Available online at www.prace-ri.eu

**Partnership for Advanced Computing in Europe**

# A Parallel Sparse Hybrid Solver and Its Relations to Graphs and Hypergraphs

Mustafa Gündoğan[a], Cevdet Aykanat[a], Murat Manguoğlu[b]

*[a]Bilkent University, Computer Engineering Department, 06800 Ankara, Turkey*
*[b]Middle East Technical University, Computer Engineering Department, 06800 Ankara, Turkey*

**Abstract**

In this whitepaper, we review the state-of-the-art hybrid solver, which uses generalized form DS factorization, for solving system of equations of the form $Ax = f$, and this solver's relations to graphs and hypergraphs. We investigate two different reordering strategies for the DS factorization preconditioning scheme: reordering via graph partitioning (GP) and reordering via hypergraph partitioning (HP).In the GP scheme, the partitioning objective of minimizing the edge cutsize corresponds to minimizing the total number of nonzeros in the off-diagonal blocks of the reordered matrix. In the HP scheme, the partitioning objective of minimizing the cutsize, according to the cut-net metric, corresponds to minimizing the total number of nonzero columns in the off-diagonal blocks of the reordered matrix. In both of the two schemes, partitioning constraint of maintaining balance on the part weights corresponds to maintaining balance on the nonzero counts of the diagonal blocks of the reordered matrix. The partitioning objective of GP relates to minimizing the number of nonzeros in the reduced system, whereas the partitioning objective of HP exactly models minimizing the size of the reduced system. We tested the performance of two partitioning schemes on a wide range of matrices for 4-, 8-, 16-, 32-, and 64-way permutations. Results showed that HP scheme performs better than GP scheme in terms of solution times.

## 1. Introduction

Given a system of equations of the form $\boldsymbol{Ax} = \boldsymbol{f}$, where $\boldsymbol{A}$ is large and sparse, it is known that hybrid solvers that contain both direct and iterative components are promising in terms of robustness and scalability on parallel computing platforms. A state-of-the-art hybrid solver, which uses the generalized form parallel $\boldsymbol{DS}$ factorization**,** is the focus of this work [1, 2]. In the $\boldsymbol{DS}$ factorization scheme, $\boldsymbol{D}$ is the block diagonal of $\boldsymbol{A}$, and the factor $\boldsymbol{S}$, given by $\boldsymbol{D}^{-1}\boldsymbol{A}$ (assuming $\boldsymbol{D}$ is nonsingular), consists of the block diagonal identity matrix modified by "spikes" to the right and left of each partition. The generalized $\boldsymbol{DS}$ factorization of the system involves reordering $\boldsymbol{A}$ to extract the block diagonal $\boldsymbol{D}$, then, multiplying both sides of the system with $\boldsymbol{D}^{-1}$, from the left side. The resulting multiplied system contains a smaller reduced system of equations according to nonzero entries in off-diagonal blocks of the reordered system. The solution of the original system, $\boldsymbol{Ax} = \boldsymbol{f}$, can be constructed from the solution of this reduced system, which can be solved independently. After ***DS*** factorization, the process of solving $\boldsymbol{Ax} = \boldsymbol{f}$ reduces to a sequence of steps that are ideally suited for a parallel execution. In general, the scalability of this factorization scheme depends on decreasing the solution time of the reduced system. Among other factors, the solution time of the reduced system depends on the size and the number of nonzeros of the reduced system.

In this work, two different reordering strategies are investigated for a successful DS factorization preconditioning scheme: reordering via graph partitioning (GP) and reordering via hypergraph partitioning (HP). In the GP scheme, the standard graph representation $G(\boldsymbol{A})$ of matrix $\boldsymbol{A}$ is used. In the HP scheme, the column-net hypergraph model [3] $H_{cn}(\boldsymbol{A})$ of matrix $\boldsymbol{A}$ is used.

## 2. Work Done

For a $K$ processor system, in the GP and HP schemes, a $K$-way partitioning is performed on $G(A)$ and $H(A)$, respectively, and the resulting partition is decoded as inducing a $K$-way symmetric permutation on the rows and columns of $A$ [3].

In the GP scheme, the partitioning objective is to minimize the edge cutsize. This objective corresponds to minimizing the total number of nonzeros in the off-diagonal blocks of the reordered matrix. In the HP scheme, the partitioning objective is to minimize the cutsize according to the cut-net metric. This objective corresponds to minimizing the total number of nonzero columns in the off-diagonal blocks of the reordered matrix. In both schemes, the partitioning constraint is maintaining balance on the part weights. This strategy allows us to maintain balance on the nonzero counts of the diagonal blocks of the reordered matrix.

The partitioning objective of the GP scheme relates to minimizing the number of nonzeros in the reduced system, whereas the partitioning objective of the HP scheme exactly models minimizing the size of the reduced system. Hence, the HP scheme can be expected to achieve better preconditioning compared to the GP scheme.

## 3. Results Obtained

Inthis project, the experimental performance comparison of the proposed GP and HP schemes for preconditioning with DS factorization are investigated by using the successful multi-level graph and hypergraph partitioning tools MeTiS and PaToH [3] on various matrices selected from the University of Florida sparse matrix collection [4]. We have tested these two partitioning schemes for 4-, 8-, 16-, 32-, and 64-way permutations. The biconjugate gradient stabilized (BiCGStab) solver is used as an iterative solver for both inner and outer systems, whereas PARDISO is used as a direct solver. The target parallel architecture is an Intel cluster of 46 nodes, where each node contains 2 Intel Xeon E5430 Quad-Core CPUs.Table 1 displays the average performance improvement of the HP and GP schemes over the unordered scheme,in terms of solution times for different $K$ values.

| K | # of matrices | GP | HP | HP/GP |
|---|---|---|---|---|
| 4 | 173 | 49.12% | 52.46% | 0.93 |
| 8 | 132 | 53.68% | 55.70% | 0.96 |
| 16 | 102 | 55.05% | 61.72% | 0.85 |
| 32 | 86 | 56.90% | 64.52% | 0.82 |
| 64 | 51 | 57.15% | 68.20% | 0.74 |

Table 1: Performance improvement of HP over GP in terms of solution times for different $K$ values.

