Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures
Final Report for JHPCN Joint Research of FY 2022

jh220031

Targeting exa-scale systems: performance portability and scalable data analyses

Yuuichi Asahi（Japan Atomic Energy Agency）

We aim at establishing performance portable implementations for high performance fluid simulations and developing large scale data analyses for extreme-scale simulations. For high performance computing, we have demonstrated that a performance portable implementation in C++ alone is possible without harming the readability and productivity. As a data-driven studies, we have developed two deep learning models. Firstly, we have developed a surrogate model to predict the plume dispersion in a complicated urban area for the emergence response capability to contaminant gas leakage events. Second, we have developed a deep learning based Sub-Grid-Scale model which allows the large eddy simulation with 1/10 of grid points compared to direct numerical simulations.

1. Basic Information

(1) Collaborating JHPCN Centers (Please choose collaborating centers)

Tokyo
Tokyo-Tech

(2) Theme Area (Please choose one)

Large scale computational science area

(3) Research Areas (Choose one area, only for HPCI resource using project)

Very large-scale numerical computation
Very large-scale data processing

(4) Project Members and Their Roles

Project representative Yuuichi Asahi works to develop a machine learning / deep learning (ML/DL) model to extract features from fluid simulation data. Shinya Maeyama performs the local plasma turbulence simulations. Julien Bigot works on the in-situ data analysis of GYSELA. Xavier Garbet works for theoretical analysis of non-local transport processes. Virginie Grandgirard gives the advice for the large scale plasma turbulence simulation. Kevin Obrejan gives the advice for the optimization of mini-apps. Thomas Padioleau gives the advice for state-of-the-art GPU implementations of mini-apps. Takashi Shimokawabe gives advice for large scale simulation and deep learning models. Keisuke Fujii contributes on data-driven analysis. Naoyuki Onodera gives the advice for the large scale LBM simulation. Yuta Hasegawa gives the optimization on GPUs. Prof. Watanabe comments on characteristics of local transport processes. Yasuhiro Idomura gives advice for the large scale simulations. Prof. Aoki gives advices about the usage of TSUBAME3.0.

2. Purpose and Significance of the Research

In this project, we plan to proceed toward Exascale numerical simulations in its implementation and corresponding data analyses. For high performance computing, we will develop an abstract layer to handle the non-uniform mesh while keeping the performance portability. For large scale data analyses, we will

establish a scalable data analyses method in an in-situ way.

High performance computing

It is essential to explore code refactoring approaches in order to get a good performance on multi-vendor CPUs and GPUs, while keeping the readability and productivity. Establishing a performance portable approach for multiple apps will contribute to computational fluid dynamics (CFD) community as demonstrators. We are planning to publish our codes on GitHub which serve as working examples.

Large scale data analyses

We will develop a machine learning (ML) or deep learning (DL) model to surrogate numerical simulation results completely or partially. Though still is a main stream, modeling everything by numerical simulations alone is getting more and more complicated on the way to Exascale supercomputing due to the ever-increased simulation and data analyses costs. Accordingly, we will investigate the potential of deep learning dedicated to CFD simulations on top of Exascale simulation and data analyses studies.

3. Significance as JHPCN Joint Research Project

Improving a performance portability is a challenging task which needs a good understanding of a large variety of architectures, algorithms and programming models. Japanese group contributes on GPU optimization strategies with Kokkos and directives. French group is developing a zero-cost mesh abstraction library 'DDC' [https://github.com/Maison-de-la-Simulation/ddc] on top of Kokkos. We expect good interactions between French and Japanese groups over performance portability, optimizations on AMD GPUs, and data structures.

For data analyses, international collaborations are necessary for both technical and physical reasons. French group has already succeeded to manage the on-memory simulation data from Dask scripts with PDI library (developed by Dr. Bigot). Japanese group offers the base scripts for machine learning or deep learning in the post hoc manner. By coupling these technologies, we can achieve in-situ ML for numerical simulations. For physical understanding of the obtained results, our team includes the well-known experts in turbulence studies.

The multi platforms offered by JHPCN framework are essential to establish the performance portable implementations over different devices including CPUs and NVIDIA/AMD GPUs. In addition, the large storage offered by JHPCN is essential for deep learning or machine learning study.

4. Outline of Research Achievements up to FY2021 (Only for continuous projects)

This is a new project.

5. Details of FY2022 Research Achievements

High performance computing

We have developed a performance portable version of a 4D (2D+2V) kinetic plasma simulation code with C++ parallel algorithm (stdpar) to run across multiple CPUs and GPUs. Using the language standard parallelism stdpar and the proposed language standard multi-dimensional array support mdspan, we have demonstrated that a performance portable implementation is possible without harming the readability and productivity. The most important benefit of this approach is the lifetime

of the application which can technically be elongated to the lifetime of the language which should be longer than that of libraries, directives, and architectures. We have compared the performance of mini-application with Thrust, Kokkos, OpenMP and stdpar over Intel Icelake, NVIDIA V100 and A100 GPUs and AMD MI100 GPU. It should be noted that stdpar is unavailable on AMD MI100 for the moment.

