Resaerch of model based development for space systems at early development phase

Report Number: R23EDG20177
Subject Category: Research and Development
URL: https://www.jss.jaxa.jp/en/ar/e2023/23726/

Responsible Representative

Taro Shimizu, Research and Development Directorate, Research Unit III

Contact Information

Hikaru Mizuno , Research Unit III, Research and Development Directorate. Japan Aerospace Exploration Agency(mizuno.hikaru@jaxa.jp)

Members

Kaname Kawatsu, Hirofumi Kurata, Hikaru Mizuno

Abstract

Recently, there are many research with regard to model-based development(MBD) approach and degialization of space systems. JAXA Reserch Unit III is developing system simulator necessary for space system design in early phase, and it can be used to improve efficiency of development time. In this project, we will develop a method to run various MATLAB/Python/C-based simulation effectively in the concept study phase ,especially large simulations such as parameter studies and Monte Carlo simulations, on JSS3.

Reasons and benefits of using JAXA Supercomputer System

All JAXA employee can use JSS quickly and easily without any complicated procedures.

The system can be connected within the JAXA intranet, so there is little risk of information leakage.

Quick access to extensive support on how to use the system.

Linux contaier technology can be applied.

Achievements of the Year

Result 1.

A framework developed in FY2022 to efficiently execute parameter studies of control parameters for the development of the HTV-X automated docking mechanism was introduced to HTV-X project members to improve the efficiency of performing numerous simulations, such as Monte Carlo simulations.

For the on-orbit demonstration of automated docking, it is necessary to ensure that the loads/moments generated during docking are less than the specified limits under various docking conditions (e.g., speed/attitude). To check this, it is necessary to consider and confirm various conditions (e.g., test piece side/environmental conditions) using Monte Carlo simulations. In this study, NASA-Trick, a docking simulator used by NASA, was built to run on the Singularity container, enabling parameter studies of automated docking to be performed prior to the actual test.

Specifically, NASA-Trick was containerized in a Docker container, converted to a Singularity container on the JSS3 side, and a mechanism was constructed to run the Trick simulation in parallel in a batch job. This enabled a maximum of 400 parallel runs, and several hundred thousand cases of Monte Carlo simulations could be completed in 1.5 months using JSS3 TOKI-RURI.

Result 2.

We have developed a method for parallel execution of Simulink models using a Windows virtual machine (Windows-VM) with many CPU cores on JSS3, which was developed in FY2021, and have successfully accelerated the parallel execution of the GNC simulator and the Monte Carlo simulations in the CRD2 project. This contributed to the efficiency of the simulation work in the project. Specifically, to evaluate GNC performance and check control stability due to sensor errors in a simulated orbit simulation environment, Monte Carlo simulations with many sensor errors were executed in parallel on the JSS3 Windows-VM. The parallel execution enabled 32-parallel computation, which is nearly 20 times faster than the previous method that took nearly 30 minutes per case, which was a bottleneck for multiple execution. The parallel execution enabled a 32-parallel execution, which is nearly 20 times faster than the conventional method.

Result 3.

Continuing on from FY 2022, we investigated the capabilities of Whisper, a transcription AI developed by Open AI, and a transcription AI developed by a Japanese company, ReazonResearch. The transcription AI requires a large amount of GPU memory to run in a local environment. In particular, the most accurate Large model requires more than 10 GB of VRAM, making it difficult to run on an ordinary PC. In this study, we developed a method to run OSS transcription AI on JSS3 by attaching TOKI-RURI GPU to Singularity containerized Whisper and ReazonResearch models, utilizing TOKI-RURI-GP, which has abundant VRAM. We developed a method to run OSS transcription AI on JSS3 by attaching TOKI-RURI GPU to Whisper and ReazonResearch models. Combined with Pyannote, a framework that enables speaker separation, we were able to achieve highly accurate transcription, including speaker separation with a single microphone, which was not possible with the conventional transcription AI provided with Microsoft Teams and other applications. We confirmed that speaker separation and recording of the content of each speaker's speech were possible using speech data evaluated at the end of the research year as a trial (accuracy was approximately at the 65-point level). In the future, we will continue to investigate the feasibility of automating the transcription, summarization, and transcription of the proceedings in combination with the result 3 described below.

Result 4.

In addition to the transcription AI introduced in Result 2, we tested local execution of a large-scale language model (LLM), which has been remarkably developed in recent years, on JSS. Specifically, we tried to run Llama2, which was developed by Meta and released as OSS, and ELYZA, an improved version of Llama2 for Japanese, on JSS. We were able to package them as GPU containers running on TOKI-RURI-GP GPUs by utilizing Singularity containers, including the construction of the development environment.

However, the Web API has not yet been successfully implemented on JSS, and we plan to continue technological development and research on this point in FY2012 and beyond.

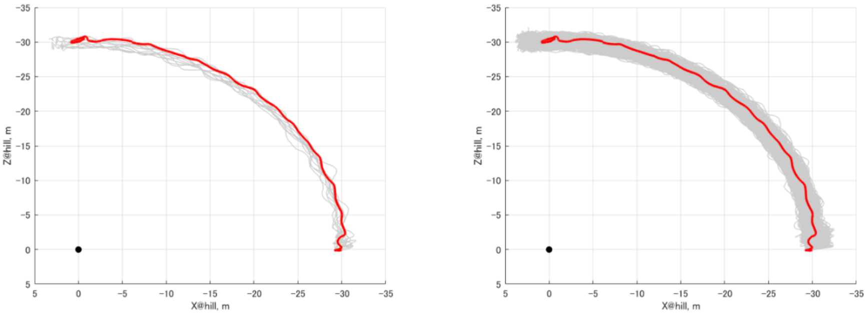

Fig. 1: Comparison of Monte Carlo simulation results from the CRD2 project with experimental data

Publications

N/A

Usage of JSS

- **Computational Information**

Process Parallelization Methods	N/A
Thread Parallelization Methods	N/A
Number of Processes	1
Elapsed Time per Case	30 Minute(s)

- **JSS3 Resources Used**

Fraction of Usage in Total Resources[*1](%): 0.03

Details

Computational Resources		
System Name	CPU Resources Used (core x hours)	Fraction of Usage[*2](%)
TOKI-SORA	0.00	0.00
TOKI-ST	224,428.43	0.24
TOKI-GP	0.00	0.00
TOKI-XM	0.00	0.00
TOKI-LM	0.00	0.00
TOKI-TST	0.00	0.00
TOKI-TGP	215.77	16.61
TOKI-TLM	0.00	0.00

File System Resources		
File System Name	Storage Assigned (GiB)	Fraction of Usage[*2] (%)
/home	30.00	0.02
/data and /data2	1,725.00	0.01
/ssd	0.00	0.00

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.

- **ISV Software Licenses Used**

ISV Software Licenses Resources		
	ISV Software Licenses Used (Hours)	Fraction of Usage[*2] (%)
ISV Software Licenses (Total)	0.00	0.00

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