AN OPTIMIZED CHEBYSHEV SMOOTHER IN GAMG SOLVER OF OPENFOAM ON SUNWAY TAIHULIGHT SUPERCOMPUTER

HANFENG GU[1], HU REN[1], CHANGXI LIU[1,2], WEI XUE[1,3], XIN LIU[1,4]
[1]*National Supercomputing Center in Wuxi, Wuxi 214072, China, hanfenggu@gmail.com*
[2]*Beihang University, Beijing 100191, China*
[3]*TsingHua University, Beijing 100084, China*
[4]*Wuxi Jiangnan Institute of Computing Technology, Wuxi 214000, China*

Keywords: *MultiGrid(GAMG), Chebyshev, Gauss-Seidel, Sunway TaihuLight supercomputer, accelerating*

Abstract

The Sunway TaihuLight supercomputer is the first system with a peak performance greater than 100 PFlop/s and has been the fastest computer in the world since June 2016[1, 2]. The computer node of Sunway system is based on a homegrown heterogeneous many-core processor called SW26010, which consists of 260 processing elements that including both the 4 management processing elements (MPEs) and 256 computing processing elements (CPEs). OpenFOAM is a leading open source software for Computational Fluid Dynamics(CFD) but not fully compatible with processor SW26010 since its heterogeneity. Each CPE has its own local device memory (LDM) space and one needs control manually the data on each CPE's LDM to take advantage of powerful accelerating ability provided by these CPEs. Some efforts have been paid to optimize the hot-spots of OpenFOAM on SW26010 and achieve significant performance improvement. In some cases, the performance of the CPE cluster on SW26010 is better than that on a single core of Intel(R) Xeon(R) CPU E5-2695 v3.[3].

GAMG solver in OpenFOAM

GAMG (Geometric agglomerated Algebraic MultiGrid) algorithm (see Figure 1) is the main solver in OpenFOAM, which is usually used to solve the pressure correction Poisson equation. According to the profiling results, the smoother usually takes more than half in GAMG solving time. The only default smoother in the latest OpenFOAM version 5.0 is Gauss-Seidel, which is difficult to maintain good parallel efficiency in the context of unstructured meshes due to its natural sequentiality.[4]. In OpenFOAM, the Gauss-Seidel smoother is designed to be as the hybrid of Jacobi-type iteration for processor boundary points and real Gauss-Seidel-type iteration for processor inner points, which make its convergence path depend on the participation of the matrix and even to diverge if the problem size per processor is not large enough[5]. The worse situation can be found on SW26010 if we want to take advantage of powerful accelerating ability provided by the CPEs. The data have to be assigned to CPEs and thus the communication between CPEs become very complicated.

Chebyshev smoother in OpenFOAM

Polynomial smoothers become the nature choice in modern heterogeneous parallel computing system for a couple of reasons. Firstly, they don't need to compute communication-intensive inner products for the determination of the recurrence coefficients, and they only need the matrix-vector multiplication, which is often highly-optimized. Secondly, they are unaffected by the parallel partitioning of the matrix, the number of parallel processes, and the ordering of the unknowns[5]. The main drawback is the cost of computing the upper and lower bounds of eigenvalues of the matrix.
To the authors' knowledge, the polynomial type smoother in OpenFOAM has not been implemented. In this paper, we present an implementation of a polynomial smoother in OpenFOAM: the Chebyshev smoother[6] combined with the Preconditioned Conjugate Gradient (PCG) solver. Figure 2 shows the algorithm, where the PCG loops are used to obtain the largest eigenvalues since it has same upper bound as for the Chebyshev iteration in the symmetric case[7, 8]. While the lower bound of eigenvalues in Chebyshev iteration is not so important since the smoother of MultiGrid in each level only eliminates the errors compared to the local mesh size. One can simply divide the largest eigenvalue by a constant value to obtain the lower bound of eigenvalue in that MultiGrid level. The MultiGird convergence does not seem very sensitive to this estimate[4].

