宇宙システム解析検証(プロジェクト上流における複合モデルベースデザイン技術の構築)

報告書番号：R23JDG20177
利用分野：研究開発
URL：https://www.jss.jaxa.jp/ar/j2023/24060/

責任者

清水太郎，研究開発部門第三研究ユニット

問い合せ先

水野 光, JAXA 研究開発部門 第三研究ユニット(mizuno.hikaru@jaxa.jp)

メンバ

河津 要，倉田 博文，水野 光

事業概要

近年，宇宙機業界でも MBD アプローチやによる開発の効率化，デジタル化による開発手法刷新についての研究が盛んに行われている．第 3 研究ユニットでは開発早期の宇宙機システム設計開発の効率化を目的として，モデルベース開発技術の取り込み・適用の研究を進め，宇宙機設計に必要なシステムシミュレーション技術の開発を実施している．

本事業では，衛星概念検討フェーズで実施される MATLAB や C/Python ベースの各種シミュレーションの計算，特にパラメータスタディやモンテカルロシミュレーションのような大量の計算を JSS3 上で効率的に実行する手法研究開発を実施する．

参考 URL: https://stage.tksc.jaxa.jp/jedi/sysd/index.html

JAXA スーパーコンピュータを使用する理由と利点

・JAXA 職員であれば煩雑な手続き無しでクイックに利用可能であること
・JAXA 内のシステムであるため，同じ JAXA イントラネット内で接続可能であり情報流出のリスクが少ないこと
・開発中の宇宙機の設計情報のような機微情報を JAXA 内で閉じて取り扱えること
・システム使用方法について手厚いサポートがクイックに受けられること
・Linux コンテナ技術に対応していること

今年度の成果

成果 1.

22 年度に開発した HTV-X 自動ドッキング機構開発向けの制御パラメータのパラメータスタディを効率的に実行する仕組みを HTV-X プロジェクトメンバーに展開し，モンテカルロシミュレーション

等のシミュレーションの多数実行を効率化した.

自動ドッキングの軌道上実証は様々なドッキング条件(速度/姿勢など)においても, ドッキング時の発生荷重/モーメント等は規定値以下であることを保証する必要がある. それを確認するために, モンテカルロシミュレーションにて様々な条件(供試体側/環境条件など)を考慮・確認する必要がある. そこで今回, NASA で使用されているドッキングシミュレータである NASA-Trick を Singularity コンテナにて動作するよう構築し, 実試験前に自動ドッキングのパラメータスタディを実施することができるようになった.

具体的には, NASA-Trick を Docker コンテナ化し, それを JSS3 側にて Singularity コンテナへ変換後, バッチジョブにて Trick シミュレーションを並列実行する仕組みを構築できた. これにより最大 400 並列の計算が実行可能となり, , JSS3 TOKI-RURI にて 1. 5 ヶ月で数十万ケースのモンテカルロシミュレーションを完遂することが出来た.

成果 2.

21 年度に開発した JSS3 上に準備した多数 CPU コアの Windows 仮想マシン(Windows-VM)による Simulink モデルの並列実行の仕組みを活用し, CRD2 プロジェクトにおける GNC シミュレータの並列実行及びモンテカルロシミュレーションの高速化を実現し, プロジェクトにおけるシミュレーション業務の効率化に貢献した. 具体的には軌道上を模擬したシミュレーション環境上におけるセンサ誤差による GNC 性能評価や制御の安定性を確認する為, センサ誤差を多数振ったモンテカルロシミュレーションを JSS3 Windows-VM 上で並列実行した. これにより, 従来は 1 ケース 30 分近くかかっていたため, 多数実行のボトルネックになっていたが, 並列実行化により, 32 並列での計算が可能となり, 20 倍近くの高速化を実現した. これにより 5000 ケースの計算が 2500 時間→約 5 日で実行可能と大幅に効率化を実現できた.

成果 3.

