Constrained Fine-Grain Parallel Sparse Matrix Distribution

Erik G Boman* Umit Catalyurek†

We consider how to distribute sparse matrices among processors to reduce communication cost in parallel sparse matrix computations, in particular, sparse matrix-vector multiplication. We allow 2d distributions, where the distribution (partitioning) is not constrained to just rows or columns. The fine-grain model is a 2d distribution introduced in [2] where nonzeros can be assigned to processors in an arbitrary general way. They proposed a hypergraph model and showed it can significantly reduce the communication volume compared to 1d distributions.

We define a constrained version of this problem, where the input and output vector distributions are given. We propose two combinatorial models. The first is based on vertex cover in the bipartite graph, and the second on hypergraph partitioning with fixed vertices. Though NP-hard, both models can be solved heuristically using existing algorithms and software.

Sparse matrix-vector multiplication is usually parallelized such that the processor that owns element a_{ij} computes the contribution $a_{ij}x_j$. This is a local operation if x_j, y_i and a_{ij} all reside on the same processor; otherwise communication is required. In general, the following four steps are performed:

Expand Send entries x_j to processors that need them.

Local multiply $y_i += a_{ij}x_j$

Fold Send partial y values to relevant processors.

Sum up Sum up the partial y values.

In parallel computing, matrices are typically partitioned in a 1d fashion, either by rows or by columns. This partitioning problem has been modeled both as graph partitioning in the symmetric case, and as bipartite graph partitioning [3] in the nonsymmetric and rectangular case. Hypergraph partitioning [1] has been shown to accurately model communication volume (for both symmetric and nonsymmetric), and is therefore often preferred today. For 1d distributions, only one communication phase is required, either the expand or the fold but not both. Recently, several 2d decompositions have been proposed. The idea is to reduce the communication volume further by giving up the simplicity of the 1d structure. We focus on the fine-grain distribution [2], because it is the most general.

In some applications, the input vector x and the output vector y may already have a parallel distribution that cannot easily be changed. One such example is an iterative linear solver with a preconditioner. In each iteration, we need to multiply by a matrix A and solve for a preconditioner. We have the flexibility to choose the parallel distribution of A but the preconditioner is often a "black box" that requires a certain distribution for the input and output vectors. In this case, our task is to find the optimal sparse matrix distribution given the vector distributions:

Given a sparse matrix $A \in R^{n\times m}$ and vectors $x \in R^n$ and $y \in R^m$. Suppose x and y have been partitioned into k partitions, for some integer $k > 1$. The *vector-constrained sparse matrix partitioning problem* is then: Assign each nonzero in A to one of the k partitions such that the communication volume in parallel computation of $y = Ax$ is minimized and each partition (processor) has approximately the same number of nonzeros.

Initially, we consider only bisection ($k = 2$), that is, two partitions or processors. This case is illustrated in Figure 1. For simplicity, we assume x and y are permuted so that the first half belongs to one processor and the second half to the other. It is clear that all nonzeros in the upper left corner should be assigned to the red processor to avoid communication. Similarly, the bottom right should go to the blue processor. The question is how to assign

*egboman@sandia.gov, Discrete Algorithms and Math Department, Sandia National Labs, Albuquerque, NM 87185-1318. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the DOE's National Nuclear Security Administration under contract number DE-AC-94AL85000.

†umit@bmi.osu.edu, Dept. of Biomedical Informatics, Ohio State University

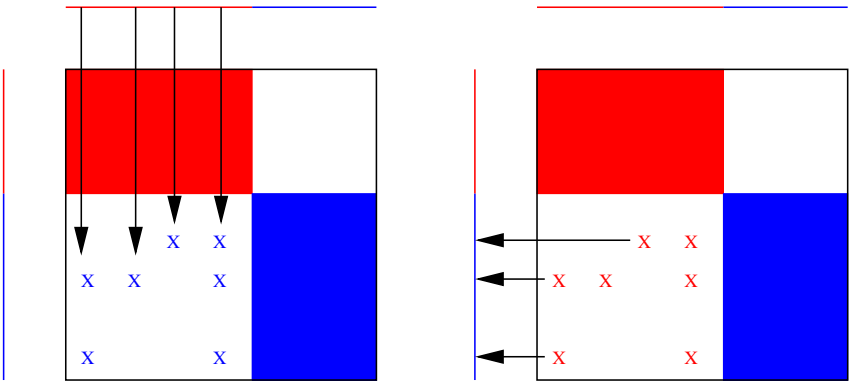

Figure 1: Matrix with constrained input (above the matrix) and output (left of matrix) vectors. There are two processors (red and blue), and the arrows represent communication in parallel matrix-vector multiply.

the nonzeros in the off-diagonal blocks. In our example, there are only off-diagonal nonzeros in the bottom left block, but in general there could be some in the top right block as well. In that case, these two subblocks can be solved independently.

Consider first 1d row or column distribution. For row distribution, we assign all nonzeros in the top half to the red processor and the bottom to the blue processor. There is no communication in the fold phase, but the expand phase requires communication. Each column that contains at least one off-block-diagonal entry requires one element of x to be sent. In other words, the communication volume equals the number of columns with at least one entry outside the block diagonal [3]. For column distribution, the same result holds, but counting rows instead of columns. In Figure 1 there are seven relevant nonzeros, and there are three rows and four columns. Row distribution has communication volume of four (left) while column distribution has volume of three (right).

A simple strategy is to compute the communication volumes for both row and column distributions and pick the better one. However, one can possibly lower the communication volume by splitting the nonzeros between the two processors. This is a core idea of this paper. We propose two models that lead to two different algorithms.

Our first model is based on the bipartite graph model. We only need consider the off-diagonal blocks. Construct the bipartite graph of one matrix block. Compute a minimal vertex cover. The cover may include both row and column vertices. We show this cover corresponds to communication volume; therefore minimizing the cover minimizes communication.

Our second model is a modification of the fine-grain hypergraph model. We include the given input and output vectors as an additional row and column, respectively, in the matrix. Further, each nonzero in these vectors correspond to a *fixed vertex* in the corresponding hypergraph, which cannot be reassigned. This gives a hypergraph partitioning problem with fixed vertices, which can be efficiently solved (approximately) using existing software like PaToH or Zoltan.

In the full paper we show that our models generalize to the k-way partitioning problem. We will present preliminary empirical experiments comparing these two approaches.

References

[1] Ü. Çatalyürek and C. Aykanat. Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication. *IEEE Trans. Parallel Dist. Systems*, 10(7):673–693, 1999.

[2] Ü. Çatalyürek and C. Aykanat. A fine-grain hypergraph model for 2d decomposition of sparse matrices. In *Proc. IPDPS 8th Int'l Workshop on Solving Irregularly Structured Problems in Parallel (Irregular 2001)*, April 2001.

[3] B. Hendrickson and T. G. Kolda. Partitioning rectangular and structurally nonsymmetric sparse matrices for parallel computation. *SIAM Journal on Scientific Computing*, 21(6):2048–2072, 2000.