Smart In-Situ Visualization Framework on the Fugaku Environment

Ken Iwata
Kobe University
Japan
1885031t@stu.kobe-u.ac.jp

Go Tamura
Kobe University /
RIKEN R-CCS
Japan

Naohisa Sakamoto
Kobe University /
RIKEN R-CCS
Japan

Jorji Nonaka
RIKEN R-CCS
Japan

1 INTRODUCTION

Continuous increase in the simulation scale and the cost of I/O have increased the attention to the in-situ processing, where the simulation data is reduced in different manners for minimizing this I/O cost. However, traditional batch-based in-situ visualization can produce large amounts of rendering results for post-hoc visual analysis, which can make difficult to gain insight into the simulation results. In this poster, we present an in-situ visualization framework, named as “Smart In-Situ Visualization”, which is focused not only on simple data reduction, but especially on the elimination of unnecessary timesteps and viewpoints for understanding the underlying simulation phenomena without missing important data features. For this purpose, we have worked on an in-situ adaptive timestep selection [4], and have also started working on an adaptive viewpoint selection to be used with the in-situ multiple viewpoint rendering [3]. We expect that these features can be useful for reducing the I/O cost, and the time required to obtain scientific knowledge from the numerical simulation results.

2 IN-SITU VISUALIZATION FRAMEWORK

Our proposed in-situ visualization framework is based on the KVS (Kyoto Visualization System) [2], a multi-platform OpenGL-based general purpose visualization library. The Mesa3D graphics library, necessary for running OpenGL based applications, was officially provided on the K computer only almost at the end of its operational period [1]. On the other hand, Fugaku can take advantage of a wider software ecosystem due to the use of the Armv8.2-A compliant Fujitsu A64FX CPU. Currently, the Mesa3D graphics library is installed, via Spack, on both Fugaku’s Compute Nodes (A64FX) and Pre/Post Processing Nodes (x86/GPU). In addition, it is also possible to use Python, which was not possible on the K computer, and we have used Python/C API for implementing part of the in-situ adaptive timestep sampling functionality. This timestep sampling is based on the amount of change between the simulation timesteps, which is estimated via Kernel Density Estimation (KDE) and Kullback-Leibler (KL) divergence.

For the porting process to the Fugaku environment, we used Fujitsu compiler (Ver. 4.3.1) for the Compute Nodes and GCC (Ver. 8.3.1) for the Pre/Post Processing Nodes; Mesa3D graphics library (Ver. 18.3.6); and Python (Ver. 3.6.8), those made available via Spack (Ver. 0.16). In addition, we used the same OpenFOAM CFD simulation code (Ver. 2.3.1) and simulation models (Unstructured Mesh Data) used on the K computer (SPARC64fx) and K Pre/Post Cloud System (x86/GPU). For the initial evaluations, we have used isosurface, orthoslice, and PBVR (Particle Based Volume Rendering) as the rendering methods. On the Fugaku’s Compute Nodes, we could successfully execute a stability evaluation by running consecutively the simulation and visualization for more than 10,000 time steps. In addition, we were also able to conduct a scalability evaluation by using up to 4,096 nodes ($16 \times 16 \times 16$ subdivision).

As some future works, we plan to continue our development work by using more recent simulation code (OpenFOAM 8) as well as other codes with different volume data formats (such as Structured Mesh Data). We are also planning to investigate the use of ADIOS2 I/O framework, available on the Fugaku, as the in-situ and in-transit adaptor.

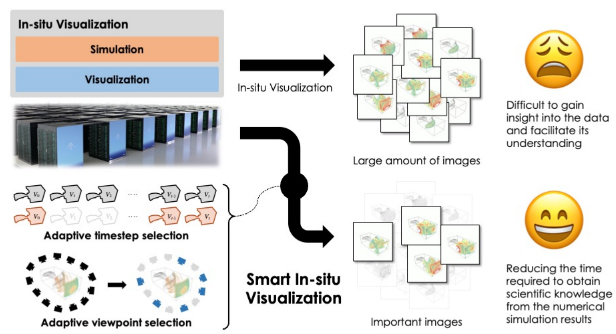

Figure 1: Overview of the Smart In-Situ Visualization.

ACKNOWLEDGMENTS

This work used computational resources of the supercomputer Fugaku provided by RIKEN. This work was partially supported by JSPS KAKENHI (Grant Numbers 19H03976, 20H04194). We are grateful to Prof. Tsukasa Yoshinaga and Prof. Kazunori Nozaki for the OpenFOAM CFD simulation models. We are grateful to Hitoshi Murai for his support with the OSS and Spack on the Fugaku.

REFERENCES

[1] Kengo Hayashi, Naohisa Sakamoto, Jorji Nonaka, Motohiko Matsuda, and Fumiyoshi Shoji. 2018. An In-Situ Visualization Approach for the K Computer Using Mesa 3D and KVS. In *High Performance Computing*, Rio Yokota, Michèle Weiland, John Shalf, and Sadaf Alam (Eds.). Springer International Publishing, Cham, 310–322.

[2] Naohisa Sakamoto and Koji Koyamada. 2015. KVS: A Simple and Effective Framework for Scientific Visualization. *JSST Journal of Advanced Simulation in Science and Engineering* 2, 1 (2015), 76–95.

[3] Taiki Sako, Naohisa Sakamoto, and Jorji Nonaka. 2021. Development of a Multiple Viewpoint Rendering Framework for Exploratory In-Situ Visualization (in Japanese). In *VSJ Symposium on Visualization 2021*.

[4] Yoshiaki Yamaoka, Kengo Hayashi, Naohisa Sakamoto, and Jorji Nonaka. 2019. In Situ Adaptive Timestep Control and Visualization based on the Spatio-Temporal Variations of the Simulation Results. In *Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization (ISAV ’19), held in conjunction with SC’19* (Denver, Colorado, USA). 12–16.