Co-desigining of FEM based CFD code FrontFlow/blue for the supercomputer Fugaku

Kiyoshi Kumahata
RIKEN Center for Computational Science
Kobe, Hyogo, Japan
kuma@riken.jp

Kazuo Minami
RIKEN Center for Computational Science
Kobe, Hyogo, Japan

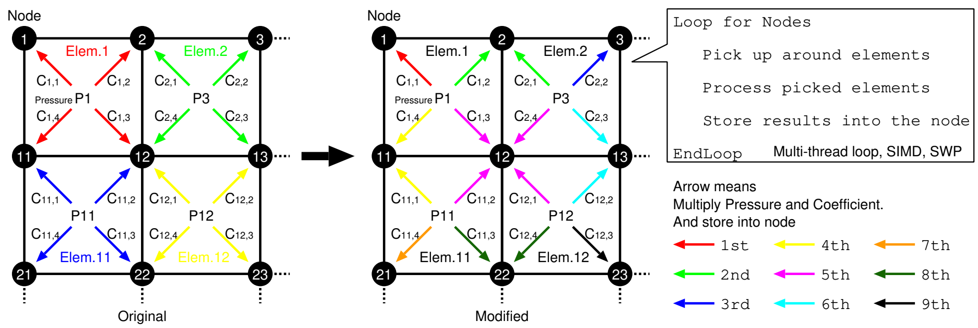

Figure 1: Schematic of store order modification

ABSTRACT

The fluid simulation software FrontFlow/blue (FFB) is vital to the manufacturing field in Japan, where it has played an important role, specifically during the K computer era. In addition, FFB software is a critical application for the supercomputer Fugaku. By co-designing it, we continue to improve the performance of the FFB software for Fugaku. Not SIMD load/store instructions are a bottleneck for the Fugaku because it has a large SIMD width. Moreover, indirect data storing using index arrays is inefficient; therefore, in the co-designing, we develop and apply methods to solve these issues. As a result, we obtained a tuned application with approximately twice faster CPU performance than before, on the K computer. On Fugaku, the application without tuning shows almost twenty times faster performance than on the K computer; therefore, the tuned application is expected to achieve approximately forty times faster performance than on the K computer.

CCS CONCEPTS

• **Software and its engineering** → **Extra-functional properties**; • **General and reference** → *General conference proceedings*.

KEYWORDS

Fluid Analysis, Performance Improvement, Supercomputer Fugaku, Finite Element Method, Store Order

1 INTRODUCTION

Recurrently, we attempt to improve the performance of FFB software [1]. In this paper, most significant methods that improve its performance are shown. In the remaining part of this paper, we briefly introduce the methods that show significant improvement in FFB software.

2 RANDOM STORE SEQUENTIALIZE FOR FUGAKU

Figure1 is a schematic of the improvement of major subroutines that calculates the pressure gradient from the element's pressure P_n and vertex coefficient $C_{i,j}$. The colored arrow signifies calculation and storing result into a node, and the arrow color represents its order. In the original procedure, elements are processed in sequential order, iterating them in a loop. Firstly, storing is in four different nodes 1, 2, 11, and 12 (red arrow). Secondly, storing is in nodes 2, 3, 12, and 13 (green arrow). In this manner, data storing in a node is repeated multiple times with an irregular cycle depending on the mesh structure. SIMD and software pipeline (SP) are not used because of their data dependency, yet a node is referred to from multiple elements. This degrades the performance of SIMD architectures with a large width, such as Fugaku. To solve this, we developed a method that sequences random storing. In the modified procedure, nodes are processed in sequential order. The procedure is implemented in a loop iterating between the nodes. Firstly, storing is done on node 1 (red arrow). Secondly, storing is done on node 2 (green arrow). In this manner, storing on a node is done once, and the cycle is sequential. This modification led to a double performance improvement on the K computer.

ACKNOWLEDGMENTS

Professor Chisachi Kato, the Institute of Industrial Science, The University of Tokyo. And Yoshinobu Yamade, Mizuho Information & Research Institute.

REFERENCES

[1] Kiyoshi Kumahata, Kazuo Minami, Yoshinobu Yamade, and Chisachi. 2018. Performance improvement of the general-purpose CFD code FrontFlow/blue on the K computer. In *HPC Asia 2018 Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region.* ACM, Tokyo, 171–182.