Implementation and Performance Evaluation of Parallel OpenACC Climate Code City-LES on GPU Cluster

Daisuke Tsuji[1], Hiroto Tadano[3], Taisuke Boku[3], Ryosaku Ikeda[3,*], Takuto Sato[2], Hiroyuki Kusaka[3]

1) Graduate School of Systems and Information Engineering, University of Tsukuba, Ibaraki, Japan.
2) Graduate School of Life and Environmental Sciences, University of Tsukuba, Ibaraki, Japan.
3) Center for Computational Sciences, University of Tsukuba, Ibaraki, Japan.

ABSTRACT

Climate simulation based in computation fluid dynamics equation is well known as one of the effective HPC application especially for GPU clusters. Center for Computational Sciences (CCS for short), University of Tsukuba, has been developing a local area climate simulation code named City-LES based on Large Eddy Simulation toward city-level detailed simulation not only for atmospheric fluid dynamics simulation but also including surface conditions such as the material of surface (soil, aspect, etc.) or structure (building, tree, forest, etc.). Since the code is complicated and using LES as basic method, the computation amount is quite huge even for city-size problems.

We have developed a GPU-ready version of City-LES code to accelerate the computation speed on large scale GPU clusters. Since April 2019, CCS introduced a new GPU+FPGA hybrid accelerated cluster named Cygnus[1]. Each computation node is equipped with four NVIDIA Tesla V100 GPU as well as two sockets of Intel Xeon CPU. The total count of computation nodes is 80, and 32 special nodes out of them are also equipped with two of Intel Stratix10 FPGA cards. In this poster we focus on GPU acceleration without FPGA.

Original City-LES code [2] targets a local area climate of city size implemented by OpenMP+MPI multi-threaded distributed memory system such as ordinary clusters. In this research, we apply OpenACC and CUDA Fortran to the City-LES code, mainly on the dominant computation part of Runge-Kutta method calculation, and evaluate the performance on Cygnus cluster system with up to 32 nodes (128 V100 GPUs) for strong scaling and weak scaling cases. There are 4 NVIDIA Tesla V100 GPUs and 4 HDR100 InfiniBand where 2 Intel Xeon Gold 6126 CPUs (12 cores) and 2 FPGAs on Cygnus node (we do not use FPGAs in this research). To avoid the complexity of GPU coding, we run multiple MPI processes even on a single node where each process controls one GPU, therefore we run four MPI processes at maximum on each node.

At the first step of GPU coding, we apply CUDA Fortran. When a partial code of City-LES is ported to CUDA Fortran, it requires certain amount of data movement between CPUs and GPUs because not entire computation because some parts are still executed on CPUs. We will implement finally the entire computation part of City-LES completely on GPUs, however a small computation part is not cost-effective to implement on GPU. The only reason why do it is just to remove the redundant data movement between CPUs and GPUs. To make things smoothly, we decided to apply OpenACC for these parts to minimize the coding complexity.

First, we compare performance between 2 CPUs and 2 GPUs in a node. The execution time on 2 CPUs is 441 seconds where it is 218 seconds on 2 GPUs. GPU accelerates the performance up to 2.03x, however the data movement taking 93 seconds is a huge overhead. We estimated that the maximum performance gain with 2 GPUs is 3.14x if we eliminate the data movement. Figure 1 shows current version with partial CPU execution, on weak scaling up to 32 nodes. Figure 2 shows the strong scaling from 4 nodes to 32 nodes. Line graphs show total execution time of CPU or GPU. For these cases, we run 4 MPI processes per node and each process run 6 OpenMP threads. Here, it is shown that the execution time increase is limited up to 1.12x on 32 nodes from single node in weak scaling where the efficiency is kept to 89%. On the strong scaling case, efficiency with 32 nodes is 63% of 4 nodes case, which is not critical. In near future, we will implement City-LES perfectly on GPU to completely avoid data movement between CPU and GPU on each node.

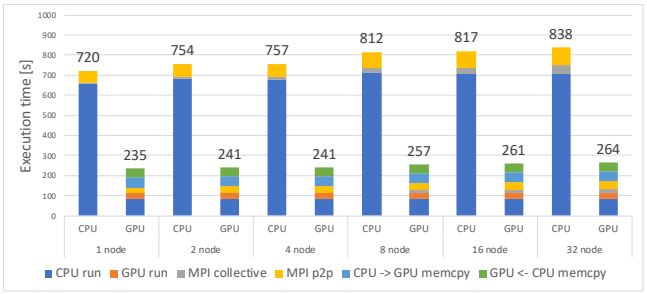

Figure 1: Weak-scaling on Cygnus.

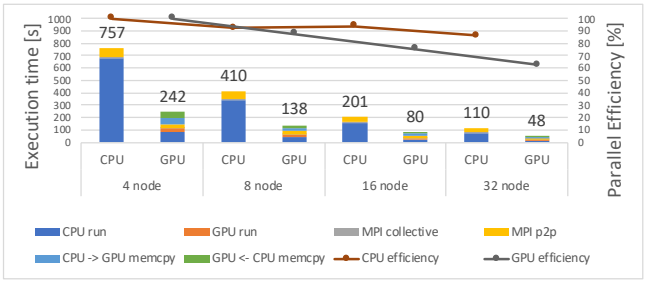

Figure 2: Strong-scaling on Cygnus.

ACKNOWLEDGMENTS

This research is partially supported by " Communication-Computation Unified Supercomputing" project under MEXT's "Next Generation Supercomputer R&D" program, Multi-Disciplinary Collaboration Research Program of CCS, University of Tsukuba, and Collaborative Research between CCS and Weathernews Inc.

REFERENCES

[1] Supercomputers - Center for Computational Sciences. https://www.ccs.tsukuba.ac.jp/eng/supercomputers/#Cygnus
[2] Ikeda, R., H. Kusaka, S. Iizuka, and T. Boku.: Development of Urban Meteorological LES model for thermal environment at city scale. 9th International Conference for Urban Climate, Toulouse, France, (July, 2015).

*Present affiliation is Weathernews Inc.