A hybrid domain decomposition type of preconditioner for Jacobi-Davidson on modern hardware

Menno Genseberger - Deltares and CWI, The Netherlands

The Jacobi-Davidson method [1] is an iterative method suitable for computing solutions of large eigenvalue problems. For the computation of a solution $(\lambda, \mathbf{x})$ to the standard eigenvalue problem $\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}$, in each iteration, this method extracts an approximate solution $(\theta, \mathbf{u})$ from a search subspace, corrects the approximate eigenvector $\mathbf{u}$ by computing a correction vector from a correction equation, and uses the correction vector to expand the search subspace. Most computational work of Jacobi-Davidson is due to the correction equation. In [2, 3] a strategy for the computation of (approximate) solutions of the correction equation was proposed. The strategy is based on a domain decomposition technique [4, 5] in order to reduce wall-clock time and local memory requirements.

The use of a preconditioner based on this domain decomposition technique leads effectively to almost un-coupled subproblems at subdomains: an ideal situation for implementation on parallel computers with distributed memory. However, hardware tends to have more and more computational cores that share memory. At the moment, modern harware consists of nodes with tens of cores, whereas nodes are connected with fast network interconnects. So, on one hand memory is shared by cores within one node and on the other memory is distributed over nodes. Distributed memory implementations with MPI of the original strategy can still be used to distribute computational work over cores within the shared memory of a single node [6, 7]. However, because of the increasing number of cores within single nodes, memory access is becoming more and more the major bottleneck instead of computational capacity. Multiple cores share the same memory and simultaneous memory access, which is likely to occur by the original strategy, will slow down the overall wall-clock time.

Therefore, in the present talk we will show how we adapted the original strategy for modern hardware by incorporating an efficient implementation of one of the building blocks. For this we used a banded direct solver optimized for multiple cores with OpenMP wihin shared memory from PARDISO [8, 9]. By numerical experiments we will illustrate the balance between the distributed and shared part of this new hybrid approach. For this we consider different computational and memory entities (nodes, sockets, cores, threads, L1/L2/L3 caches).

References

[1] Sleijpen, G. L. G., van der Vorst, H. A.: A Jacobi-Davidson iteration method for linear eigenvalue problems. SIAM J. Matrix Anal. Appl. 17, 401–425 (1996)

[2] Genseberger, M., Sleijpen, G. L. G., van der Vorst, H. A.: An optimized Schwarz method in the Jacobi-Davidson method for eigenvalue problems. In: Domain Decomposition Methods in Science and Engineering, pp. 289–296. UNAM, Mexico (2003)

[3] Genseberger, M.: Domain decomposition in the Jacobi-Davidson method for eigenproblems. Ph.D. thesis, Utrecht University (2001)

[4] Tan, K. H., Borsboom M. J. A.: On generalized Schwarz coupling applied to advection-dominated problems. In: Domain Decomposition Methods in Scientific and Engineering Computing, pp. 125–130. American Mathematical Society, Providence (1994)

[5] Tan, K. H.: Local coupling in domain decomposition. Ph.D. thesis, Utrecht University (1995)

[6] Genseberger, M.: Improving the parallel performance of a domain decomposition preconditioning technique in the Jacobi-Davidson method for large scale eigenvalue problems. Appl. Numer. Math. 60, 1083–1099 (2010)

[7] M. Genseberger, Practical aspects of domain decomposition in Jacobi-Davidson for parallel performance, In: Domain Decomposition Methods in Science and Engineering XXI, pp. 840–847. Springer (2014)

[8] Kuzmin, A., Luisier, M, Schenk, O.: Fast methods for computing selected elements of the Greens function in massively parallel nanoelectronic device simulations. In: Lecture Notes in Computer Science Vol. 8097, pp. 533–544. Springer (2013)

[9] Schenk, O., Gärtner, K.: Solving unsymmetric sparse systems of linear equations with PARDISO. J. Future Generation Computer Systems 20, 475–487 (2004)