Optimization of serial and parallel communications towards scalable algorithms on supercomputers in next generation

Kengo Nakajima
Information Technology Center, The University of Tokyo
nakajima@cc.u-tokyo.ac.jp

1. Introduction

Optimization of both of serial and parallel communications is a critical issue for development of scalable algorithms on the supercomputer systems in next generation. The *serial* communication is the data transfers through memory hierarchies of each processor, while the parallel communication is by message passing between computing nodes through the network using MPI. A multigrid is a scalable method for solving linear equations and preconditioning Krylov iterative linear solvers, and is especially suitable for large-scale problems. The *parallel* multigrid method is expected to be one of the powerful tools on post-peta/exa-scale systems. Recently, HPCG (High Performance Conjugate Gradients) [1] was proposed as a new benchmark for evaluation of the practical performance of supercomputer systems. HPCG solves sparse matrices derived from finite-element application using conjugate gradient linear solver (CG) preconditioned with multigrid method (MGCG).

The parallel multigrid method and MGCG include both of serial and parallel communication processes which are generally expensive. This article summarizes recent efforts of optimization of serial and parallel communications in parallel MGCG solvers with geometric multigrid procedures using up to 4,096 nodes (65,536 cores) of Fujitsu PRIMEHPC FX10 [2]. Target application, *pGW3D-FVM*, is a 3D finite-volume simulation code, which solves groundwater flow problems through heterogeneous porous media, by parallel MGCG method [2]. Performance of both of *flat MPI* and *HB M×N* (*M*: number of threads on each MPI process, *N*: number of MPI processes on each node) has been evaluated.

2. Summary and Results of Optimization

In the present work, new format for sparse matrix storage based on *sliced* ELL [3] (Fig.1), which has been well-utilized for optimization of SpMV, is proposed for optimization of serial communication on memories, and hierarchical coarse grid aggregation (*h*CGA) (Fig.2) is introduced for optimization of parallel communication by message passing. The proposed methods are implemented for pGW3D-FVM, and the robustness and performance of the code was evaluated using up to 4,096 nodes (65,536 cores) of the Fujistu FX10 system. The parallel MGCG solver using the *sliced* ELL format provided performance improvement in both weak scaling (25%–31%) and strong scaling (9%–22%) compared to the code using the original ELL format. Moreover, *h*CGA provided excellent performance improvement in both weak scaling (1.61 times) and strong scaling (6.27 times) for *flat MPI* parallel programming model. *h*CGA was also effective for improvement of parallel communications (Fig.3(a,b)).

Effect of *sliced* ELL on serial communication was significant, while that of *h*CGA on parallel communication was not so impressive except for *flat MPI* cases. Because *h*CGA proved to be very effective for reducing overhead of coarse grid solver, that will also provide more significant effect on hybrid parallel programming models with larger number of nodes. Computational amount of coarse grid solver for each core of *flat MPI* is 256 (=16×16) times as large as that of HB 16×1. Therefore, *h*CGA is expected to be really effective for HB 16×1 with more than 2.50×10^5 nodes (4.00×10^6 cores) of Fujitsu FX10, where the peak performance is more than 60 PFLOPS.

3. Future Works

In the present work, we focused on optimization of communications with neighboring MPI processes at process boundaries by MPI_Isend/Irecv/Waitall as *parallel communication*, but overhead by collective communication (e.g. MPI_Allreduce) is more significant with more than

2.50×10^5 nodes. In order to reduce overhead by this type collective communication, communication avoiding/hiding type of algorithm combined with MPI functions for non-blocking collective communications supported in MPI 3.0 specification is also effective.

More efficient storage of *sliced ELL* arrays and optimization of *h*CGA function are the critical issues for future work. Especially, improvement of communication procedures for repartitioning of MPI processes is important. CGA and *h*CGA include a various types of parameters, and the optimum values of those were derived through empirical studies in the present work. Development of methods for automatic selection of these parameters [4] is also an interesting technical issue for future work. Optimum parameters can be estimated based on calculation of computational amounts, performance models, parameters of hardware, and some measured performance of the system. But it is not so straightforward. Because some of these parameters also make effects on convergence, construction of such methods for automatic selection is really challenging.

