Application of GPUs in CFD-based Turbine Wake Simulation

Ji Qi
Kyushu University
Fukuoka, Japan
qi.ji.558@s.kyushu-u.ac.jp

Kenji Ono
Kyushu University
Fukuoka, Japan
ono.kenji.693@m.kyushu-u.ac.jp

1 INTRODUCTION

Understanding wind turbine wakes is a key aspect of wind farm design. The growth of computation capacity has allowed turbine wake research to be conducted on computers using methods based on computational fluid dynamics (CFD). However, such methods involve solving the fluid equations repeatedly on numerous grid points, which makes the performance of computation crucial to both the speed and quality of simulation.

In this study, we explore the potential of GPU as a highly parallel computational device in CFD-based wind turbine wake simulation, and analyze the perfomance of our program.

2 METHODOLOGY

2.1 Solver Algorithm

The highly turbulent wind field is simulated using the large eddy simulation (LES) method, which adds an additional viscosity term to the incompressible Navier-Stokes equation to represent vortices smaller than the grid size. The LES Navier-Stokes equation is then solved using the fractional step method, which involves three steps of velocity-pressure coupled time integral [1].

$$\boldsymbol{u}^* = \boldsymbol{u}^n + \Delta t \left(-\boldsymbol{u}^n \cdot \boldsymbol{\nabla} \boldsymbol{u}^n + \boldsymbol{\nabla} \cdot \left(\frac{1}{Re} + \nu_t \right) \boldsymbol{\nabla} \boldsymbol{u}^n \right) \tag{1}$$

$$\nabla^2 p^{n+1} = \boldsymbol{\nabla} \cdot \boldsymbol{u}^* / \Delta t \tag{2}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \Delta t \boldsymbol{\nabla} p^{n+1} \tag{3}$$

Equation (2) is solved by PBiCGStab method using a Jacobi preconditioner.

The wind turbine is represented using the actuator line method, so that the program can use a structured orthogonal grid instead of a boundary fitted coordinate grid for the turbine [3].

2.2 Implemantation

NVidia HPC SDK is used to develop the multi-GPU program in this study, with each GPU managed by one process on the host side. CUDA C++ is used for the GPU part of the program and MPI for inter-process communication [4].

To reduce the overhead of inter-process communication, we adopt the communication-hiding technique in the program, which overlaps the time of non-blocking MPI communication with the computation time of grid points that do not require data exchange with other processes [2].

For the linear solver, the stencil discretization of equation (2) on a structured grid results in a well-formed banded coefficient matrix, which can be represented efficiently by a simple array in our program.

kernel	time(s)	time(%)
Jacobi preconditioner	22.85	16.68%
Calculation of $\boldsymbol{u}^*$ in equation (1)	20.34	14.85%
Matrix-vector product	11.45	8.36%

Table 1: Kernel time

number of GPUs	4	8	12	16
time (s)	1756	1011	779	659
speedup	1	1.74	2.25	2.66
efficiency	1	0.87	0.75	0.67

Table 2: Strong scaling performance

number of GPUs	1	2	4	8	12	16
time (s)	1765	2027	2232	2302	2329	2329
speedup	1	1.74	3.16	6.14	9.09	12.13
efficiency	1	0.87	0.79	0.77	0.76	0.76

Table 3: Weak scaling performance

3 PERFORMANCE

Performance experiments are carried out on Kyushu University's supercompuer system ITO. Each node of ITO's subsystem B has four Tesla P100 GPUs.

3.1 Kernel Execution Time

Kernel profiling is carried out for 1000 time steps (undimensional time 0∼1) using the nvprof tool, the most time consuming kernels are listed in table 1.

3.2 Scalability

The strong scaling uses a grid of 900×300×300 points in total, which is a normal grid size of our targeted application scenario, while the weak scaling uses 300×300×300 grid points per GPU. Both experiments are run for 10000 time steps (undimensional time 0∼10).

REFERENCES

[1] Takeo Kashijima. 1999. *Numerical Simulation of Turbulent Flows.* Yokendo Ltd.
[2] Jianqi Lai, Hang Yu, Zhengyu Tian, and Hua Li. 2020. Hybrid MPI and CUDA Parallelization for CFD Applications on Multi-GPU HPC Clusters. *Scientific Programming* 2020 (2020). https://doi.org/10.1155/2020/8862123
[3] Jens Nørkær Sørensen and Wen Zhong Shen. 2002. Numerical Modeling of Wind Turbine Wakes. *Journal of Fluids Engineering* 124, 2 (05 2002), 393–399. https://doi.org/10.1115/1.1471361
[4] Xi Zhang, Xiaohu Guo, Yue Weng, Xianwei Zhang, Yutong Lu, and Zhong Zhao. 2023. Hybrid MPI and CUDA paralleled finite volume unstructured CFD simulations on a multi-GPU system. *Future Generation Computer Systems* 139 (2023), 1–16. https://doi.org/10.1016/j.future.2022.09.005