Data staging technique for improving post-processing performance in large-scale CFD analysis

Report Number: R20EACA42
Subject Category: JSS Inter-University Research
URL: https://www.jss.jaxa.jp/en/ar/e2020/14230/

Responsible Representative

Keichi Takahashi, Assistant Professor, Nara Institute of Science and Technology

Contact Information

Keichi Takahashi(keichi@is.naist.jp)

Members

Keichi Takahashi

Abstract

Conventional post-processing of CFD simulations was achieved by saving the entire simulation output on a parallel file system and then processing the output. However, this approach is becoming increasingly challenging due to the limitations in storage size and IO bandwidth. Therefore, data staging, where the simulator transfers its output to a post-processing application during runtime, is attracting attention. In this research, we evaluate the feasibility of leveraging data staging technologies on HPC environments exemplified JSS2 and analyze the requirements for data staging middleware and HPC environment.

Reasons and benefits of using JAXA Supercomputer System

We use JSS3 because it comprises two subsystems, which are the main HPC system (TOKI-SORA) and the general purpose system (TOKI-RURI), and it allows communication between the two subsystems.

Achievements of the Year

We tested ADIOS2, a middleware for data staging developed at the Oak Ridge National Laboratory, and evaluated the staging communication performance on JSS3, which became available this fiscal year. We built ADIOS2 v2.7.1 using the Fujitsu compiler 4.4.0 ("clang" mode) on a SORA compute node and succeeded. All unit tests completed without any problems. Note that CMake fails to recognize the Fujitsu compiler in "trad" mode correctly. Thus we could not build ADIOS2 in trad mode.

Next, we used the iotest utility bundled in ADIOS2 to evaluate the staging communication performance on SORA. iotest was used to simulate two applications with the same number of processes, and 100MB of data per process was sent and received between the two applications. We compared the performance of InSituMPI and SSC (Strong Staging Coupler), the two MPI-based staging communication engines that are expected to deliver the best performance. In the case of SSC, we compared its three modes, which are TwoSided, OneSidedPostPush and

OneSidedPostPull. TwoSided is implemented using point-to-point communication functions and OneSidedPostPush/Pull is implemented using one-sided communication functions.

Figure 1 shows the communication throughput between two applications. When running on two nodes, InSituMPI achieved 5.5 GB/s, whereas SSC in all modes achieved 5.9 GB/s. Since the link bandwidth of the Tofu-D interconnect is 6.8 GB/s, InSituMPI and SSC were able to achieve 81% and 87% of the link bandwidth, respectively. As the number of nodes scaled out, the performance of InSituMPI became higher than that of SSC. At 512 nodes, InSituMPI delivered 418 GB/s while SSC delivered 270GB/s regardless of the modes.

These evaluation results demonstrate that ADIOS can utilize the high-performance interconnect of SORA effectively and achieve massive staging communication performance. However, we still need to investigate why SSC does not scale very well compared to InSituMPI. Furthermore, we will test and evaluate staging communication spanning two heterogeneous systems (SORA and RURI).

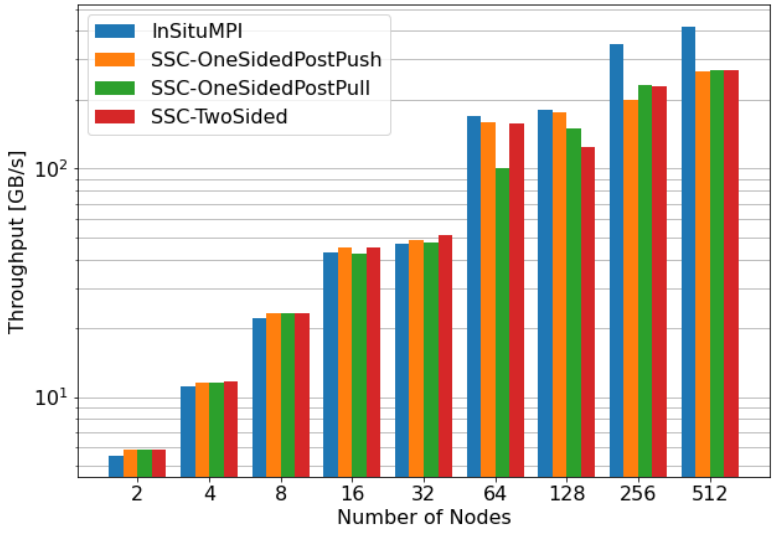

Fig. 1: Throughput between two applications measured using adios_iotest (48 processes per node, maximum of five measuremnets)

Publications

N/A

Usage of JSS

Computational Information

Process Parallelization Methods	MPI
Thread Parallelization Methods	N/A
Number of Processes	1 - 24576
Elapsed Time per Case	10 Minute(s)

- **Resources Used(JSS2)**

Fraction of Usage in Total Resources[*1](%): 0.00

Details

Computational Resources		
System Name	Amount of Core Time (core x hours)	Fraction of Usage[*2](%)
SORA-MA	220.60	0.00
SORA-PP	0.00	0.00
SORA-LM	0.00	0.00
SORA-TPP	0.00	0.00

File System Resources		
File System Name	Storage Assigned (GiB)	Fraction of Usage[*2](%)
/home	9.54	0.01
/data	95.37	0.00
/ltmp	1,953.13	0.17

Archiver Resources		
Archiver Name	Storage Used (TiB)	Fraction of Usage[*2](%)
J-SPACE	0.00	0.00

[*1]: Fraction of Usage in Total Resources: Weighted average of three resource types (Computing, File System, and Archiver).

[*2]: Fraction of Usage：Percentage of usage relative to each resource used in one year.

- **Resources Used(JSS3)**

Fraction of Usage in Total Resources[*1](%): 0.01

Details

Computational Resources		
System Name	Amount of Core Time (core x hours)	Fraction of Usage[*2](%)
TOKI-SORA	26,332.37	0.01
TOKI-RURI	0.00	0.00
TOKI-TRURI	0.00	0.00

File System Resources		
File System Name	Storage Assigned (GiB)	Fraction of Usage[*2](%)
/home	9.54	0.01
/data	95.37	0.00
/ssd	95.37	0.05

Archiver Resources		
Archiver Name	Storage Used (TiB)	Fraction of Usage[*2](%)
J-SPACE	0.00	0.00

[*1]: Fraction of Usage in Total Resources: Weighted average of three resource types (Computing, File System, and Archiver).

[*2]: Fraction of Usage：Percentage of usage relative to each resource used in one year.