22 年度に引き続き, Open AI が開発した文字起こし AI である Whisper と日本の企業である ReazonResearch が開発した文字起こし AI の実力調査を実施した. 文字起こし AI は, ローカル環境で実行するためには大量の GPU メモリを必要とする. 特に最も精度の高い Large モデルは要求 VRAM が 10GB 以上であり, 普通の PC で動作させることは困難であった. そこで今回, 豊富な VRAM を備えた TOKI-RURI-GP を活用し, Singularity コンテナ化された Whisper 及び ReazonResearch のモデルに TOKI-RURI の GPU をアタッチすることにより, OSS の文字起こし AI を JSS3 上で動作させる手法を開発した. また, 話者分離が可能なフレームワークである Pyannote と組合せ, Microsoft Teams 等に付属する従来の文字起こし AI ではでは実施できなかった 1 つのマイクでの話者分離を含めた高精度な文字起こしを実現することができた. トライとして研究の年度末評価の音声データを利用し, 話者分離及び各話者の話の内容の記録が可能であることを確認した(精度は概ね 65 点レベル). 今後は後述の成果 3 と組合せた文字起こし→要約→議事録化などの自動化がどこまで実現可能であるかの調査を続けていく.

成果 4.

成果 2 で紹介した文字起こし AI に加え, 近年著しい発展をしている大規模言語モデル(LLM)の JSS 上でのローカル実行トライを実施した. 具体的には Meta 社が開発し, OSS で公開している Llama2 及びその日本語向け改良版の ELYZA の JSS 上での実行を試行した. 開発環境の構築も含めて Singularity コンテナを活用し, TOKI-RURI-GP の GPU を使用して動作する GPU コンテナとしてパッケージ化することが出来た.

しかしながら，最終的に目標としていた API によるサービス化については Web API がまだ JSS 上では上手く動作できていない為，この点に関しては 24 年度以降も技術開発及び調査を進めていく予定である．

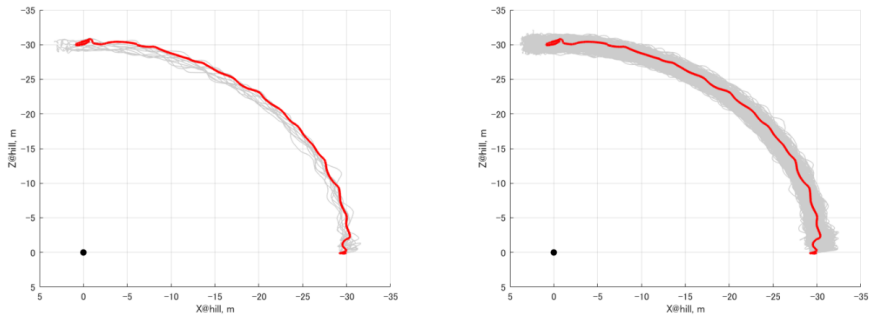

図 1: CRD2 プロジェクトによるモンテカルロシミュレーション実行結果と実験値との比較

成果の公表

なし

JSS 利用状況

計算情報

プロセス並列手法	非該当
スレッド並列手法	非該当
プロセス並列数	1
1 ケースあたりの経過時間	30 分

● JSS3 利用量

総資源に占める利用割合※1 (%)： 0.03

内訳

計算資源		
計算システム名	CPU 利用量(コア・時)	資源の利用割合※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

ファイルシステム資源		
ファイルシステム名	ストレージ割当量(GiB)	資源の利用割合※2 (%)
/home	30.00	0.02
/data 及び/data2	1,725.00	0.01
/ssd	0.00	0.00

アーカイバ資源		
アーカイバシステム名	利用量(TiB)	資源の利用割合※2 (%)
J-SPACE	0.00	0.00

※1 総資源に占める利用割合：3 つの資源(計算,ファイルシステム,アーカイバ)の利用割合の加重平均
※2 資源の利用割合：対象資源一年間の総利用量に対する利用割合

● ISV 利用量

ISV ソフトウェア資源		
	利用量(時)	資源の利用割合※2 (%)
ISV ソフトウェア(合計)	0.00	0.00

※2 資源の利用割合：対象資源一年間の総利用量に対する利用割合