A Study of High Performance Parallelization for Incompressive Viscosity Flow Simulation using Finite Difference Method

Motoyoshi Kurokawa
School of Information Science,
Japan Advanced Institute of Science and Technology

January 10, 2002

Abstract

In this research, I examined three effective techniques for highly effective, parallel Computational Fluid Dynamics (CFD) simulation. I clarified a parallel performance in parallel CFD simulations which applied those techniques. It is insufficient to examine three methods individually to improve the efficiency of a parallel CFD simulation. These methods of each individual method were examined. However, the performance improvement using all techniques is not examined overall. To show the effectiveness of three techniques, I discussed the effectiveness of the individual technique. I applied three methods at the same time to the incompressible viscous flow simulation program which used the MAC(Maker And Cell) method. Each method used in this research can be achieved without a big change from the existing simulation program code and the linear solver algorithm. This is very useful for the efficiency improvement of a parallel CFD simulation, because the program change for the researcher in the CFD field is very easy to apply these method. I showed that the improvement of the parallel computation performance of the CFD simulation was actually high. I simulated incompressible viscous flows using the MAC method. The computational model targeted the pipe flow problem, the cavity flow problem and the flow around circular cylinder problem as a three-dimensional model. I studied overall performance and effectiveness of parallel simulation of incompressible viscous flows (included computation and convergence performance).

The method of the performance improvement of the parallel CFD calculation examined by this research is as follows. First, a selection method for better domain partitioning pattern. Second, a method of cutting down the communication time. Third, a Multi Colored Line Successive Over Relaxation (SOR) (MCLSOR) method for a scalar type parallel machine The selection method for better domain partitioning pattern predicts the performance change in the domain decomposition method by the partitioning pattern. In addition, I showed the selection method of the partitioning pattern for the performance maintenance. The method of cutting down the communication time is an overlapping communication with computation method (the systolic communication-computation overlap method). The systolic communication-calculation overlap method cut down the communication time in the parallel CFD simulation. I showed the actual improvement of the parallel performance. The MCLSOR method is an improvement method of the computation processing performance on a scalar parallel computer. The program code of the CFD simulation will execute a parallel computation more effectively, when the selection method and the systolic communication-computation overlap method, the MCLSOR method combine with the parallel computation.

The CFD simulation which used these methods obtained a high parallel performance. When the selection method of the best domain partitioning pattern was used, the best partitioning pattern was obtained by almost 100 %. The systolic communication-computation overlap method by which the best domain partitioning pattern was used obtained the improvement of the parallel performance about $15 \sim 30\%$. In the MCLSOR method, the serial version of the MCLSOR method obtained about 40 % higher performance against the Multi-color SOR mehtod, in addition, the performance decrease was not seen greatly in the parallel processing. Parallel efficiency improved when the systolic communication-computation overlap method was applied to the MCLSOR method. The parallel performances improve more than the conventional system about $2 \sim 3$ times, when these methods are applied to CFD simulation.

Key Words: **Computational Fluid Dynamics(CFD), Parallel CFD Simulation, Domain Decomposition method, Overlapping communication with computation, Modified SOR Method**