

PARTITIONING AND SPARSE COMPUTATIONS AT EXASCALE

ERIK G. BOMAN, KAREN D. DEVINE, ERIC T. PHIPPS, AND SIVASANKARAN RAJAMANICKAM
SANDIA NATIONAL LABORATORIES

1. BACKGROUND

Exascale computing is a great challenge, and will require new hardware, software, and algorithms. In this position paper we highlight some issues related to data partitioning and solvers that we believe have been insufficiently addressed by the exascale programs so far. Our selected topics fall into the area of *combinatorial scientific computing*. We make the assumption that future exascale systems will have both distributed and shared memory. It is sometimes assumed that data partitioning is a “solved problem”, but we argue this is not the case. In fact, since future systems likely will have deeper memory hierarchies, the importance of data partitioning and placement will grow. This is an issue both across nodes and within a single node.

2. COMPLEX SYSTEMS AND NETWORK ANALYSIS

The amount of data available for analysis is growing rapidly. Much of this data does not come from physics-based simulation, but from other domains such as social networks or complex systems. Such data can often be represented as graphs (or hypergraphs) that represent relationships (edges) between data objects (vertices). These graphs are highly irregular, and tend to be *scale-free* with a power-law degree distribution. Traditional HPC has had moderate success handling such graphs. One problem is that they are highly unstructured and difficult to partition across processors/nodes for parallel computing. Therefore, alternative computing models such as data-flow and Map-Reduce are sometimes preferred. We believe that traditional HPC still has a role to play in data analysis of complex networks and scale-free graphs. For example, eigenanalysis is commonly used for clustering and community detection. A key challenge has been to partition the data to enable scalable matrix-vector multiply for iterative solvers. Current graph (hypergraph) partitioners were designed for mesh-based problems, where the graphs have bounded degrees, and do not perform well on power-law graphs. It was shown in [4] that the traditional 1-dimensional data distribution is ineffective and not scalable for scale-free graphs, while 2-dimensional distributions reduced run times for an eigensolver by up to two orders of magnitude for large problems. Recently, new 2-dimensional data distributions have shown even better speedups and good scalability up to 16K cores [1]. We believe such data distributions will be essential to exascale for certain types of problems. Therefore, further research into partitioning for complex networks, scale-free graphs, and other highly irregular data is needed. Furthermore, the existing solver stack must adapt to handle such data distributions. While frameworks such as Trilinos and PETSc can handle 2D distributions for the matrix-vector multiplication, most preconditioners do not. Both algorithmic research and software development is needed. Two-dimensional data layouts can also be useful in other application areas, for example, nuclear structure [2].

3. PARTITIONING FOR SOLVERS

Generic tools such as graph partitioners have been highly successful at helping applications distribute data for parallel computing. Unfortunately, the current use has several flaws. Currently, partitioners partition data to balance the load and to minimize communication. However, the data partitioning also effects the solvers, primarily by changing the preconditioners. The most popular preconditioners at large scale are multigrid methods and domain decomposition. In both cases, the effects of the partitioning on the convergence rate and the solver performance is poorly understood. We believe that for exascale problem

Sandia is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

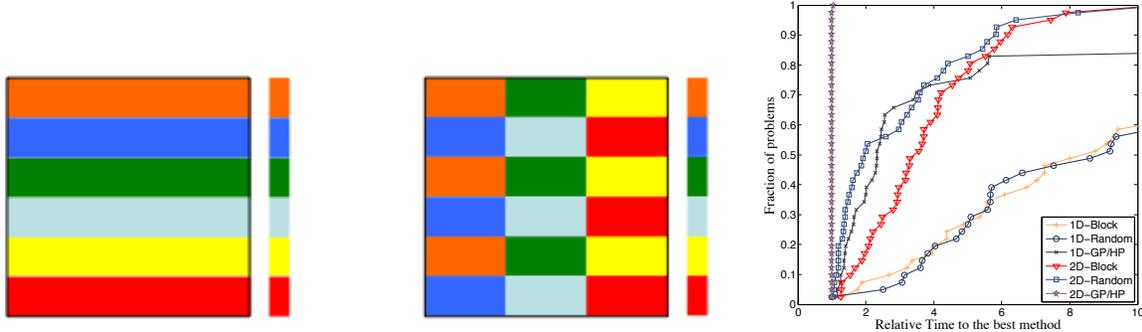


FIGURE 1. Block 1D (left) and block 2D (center) data distributions. To the right, a performance profile of matrix-vector multiply times for several 1D and 2D layouts for a set of scale-free graphs run on 1024 to 16K cores.

sizes, this issue can no longer be ignored. New partitioning approaches must be developed that take the physics of the application, the numerics of the linear systems, and the properties of the preconditioner into account. In domain decomposition, the choice of subdomains clearly impacts convergence, though this is difficult to model mathematically. In multigrid solvers, the aggregation is typically done locally within a node, so it is affected by the data distribution. Interestingly, both multigrid (AMG) solvers and partitioners use a hierarchy of coarser grids. Currently these are completely separate; it would be interesting to explore whether they could be combined to both save time in the preconditioner setup phase (which is often costly) and improve convergence.

Research is needed to (a) understand the mathematical issues involved, (b) develop new algorithms, and (c) implement into software. Modest progress has recently been made in [3], but their analysis and techniques are limited to additive Schwarz with no overlap on symmetric-positive-definite linear systems.

4. SPARSE TENSOR COMPUTATIONS

Much work has been done on partitioning sparse matrices for parallel computing. The next extension is sparse tensors, which can be viewed as sparse multi-dimensional arrays. Sparse tensor computations are becoming increasingly important as data becomes more multi-dimensional. The stochastic Galerkin method for UQ is one example highly relevant to exascale. A key kernel is to compute “contractions” over certain modes in the tensors. For example, when C is a sparse 3-tensor, then $\sum_j C_{ijk}$ is a sparse matrix. In stochastic Galerkin one needs to solve systems of the type

$$\sum_{jk} C_{ijk} a_j x_k = b_i, \quad \forall i,$$

for x . In an iterative method, the tensor C_{ijk} will be accessed repeatedly. An efficient data structure for sparse tensors is key, but the data partitioning is also important in a parallel setting. While much work has been done on partitioning sparse matrices, almost no work has been done on tensors. While some techniques may generalize to tensors, most do not, so new research into models and algorithms is needed.

REFERENCES

1. Erik G. Boman, Karen D. Devine, and Sivasankaran Rajamanickam, *Scalable matrix computations for large scale-free graphs using 2d graph partitioning*, Tech. report, Sandia National Labs, 2013, submitted for publication.
2. Philip Sternberg, Esmond G. Ng, Chao Yang, Pieter Maris, James P. Vary, Masha Sosonkina, and Hung Viet Le, *Accelerating configuration interaction calculations for nuclear structure*, Proceedings of the 2008 ACM/IEEE conference on Supercomputing (Piscataway, NJ, USA), SC '08, IEEE Press, 2008, pp. 15:1–15:12.
3. Eugene Vecharynski, Yousef Saad, and Masha Sosonkina, *Graph partitioning with matrix coefficients for symmetric positive definite linear systems*, Tech. Report UMSI-2011-143, U. of Minnesota, 2011, submitted for publication.
4. Andy Yoo, Allison H. Baker, Roger Pearce, and Van Emden Henson, *A scalable eigensolver for large scale-free graphs using 2d graph partitioning*, Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (New York, NY, USA), SC '11, ACM, 2011, pp. 63:1–63:11.