Scalable Distributed Schur Complement Methods for CFD Simulation on Many-Core Architectures

List of authors:

A. Basermann [1]

H.-P. Kersken [2]

At the German Aerospace Center (DLR), the parallel simulation systems TAU [1] and TRACE [2] have been developed for the aerodynamic design of aircrafts or turbines for jet engines. Large-scale computing resources allow more detailed numerical investigations with bigger numerical configurations. Up to 50 million grid points in a single simulation are becoming a standard configuration in the industrial design. Both CFD solvers, TAU and TRACE, require the parallel solution of large, sparse real or complex systems of linear equations.

For the parallel iterative solution of sparse equation systems to be solved within the TAU or TRACE solvers, FGMRes [3] with Distributed Schur Complement (DSC) preconditioning [4] for real or complex matrix problems has been investigated.

The DSC method requires adaquate partitioning of the matrix problem since the order of the approximate Schur complement system to be solved depends on the number of couplings between the sub-domains. Graph partitioning with ParMETIS [5] from the University of Minnesota is suitable since a minimization of the number of edges cut in the adjacency graph of the matrix corresponds to a minimization of the number of the coupling variables between the sub-domains. The latter determine the order of the approximate Schur complement system used for preconditioning.

Matrix permutations like Reverse Cuthill-McKee (RCM), Minimum Degree (MD) or Nested Dissection are employed per sub-domain in order to reduce fill-in in incomplete factorizations which are part of the DSC preconditioner. These reordering methods together with a threshold strategy for incomplete factorizations decrease the costs of local approximate LU decompositions within the DSC method (ILUT).

For the solution of TAU or TRACE linear equation systems, we developed a parallel iterative FGMRes algorithm with DSC preconditioning. In [6] and [7], we demonstrated the superiority of the DSC preconditionier over (approximate) block-Jacobi preconditioning. Block-local preconditioning methods like block-Jacobi are the standard preconditioners in TRACE. In [8], we described and demonstrated the effect of matrix (re-)partitioning and reordering on the DSC algorithm in detail.

In the following, we particularly discuss numerical and performance results of DSC methods for typical complex TRACE CFD problems on many-core architectures.

[1] German Aerospace Center (DLR), Simulation and Software Technology, Porz-Wahnheide, Linder Höhe, Cologne, Germany

[2] German Aerospace Center (DLR), Simulation and Software Technology, Porz-Wahnheide, Linder Höhe, Cologne, Germany

Fig. 1 displays execution times of the real and complex DSC solver software developed on 4 to 32 processor cores of DLRs AeroGrid cluster (45 Dual-processor nodes; Quad-Core Intel Harpertown; 2.83 GHz; 16 GB main memory per node; InfiniBand interconnection network between the nodes) for a complex and the corresponding real small TRACE matrix problem of order 28,120 or 56,240. METIS Nested Dissection was used for reordering of the local diagonal blocks. ILUT with threshold 10^{-3} was applied for factoring the local blocks. The FGMRes iteration was stopped when the current residual norm divided by the initial residual was smaller than 10^{-5}. The complex DSC solver version distinctly outperforms the real version

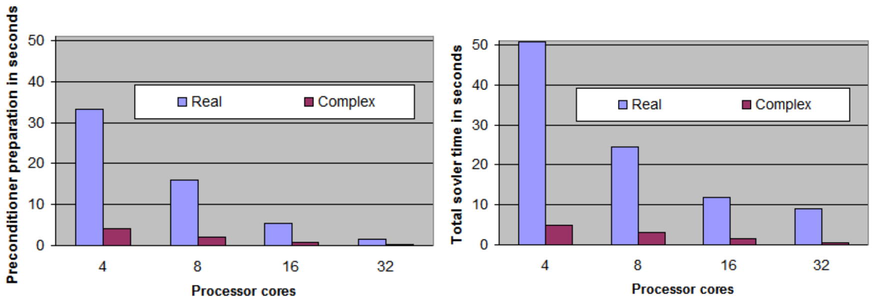

Figure 1: Execution times of the DSC preconditioner preparation (left) and the total DSC solver (right) for a complex and the corresponding real small TRACE matrix problem on a many-core cluster.

both for preconditioner preparation and for the total DSC solver. This is caused by the lower problem order and a more advantageous matrix structure in the complex case opposite to the real case. In addition, the complex formulation results in higher data locality (storage of real and imaginary part in adjacent memory cells) and a better ratio of computation to memory access due to complex arithmetics in comparison with the real formulation. Fig. 1 also shows an advantageous strong scaling behavior of the DSC method on the many-core cluster, even for this small matrix problem.

