Abstract for CSC14: The Sixth SIAM Workshop on Combinatorial Scientific Computing

Nested dissection with balanced halo

Astrid Casadei[1,3], Pierre Ramet[1,3], and Jean Roman[1,2]

[1]INRIA Bordeaux Sud-Ouest & CNRS (LaBRI UMR 5800)
[2]Bordeaux Institute of Technology (IPB)
[3]Bordeaux University

Nested Dissection (ND) has been introduced by A. George in 1973 [2] and is a well-known and very popular heuristic for sparse matrix ordering to reduce both fill-in and operation count during Cholesky factorization. This method is based on graph partitioning and the basic idea is to build a "small separator C" associated with the original matrix in order to split the remaining vertices in two parts A and B of "almost equal sizes". The vertices of the separator C are ordered with the largest indices, and then, the same method is applied recursively on the two subgraphs induced by A and B. Good separators can be built for classes of graphs occurring in finite element problems based on meshes which are special cases of bounded density graphs or more generally of overlap graphs. In d-dimension, such n-node graphs have separators whose size grows as $\mathcal{O}(n^{(d-1)/d})$. In this presentation, we focus on the cases $d = 2$ and $d = 3$ which correspond to the most interesting practical cases for numerical scientific applications. ND has been implemented by graph partitioners such as MeTiS or Scotch[6].

Moreover, ND is based on a divide and conquer approach and is also very well suited to maximize the number of independent computation tasks for parallel implementations of direct solvers. Then, by using the block data structure induced by the partition of separators in the original graph, very efficient parallel block solvers have been designed and implemented according to supernodal or multifrontal approaches. To name a few, one can cite MUMPS, PaStiX and SuperLU. However, if we examine precisely the complexity analysis for the estimation of asymptotic bounds for fill-in or operation count when using ND ordering[5], we can notice that the size of the halo of the separated subgraphs (set of external vertices adjacent to the subgraphs and previously ordered) play a crucial role in the asymptotic behavior achieved. The minimization of the halo is in fact never considered in the context of standard graph partitioning and therefore in sparse direct factorization studies.

In this presentation, we will focus on hybrid solvers combining direct and iterative methods and based on domain decomposition and Schur complement approaches. The goal is to provide robustness similar to sparse direct solvers, but memory usage more similar to preconditioned iterative solvers. Several sparse solvers like HIPS, MaPHyS, PDSLIN and ShyLU implement different versions of this hybridification principle.

In this context, the computational cost associated to each subdomain for which a sparse direct elimination based on ND ordering is carried out, as well as the computational cost of the iterative part of the hybrid solver, critically depend on the halo size of the subdomains. However, to our knowledge, there does not exist a domain decomposition tool leading to a good balancing of both the internal node set size and the halo node size. Standard partitioning techniques, even by using k-way partitioning approach, which intends to construct directly a domain decomposition of a graph in k sets of independent vertices[4], do not lead in general to good results for the two coupled criteria, and for general irregular graphs coming from real-life scientific applications.

For this purpose, we revisit the original algorithm introduced by Lipton, Rose and Tarjan [5] in 1979 which performed the recursion for nested dissection in a different manner: at each level, we apply recursively the method to the subgraphs induced by $A \cup C$ on one hand, and $B \cup C$ on the other hand. In these subgraphs, vertices already ordered (and belonging to previous separators) are the halo vertices. The partition of these subgraphs will be performed with three objectives: balancing of the two new parts A' and B', balancing of the halo vertices in these parts A' and B' and minimizing the size of the separator C'.

We implement this strategy in the Scotch partitioner. Scotch strategy is based on the multilevel method[3] which consists in three main steps: the (sub)graph is coarsened multiple times until it becomes small enough, then an algorithm called greedy graph growing is applied on the coarsest graph to find a good separator, and finally the graph is uncoarsened, projecting at each level the coarse separator on a finer graph and refining it using the Fiduccia-Mattheyses algorithm[1].

We have adapted the multilevel framework of Scotch in order to take into account the halo vertices from orginal to coarsest graph. Moreover, we have worked on two variants of greedy graph growing. The first one is called *double greedy graph growing* (DG). Its principle is to pick two seed vertices as far as possible among the halo, and to make parts A and B grow from them, with attention paid to keep halo balanced among the growing parts. The second approach, called *halo-first greedy graph growing* (HF), works in a first stage on the sole halo graph, finding a separator of it. Once it is done, it defines the two halo parts A_h and B_h as two sets of seeds and make these sets grow in the whole graph to build A and B. Finally, we have also changed the Fiduccia-Mattheyses refinement algorithm (FM) in order to preserve the good balancing in the finer graphs. Our algorithms will be explained more deeply during the presentation.

We made tests on a pool of 30 graphs from 140,000 to over 10 millions vertices. We measured both halo and domain interior imbalance. On 16 domains, our algorithms achieve an average gain of 39% on the halo imbalance, while not degrading interior imbalance. We increased the number of domains up to 512 on our biggest graphs and still got very good gains, in particular with HF. More detailed results will be given in the presentation.

References

[1] C.M. Fiduccia and R. M. Mattheyses. A linear-time heuristic for improving network partitions. In *19th Conference on Design Automation*, pages 175–181, June 1982.

[2] A. George. Nested dissection of a regular finite element mesh. *SIAM Journal on Numerical Analysis*, 10(2):345–363, 1973.

[3] B. Hendrickson and R. Leland. A multi-level algorithm for partitioning graphs. In *Supercomputing, 1995. Proceedings of the IEEE/ACM SC95 Conference*, pages 28–28, 1995.

[4] G. Karypis, V. Kumar, and Vipin Kumar. Multilevel k-way partitioning scheme for irregular graphs. *Journal of Parallel and Distributed Computing*, 48:96–129, 1998.

[5] R. J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16(2), 1979.

[6] F. Pellegrini and J. Roman. Scotch: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. In *High-Performance Computing and Networking*, volume 1067 of *Lecture Notes in Computer Science*, pages 493–498. Springer, 1996.