

A CFD Heterogeneous Parallel Solver Based On Collaborating CPU and GPU

Jianqi Lai, Zhengyu Tian, Hua Li, Sha Pan

College of Aerospace Science and Engineering, National University of Defense Technology
Deya Road, Kaifu District, Changsha, Hunan, China
laijianqi_kd@nudt.edu.cn

Extended Abstract

Since Graphic Processing Unit (GPU) has a strong ability in floating-point computation power and memory bandwidth for data parallelism, GPU has been widely used in the area of common computing such as molecular dynamics (MD), computational fluid dynamics (CFD) and so on [1]. GPU has brought unprecedented opportunities and challenges to high performance computing with its fast development. The emergence of compute unified device architecture (CUDA), which reduces the complexity of compiling program, brings the great opportunities to CFD [2]. A case study of NVIDIA GTX 1070, performance parameters are as follows: the compute capability is 6.1, the single-precision float is 6080 Gflop/s, the double-precision is 194 Gflop/s and the bandwidth is 256 GB/s, which could meet the demand of large-scale parallel solution of incompressible or compressible Navier-Stokes (NS) equations.

There are three different patterns for parallel solution of NS equations: parallel solver based on CPU, parallel solver based on GPU and heterogeneous parallel solver based on collaborating CPU and GPU [3-4]. As we can see GPUs are relatively rich in compute capacity but poor in memory capacity and the CPUs do the opposite. We need to make full use of GPU and CPU, so a CFD heterogeneous parallel solver based on collaborating CPU and GPU has been established. This solver is implemented in visual C++ 2010 and governing equations are discretized by cell-centred finite volume method. AUSM+UP scheme and central scheme are applied to the discretization of convective fluxes and viscous fluxes, respectively. In addition, we choose explicit multistep Runge-Kutta method for the time integration. As single precision data has a higher efficiency compared with the double in GPU, the single precision data is implemented. Each heterogeneous parallel computer contains one NVIDIA GTX 1070 with a 6GB device memory and one Intel Xeon 2603 with 8 cores.

The double ellipsoid and aerospace plane models are used as tests to analyse the results and heterogeneous parallel efficiency. The results of numerical calculation compared well with experiment data, so the heterogeneous parallel Solver has high computational precision. We use the CPU operation time as a reference to compute heterogeneous parallel efficiency. The heterogeneous parallel efficiency is defined as the CPU operation time divided by the heterogeneous parallel operation time while implemented in the same conditions. To collect the operation time, we simulate 10 time steps, with each having 100 sub_interations, and then average the execution time [5]. The solver achieves a speedup of more than 40, which demonstrates that the heterogeneous parallel solver has high computational efficiency. What's more, for the double ellipsoid and aerospace plane models, the speedup increases as the grid size becomes larger. This is due to the proportion of single instruction task increases. As we all know, GPUs specialize in dealing with single instruction task, while the CPUs are good at dealing with branch instruction task.

Keywords: Graphic Processing Unit; CFD; Heterogeneous Parallel; Speedup

References

- [1] X. Liu, Z. Zhong, K. Xu, "A hybrid solution method for CFD applications on GPU-accelerated hybrid HPC platforms," *J. Future Generation Computer Systems*, vol. 56, no. C, pp. 759-765, 2016.
- [2] CUDA C programming guide. Nvidia Corporation, vol. 120, no. 18, p. 8, 2011.

- [3] D. A. Jacobsen, J. C. Thibault, I. Senocak, "An MPI-CUDA Implementation for Massively Parallel Incompressible Flow Computations on Multi-GPU Clusters," *48th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, 2010.
- [4] E. Elsen, P. Legresley, E. Darve, "Large calculation of the flow over a hypersonic vehicle using a GPU," *Journal of Computational Physics*, vol. 227, no. 24, pp. 10148-10161, 2008.
- [5] C. Xu, J. Fang, Y. Che et al, "Collaborating CPU and GPU for large-scale high-order CFD simulations with complex grids on the TianHe-1A supercomputer," *Journal of Computational Physics*, vol. 278, no. C, pp. 275-297, 2014.