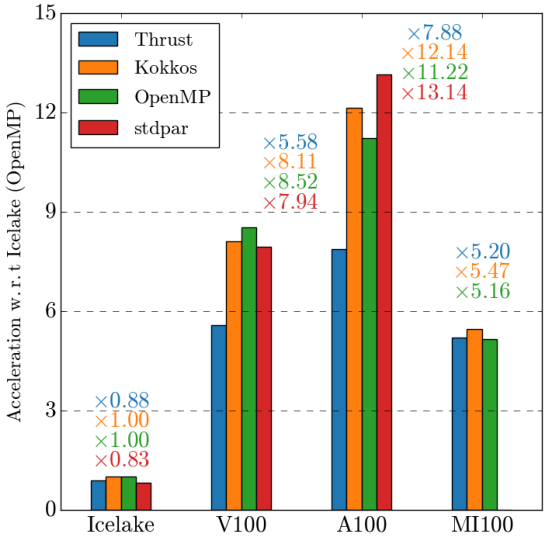

Fig. 1 Performance comparison for MPI version of mini-app with Thrust, Kokkos, OpenMP and stdpar. The problem size is fixed as (Nx, Ny, Nvx, Nvy) = (128, 128, 128, 128) and the number of iterations is 40. We used 2 MPI processes corresponding to 2 Icelake sockets, V100, A100 and MI100 GPUs.

Figure 1 shows the performance of the entire mini-app in terms of acceleration with respect to the baseline OpenMP version on Icelake. The stdpar version exhibits performance in the range of ±20% to the Kokkos version on Icelake, V100, and A100 (Kokkos is known as one of the most promising performance portable layers). The stdpar version demonstrated the best overall performance on A100. It is concluded that stdpar has competitive performance portability. Thanks to the mdspan, we can access to multi-dimensional data which improves the readability with small performance overheads. The most importantly, we can achieve a readable and performant code just with the standard language. Thus, stdpar can be a reasonable choice as an Exascale performance portable implementation assuming it becomes available on AMD and/or Intel GPUs in the future. This work has been presented in the supercomputing conference [3]. The source codes are available for public use [4].

Our production level application CityLBM has also been ported to AMD MI100 GPUs with MPI + HIP. After optimizations, the performance on MI100 is between those on V100 and A100, as expected from the peak memory bandwidth.

The optimization techniques developed in JHPCN projects have been applied to the plasma turbulence code GKV to achieve a remarkable performance on Fugaku. Thanks to large scale simulations on Fugaku, a novel suppression mechanism of turbulent electron heat transport has been demonstrated [2].

Large scale data analyses

We have developed multiple AI models. We also succeeded to integrate Python script for machine learning and a C++ simulation code using the PDI library [https://pdi.julien-bigot.fr].

Firstly, we have developed a DL model to predict the plume dispersion in a complicated urban area for the emergence response

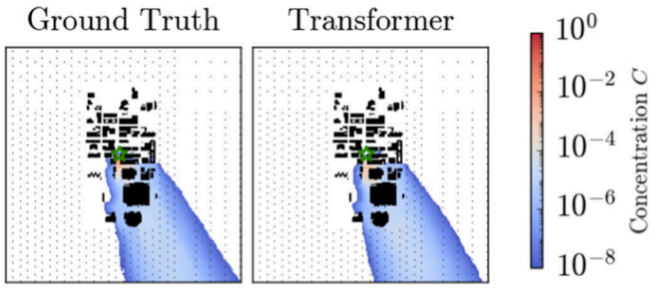

Fig. 2 Prediction of plume dispersion with CNN/Transformer model.

capability to contaminant gas leakage events [1, 5]. Our model can predict the ground level plume concentration from realistically available data such as the time series monitoring data at a few observation stations, and the building shapes and the source location (Fig. 2).

It is also shown that the same model can be applied to predict the source location and amplitude from the time series monitoring data.

Secondly, we have developed a deep learning based sub-grid-scale (SGS) model to extract the effects of small-scale fluctuations from sequential turbulence simulation data. Based on the projection operator method, we have constructed the linear (so-called Mori-Zwanzig (MZ) projection operator) and nonlinear (using neural networks) projection operators from the data.