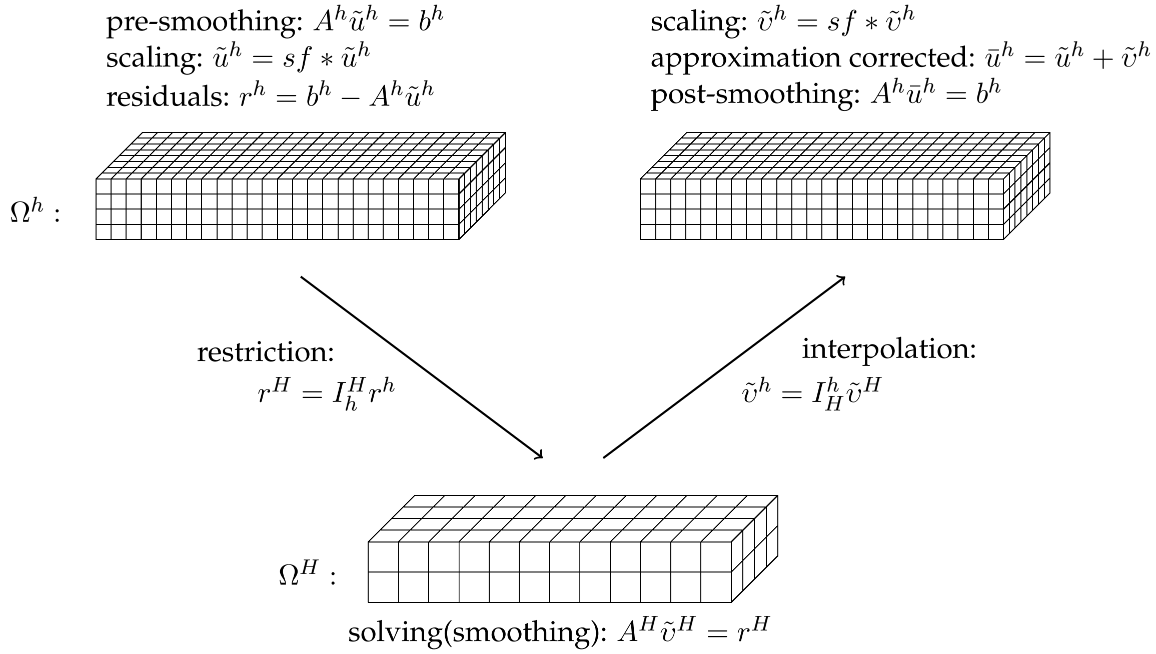

Figure 1: A two-level GAMG example in OpenFOAM.

