numerical values of matrix entries. Numerical properties, however, are an important factor to consider when partitioning a difficult linear system for an iterative solution. When matrix values are taken into account, the resulting subproblems may be better conditioned if, for example, all strongly coupled coefficients reside in the same subproblem [2]. In this talk, we show a way to define hyperedges and their weights based on a preprocessing analysis of linear system. Specifically, first, we compute a relative degree of diagonal dominance for each matrix row. Then we construct a hyperedge to include a certain number of adjacent vertices depending on this relative degree. The larger this degree the smaller the number of vertices that constitute a hyperedge. We also show how criteria of load balancing and of minimization of communication overhead affect our hyperedge definitions.

Numerical experiments with linear systems arising in real world applications will be reported. We compare several graph partitioning techniques, including hypergraph models with typical hyperedge definitions. Along with measuring communication overhead and load balance, we monitor the quality of the resulting distributed preconditioner, keeping in mind that the ultimate goal is to produce faster convergence in terms of number of iterations and execution time.

Bibliography

[1] Bruce Hendrickson and Tamara G. Kolda. Graph partitioning models for parallel computing. *Parallel Computing*, 26(12):1519–1534, 2000.

[2] Y. Saad and M. Sosonkina. Non-standard parallel solution strategies for distributed sparse linear systems. In A. Uhl P. Zinterhof, M. Vajtersic, editor, *Parallel Computation: Proc. of ACPC'99*, Lecture Notes in Computer Science, Berlin, 1999. Springer-Verlag.