As seen in Table 1, HP and GP schemes achieve 50-70% improvement in the solution times for different problem categories on average. The last column of Table 1 displays the ratio of the solution times of HP and GP schemes averaged over problem categories. Values smaller than one indicate the categories where HP scheme performs better than the GP scheme on average. As seen in the last column of Table 1, the HP scheme performs considerably better than the GP scheme for different *K* values.

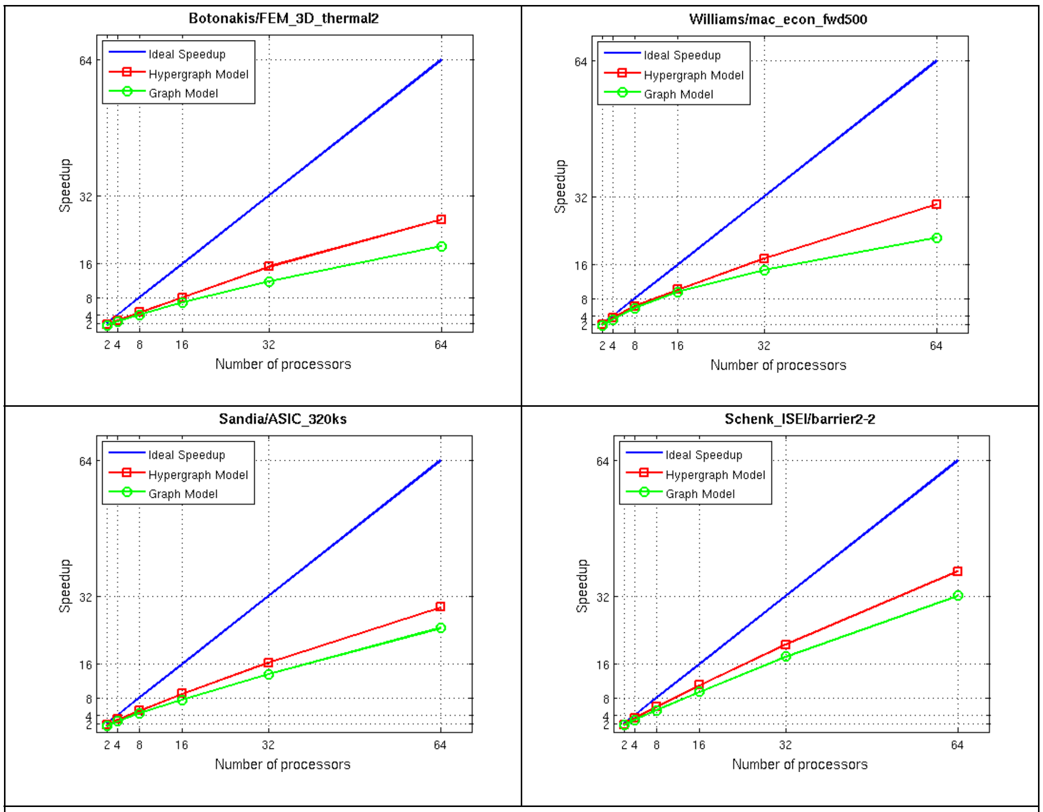

Figure 1: Speedup curves for the solution of four different linear systems on a 64-processor system.

Figure 1 displays speedup curves for the solution of four different linear systems selected from the list of matrices used in the experiments. As seen in the figure, HP achieves considerably better speedup than GP and with increasing number of processors the performance gap slightly increases in favor of HP.

## 4. Conclusion

We reviewed the state-of-the-art hybrid solver for solving system of equations of the form $Ax = f$. We proposed two reordering strategies, GP and HP, as a preconditioning scheme for the hybrid solver. The objective of the preconditioning is to minimize the size of the reduced system in order to achieve the solution faster. While GP scheme gives an approximation to this objective as GP minimizes the number of nonzero in the reduced system, HP scheme exactly models the minimizing the size of the reduced system. We compared the results of GP and HP schemes in terms of system solution times. Our experiments conducted on a wide range of matrices showed that the HP scheme performs better than the GP scheme.

## Acknowledgements

The HPC resources used in this work were provided by the Department of Computer Engineering, Middle East Technical University, Turkey.

## References

1. M. Manguoglu, M. Koyuturk, A. H. Sameh, and A. Grama,Weighted matrix ordering and parallel banded preconditioners for iterative linear system solvers, SIAM J. Scientific Computing, vol. 32, no. 3, pp. 1201–1216, 2010.
2.M. Manguoglu, A. Sameh, and O. Schenk, A parallel hybrid sparse linear system solver, LNCS - Proceedings of EURO-PAR09, vol. 5704, pp. 797–808, 2009.
3. U. V. Catalyurek and C. Aykanat, Hypergraph-Partitioning-Based Decomposition for Parallel Sparse-Matrix Vector Multiplication , IEEE Transactions on Parallel and Distributed Systems, vol. 10, no. 7, pp. 673-693, 1999.
4.T. A. Davis, University of Florida sparse matrix collection, NA Digest, 1997.