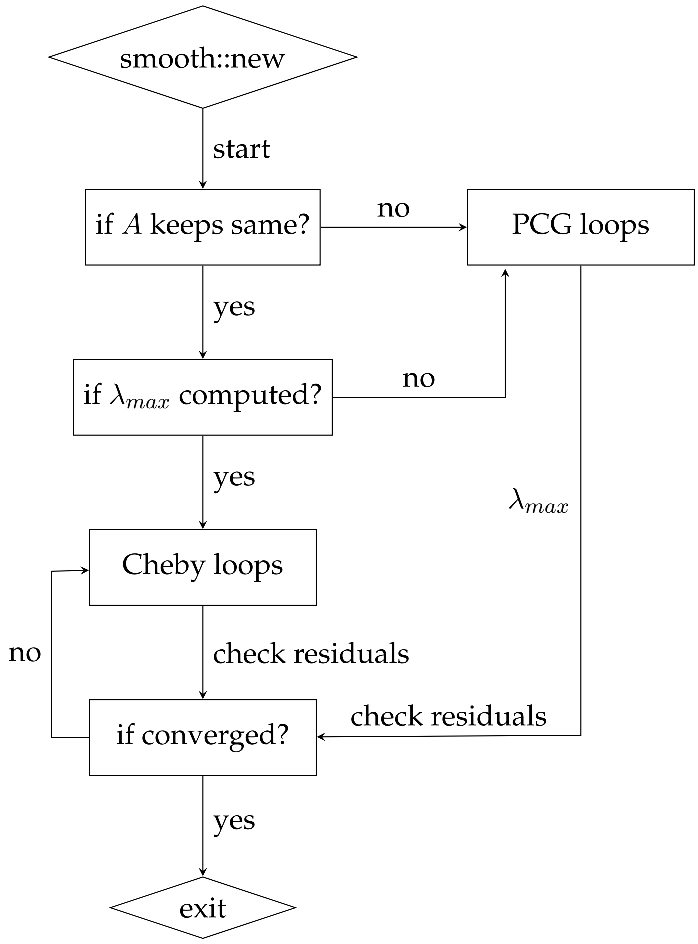

Figure 2: Chebyshev smoother combined with PCG in OpenFOAM.

Implementations and Results

The advantages of Chebyshev smoother make it easier to be fully parallelized compared to Gauss-Seidel smoother. The kernel in Chebyshev smoother is the matrix-vector multiplication, which has been already fully accelerated in previous work done by C. Chang[9]. For the rest of continuous vector operations, we proposed a unified accelerating interface to involve all of the rest kernels. To improve the efficiency of obtaining the largest eigenvalue, we also modified the implementation of PCG to reduce the global synchronization according the algorithm proposed in[10]. The Table 1 shows that the optimized Chebyshev smoother is 3.55x faster the Gauss-Seidel smoother.

Table 1: Results comparison between Gauss-Seldel and Chebyshev smoother in a simpleFOAM case

	Gauss-Seidel	Chebyshev
Mesh	50 million, unstructured	
MPIs	256	256
time (seconds)	70.49s	19.88s

References

[1] H. Fu, J. Liao, J. Yang, L. Wang, Z. Song, X. Huang, C. Yang, W. Xue, F. Liu, F. Qiao *et al.*, "The sunway taihulight supercomputer: system and applications," *Science China Information Sciences*, vol. 59, no. 7, p. 072001, 2016.

[2] J. Dongarra, "Sunway taihulight supercomputer makes its appearance," *National Science Review*, vol. 3, no. 3, pp. 265–266, 2016.

[3] D. Meng, M. Wen, J. Wei, and J. Lin, "Hybrid implementation and optimization of openfoam on the sw26010 many-core processor," 2016.

[4] M. Adams, M. Brezina, J. Hu, and R. Tuminaro, "Parallel multigrid smoothing: polynomial versus gauss–seidel," *Journal of Computational Physics*, vol. 188, no. 2, pp. 593–610, 2003.

[5] A. H. Baker, R. D. Falgout, T. V. Kolev, and U. M. Yang, "Multigrid smoothers for ultraparallel computing," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2864–2887, 2011.

[6] M. H. Gutknecht and S. Röllin, "The chebyshev iteration revisited," *Parallel Computing*, vol. 28, no. 2, pp. 263–283, 2002.

[7] R. Barrett, M. W. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. Van der Vorst, *Templates for the solution of linear systems: building blocks for iterative methods*. Siam, 1994, vol. 43.

[8] J. A. Scales, "On the use of conjugate gradient to calculate the eigenvalues and singular values of large, sparse matrices," *Geophysical Journal International*, vol. 97, no. 1, pp. 179–183, 1989.

[9] C. Liu, B. Xie, X. Liu, W. Xue, H. Yang, and X. Liu, "Towards efficient spmv on sunway many-core architectures," in *Proceedings of the 32th ACM on International Conference on Supercomputing, ICS'18, Beijing, China, June 12 - 15, 2018*, 2018.

[10] E. D'Azevedo, V. Eijkhout, and C. Romine, "Conjugate gradient algorithms with reduced synchronization overhead on distributed memory multiprocessors," 1999.