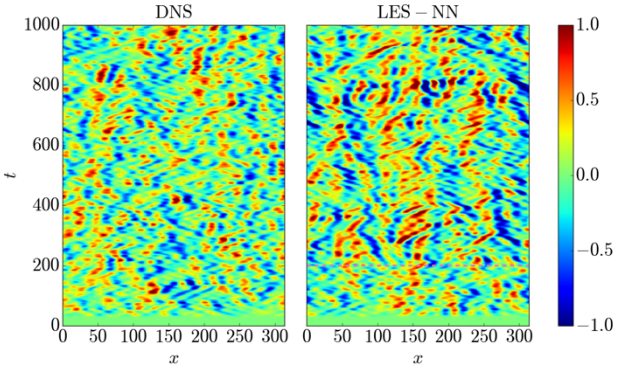

Fig. 3 Spatio-temporal evolution of fluctuations with DNS and LES. We apply the low-path filter to the DNS data for comparison. In LES-NN model, we first run DNS up to t=200 and LES-NN model is enabled.

Figure 3 shows the spatio-temporal evolution of 1D Kuramoto-Sivashinsky, with direct numerical simulations (DNS) and LES-NN models. LES-NN model employs the SGS operator trained on a large amount of DNS data. We employed the Transformer architecture to model the SGS operator from DNS sequential data. In the LES model, we only use 1/10 of grid points in DNS wherein the scale of instability source and dissipation scale are not even resolved (see Fig. 4). We also confirmed good agreements of time average of energy spectral with DNS, LES (MZ) and LES (NN) models as shown in Fig. 4.

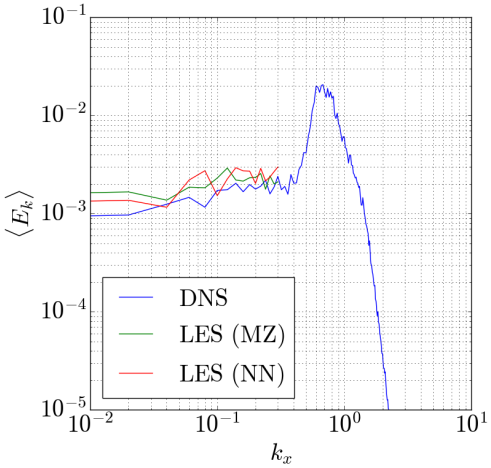

Fig. 4 Time averaged energy spectral with DNS, LES-MZ and LES-NN models.

6. Self-review of Current Progress and Future Prospects

High performance computing

In 2022, we have newly explored the performance portable implementation with C++ parallel algorithm (stdpar). We have also ported and optimized our production code on AMD GPUs. Although we have initially planned to rebase our mini-apps with 'DDC', we have not completed this task yet (partially done by French group). This remains as a future task for 2023.

In 2023, we further explore the performance portable and asynchronous implementation in C++ "senders/receivers" that is proposed for C++26. We are planning to implement our mini-apps in "senders/receivers" and compare the performance with the SYCL version which also allows asynchronous executions. We aim to establish a baseline implementation for asynchronous execution of fluid simulation codes while keeping the performance portability.

Large scale data analyses

In 2022, we have successfully developed DL

models that surrogate simulation results or assist the simulation itself. We have integrated the incremental PCA script into the Voice 1D+1V code in C++ using the PDI library. The AI-assisted fluid simulation models have just been applied for simple 1D cases as a proof-of-concepts. In 2023, we will aim at demonstrating that these approaches are applicable to 2D fluid simulations. We will keep our implementation in a scalable manner to illustrate a pathway to the production codes. We will implement the simulation code in C++ and the AI code in Python, which are coupled using PDI library for flexibility.

7. List of Publications and Presentations

Please see an example in a comment above

(1) Journal Papers (Refereed)

[1] Y. Asahi, N. Onodera, Y. Hasegawa, T. Shimokawabe, H. Shiba, and Y. Idomura, "CityTransformer: A Transformer-Based Model for Contaminant Dispersion Prediction in a Realistic Urban Area", Boundary-Layer Meteorology **186**, 659-692 (2023).

[2] S. Maeyama, T.-H. Watanabe, M. Nakata, M. Nunami, Y. Asahi, and A. Ishizawa, "Multi-scale turbulence simulation suggesting improvement of electron heated plasma confinement", Nature Communications 13, 3166 (2022).

(2) Proceedings of International Conference Papers (Refereed)

[3] Y. Asahi, T. Padioleau (+), G. Latu (+), J. Bigot (+), V. Grandgirard (+) and K. Obrejan (+), "Performance portable Vlasov code with C++ parallel algorithm," *2022 IEEE/ACM International Workshop on Performance, Portability and Productivity in HPC (P3HPC)*, Dallas, TX, USA, 2022, pp. 68-80, doi: 10.1109/P3HPC56579.2022.00012.

(3) Presentations at International conference (Non-refereed)

(4) Presentations at domestic conference (Non-refereed)

(5) Published open software library and so on.

[4] Examples of performance portable implementations of mini-applications in Kokkos, OpenMP, Thrust and stdpar.
https://github.com/yasahi-hpc/P3-miniapps

[5] A Deep learning model for plume concentration prediction in the realistic urban area
https://github.com/yasahi-hpc/CityTransformer

(6) Other (patents, press releases, books and so on)