Fig. 2, left, illustrates the strong scaling behavior of the total complex DSC method on 4 to 192 processor cores of the AeroGrid cluster for a medium size complex TRACE matrix problem of order 378,400 and with 45,456,500 non-zeros.

Fig. 2, right, shows execution times of the total complex DSC method on 32 to 192 processor cores of the AeroGrid cluster for a large complex TRACE matrix problem of order 4,497,520 and with 552,324,700 non-zeros. In order to achieve the required accuracy in each component of the solution vector the FGMRes iteration was here stopped when the current residual norm divided by the initial residual was smaller than 10^{-10}.

Figs. 1 and 2 demonstrate that the complex DSC algorithm scales very well on many-core clusters for differnt CFD matrix problems of varying size.

In addition to the TRACE results shown above, numerical and performance experiments will

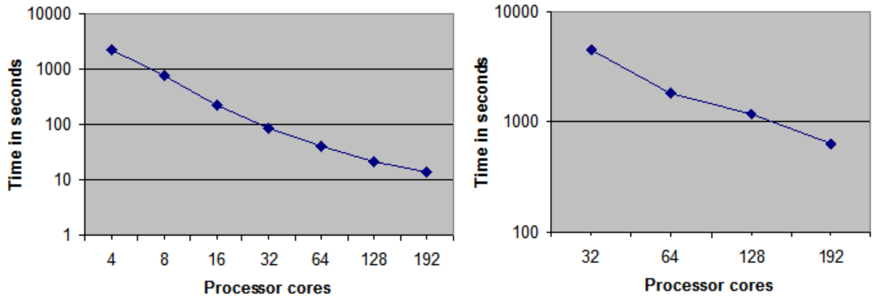

Figure 2: Execution times of the total complex DSC solver for a complex medium size TRACE matrix problem of order 378,400 (left) and for a complex large TRACE matrix problem of order 4,497,520 (right) on a many-core cluster.

be presented with the real version of the DSC method developed for typical real TAU CFD problems on many-core architectures.

References

[1] `http://www.dlr.de/as/`

[2] `http://www.dlr.de/at/desktopdefault.aspx/tabid-1519/2123_read-3615/`

[3] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM J. Sci. Statist. Comput.* 14:461–469, 1993.

[4] Y. Saad, M. Sosonkina. Distributed Schur complement techniques for general sparse linear systems. *SISC* 21:1337–1356, 1999.

[5] G. Karypis, V. Kumar. ParMETIS: Parallel graph partitioning and sparse matrix ordering library. Tech. rep. # 97-060, University of Minnesota, 1997.

[6] A. Basermann, F. Cortial-Goutaudier, U. Jaekel, K. Hachiya. Parallel solution techniques for sparse linear systems in circuit simulation. In: *Proceedings of the 4th International Workshop on Scientific Computing in Electrical Engineering, Series: Mathematics in Industry*, Springer, 2002.

[7] T. Alrutz, P. Aumann, A. Basermann. HICFD – Hocheffiziente Implementierung von CFD-Codes für HPC-Many-Core-Architekturen. In: *Mitteilungen - Gesellschaft fr Informatik e. V., Parallel-Algorithmen und Rechnerstrukturen*, ISSN 0177 - 0454, pp. 27–35, 2009.

[8] A. Basermann, U. Jaekel, M. Nordhausen, K. Hachiya. Parallel iterative solvers for sparse linear systems in circuit simulation. *Future Generation Computer Systems*, 21:1275–1284, 2005.