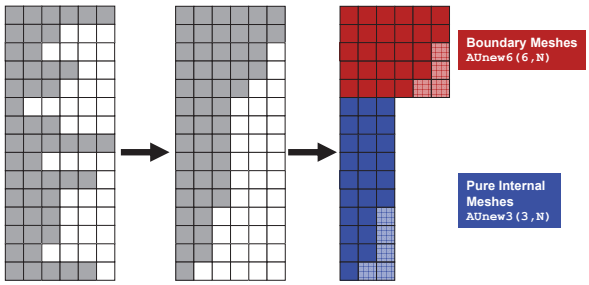

Fig.1 Idea of sliced ELL format [3], 2 slices in the present work

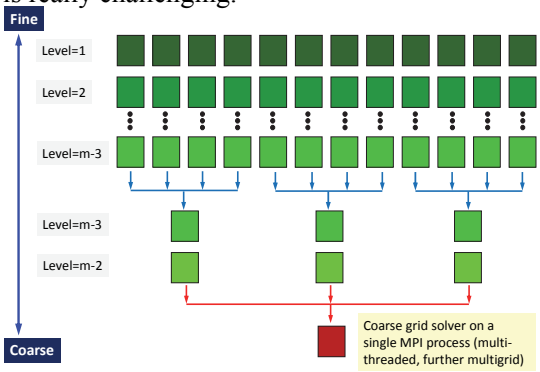

Fig.12. Procedures of hierarchical CGA (*h*CGA), where number of MPI processes is reduced before the final coarse grid solver of CGA on a single MPI process

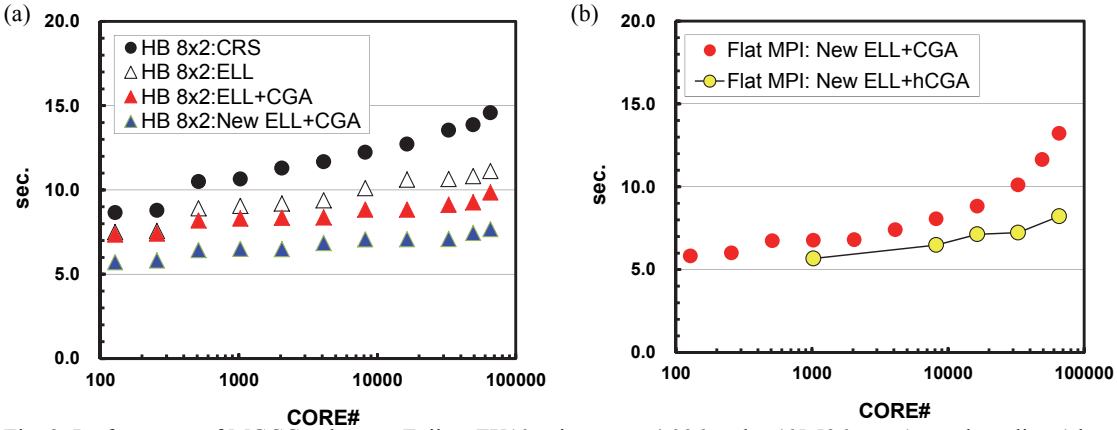

Fig. 3. Performance of MGCG solver on Fujitsu FX10 using up to 4,096 nodes (65,536 cores), weak scaling (elapsed time for MGCG): 262,144 ($=64^3$) meshes/core, max. total problem size: 17,179,869,184 meshes

References

[1] HPCG: High Performance Conjugate Gradients: https://software.sandia.gov/hpcg/

[2] Nakajima, K., Optimization of Serial and Parallel Communications for Parallel Geometric Multigrid Method, Proceedings of the 20th IEEE International Conference for Parallel and Distributed Systems (ICPADS 2014) (2014) 25-32

[3] Monakov, A., A. Lokhmotov, and A. Avetisyan, Automatically tuning sparse matrix-vector multiplication for GPU architectures, Lecture Notes in Computer Science 5952 (2010) 112-125

[4] Nakajima, K., Automatic Tuning of Parallel Multigrid Solvers using OpenMP/MPI Hybrid Parallel Programming Models, Lecture Notes in Computer Science 7851 (2013) 435-450