Performance Study of the Domain Decomposition Method with Many RHS Vectors and Explicit Shur Complement Computation

Tomonori Yamada* and Shinobu Yoshimura†

*Research into Artifacts, Center for Engineering (RACE)
The University of Tokyo
Kashiwanoha 5-1-5, Kashiwa, Chiba, Japan
e-mail: yamada@race.u-tokyo.ac.jp, web page: http://save.sys.t.u-tokyo.ac.jp

†Department of Systems Innovation, School of Engineering
The University of Tokyo
Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan
e-mail: yoshi@sys.t.u-tokyo.ac.jp,
web page: http://save.sys.t.u-tokyo.ac.jp, http//adventure.sys.t.u-tokyo.ac.jp

ABSTRACT

Today, high performance computing with iterative solvers on multi core architectures is one of the key issues in the field of computer science because current multi core architectures have a memory wall problem [1], and many iterative solvers of large linear equations suffer from this problem. One of the prominent approaches of iterative solvers for this memory wall problem is the implementation of the treatment of many RHS (Right Hand Side) vectors at a time. As the number of RHS vectors increases, the number of floating point operations for the matrix-vectors multiplication also increases, though the required memory access remains almost the same.

This paper reports a numerical benchmark result of the domain decomposition method [2], which is one of the most efficient parallel computing schemes for the finite element method, with the implementation of many RHS vectors on the latest many integrated core architecture, Intel Xeon Phi (Knights Landing). Furthermore, an explicit Shur complement computation is also introduced in our code for the use of level 3 BLAS function (dgemm). Finally, the achieved performance ratio to the theoretical peak performance is investigated.

ACKNOWLEDGEMENT

This work was supported in part by MEXT as a social and scientific priority issue (Accelerated Development of Innovative Clean Energy Systems) to be tackled by using post-K computer.

REFERENCES

[1] W. A. Wulf and S. A. McKee, “Hitting the memory wall: implications of the obvious”, *ACM SIGARCH computer architecture news*, Vol. **23** (1), pp.20-24, (1995).

[2] A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, Springer, 2005.