Effects of Blocking and Re-Ordering of Matrix Index in a Parallel Linear Iterative Solver of FEM Application Development Support Middleware

Takeshi Kitayama[1, 3], Olav Aanes Fagerlund[2] and Hiroshi Okuda[1, 3]

[1] Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5 Kashiwanoha, Kashiwa, Chiba 277-8568, Japan
[2] Department of Systems Innovation, School of Engineering, The University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo 113-8656, Japan
[3] Japan Science and Technology Agency, CREST, 7, Gobancho, Chiyoda-ku, Tokyo, 102-0076, Japan
kitayama@multi.k.u-tokyo.ac.jp, olav@multi.k.u-tokyo.ac.jp, okuda@k.u-tokyo.ac.jp

Abstract. A development support middleware implementing the Finite Element Method (FEM) is developed. To improve the performance of a hybrid parallelized linear iterative solver, the effect of utilizing blocking algorithms within loops implementing the sparse matrix vector product (SpMV) operation is tested. An incomplete Cholesky matrix preconditioner is parallelized using OpenMP. It includes multicolor reordering of the matrix index in order to remove loop dependency. By changing the number of colors used in the reordering, the locations of the non-zero matrix elements are shifted. This causes performance differences. The effects of this blocking and reordering of the matrix is discussed.

Keywords: Finite Element Method, Middleware, Linear Iterative Solver

1 Introduction

We are developing the ppOpen-APPL/FEM middleware, which provides APIs for the Finite Element Method [1]. Here, the linear equation solver is the most time consuming step of the FEM structural analysis. In this study, performance evaluation of the OpenMP parallelized iterative linear equation solver is carried out. As such, the effect of blocking and re-ordering of the matrix index in a Parallel Linear Iterative Solver is investigated.

2 Calculation Methods, Results and Discussion

All calculations are carried out on the FX10 supercomputer at Information Technology Center, The University of Tokyo. Each calculation node has 16 cores and 16 OpenMP threads are used in the intra node parallelization. MPI is used in inter node parallelization. The calculation model is a hinge made of metal. Second order tetrahedral element is used for discretization. The number of degrees of freedom is 252,138. Linear static structural analysis is carried out and a profiler measures elapsed time for the convergence of the iterative CG solver. The block incomplete Cholesky level 0 preconditioner is used [2]. Multicolor reordering of the matrix index is carried out to make OpenMP parallelization possible [3]. Effect of different blocking strategies for the assignment of loops to an OpenMP thread, and number of colors for multicolor re-ordering for the re-ordering of the matrix index are tested.

Table 1 shows the elapsed time of the preconditioner, sparse matrix vector product (SpMV) and other routines in the iterative solver when using a single calculation node. Two kinds of blocking strategies of loops and three kinds of coloring schemes for multicolor re-ordering is tested and compared with serial base code. Among these combinations, the blocking algorithm using equivalent number of non-zero elements with the number of colors set to 10 while using 16 threads gives 12.22times faster speed-up to serial base code and is the best performer. The performance depends on both the algorithm and the number of colors in use, as re-ordering changes the non-zero matrix element pattern and thus affects the memory access efficiency.

Table 1 Elapsed time of the preconditioner, SpMV and other routines in the iterative solver, for different blocking algorithms and different number of colors used for the reordering.

Blocking algorithm (number of threads)	Number of colors	Elapsed time (sec)			
		SpMV	Preconditioner	Other	Total
Base code (serial)	-	187.75	278.65	3.89	470.29
Block decomposition of row (16)	10	20.18	34.54	4.00	58.72
	100	30.30	30.63	4.19	65.12
	1000	31.98	50.72	3.99	86.69
Equivalent number of non-zero matrix elements (16)	10	18.50	15.99	4.00	38.49
	100	17.94	23.76	4.19	45.89
	1000	17.61	155.28	3.99	176.87

References

1. Nakajima, K. : ppOpen-HPC: open source infrastructure for development and execution of large-scale scientific applications on post-peta-scale supercomputers with automatic tuning (AT). ATIP '12 Proceedings of the ATIP/A*CRC Workshop on Accelerator Technologies for High-Performance Computing: Does Asia Lead the Way? Article No. 24

2. Benzi, M. : Preconditioning Techniques for Large Linear Systems: A Survey. Journal of Computational Physics 182, 418–477 (2002)

3. Doi, S., Wahio, T. : Ordering strategies and related techniques to overcome the trade-off between parallelism and convergence in incomplete factorizations. Parallel Computing 25 1995-2014 (1999)