

# 通信障害耐性を備えた実時間分散協調型シミュレーション フレームワーク SilkRoad の広域連携環境での実機実装と検証

大吉 泰嗣<sup>†</sup> 田中 涼<sup>†</sup> 福間 慎治<sup>†</sup> 森 眞一郎<sup>†</sup>

福井大学<sup>†</sup>

## 1. はじめに

実時間シミュレーションはセンサデータやユーザーの操作がリアルタイムに実行中のシミュレーションに反映されるシミュレーションである。一般的に実時間シミュレーションでは計算領域の時空間精度に関するトレードオフが発生するため、領域サイズ、分解能を上げると実時間性が低下する。このトレードオフを解消あるいは低減する手法に実時間協調シミュレーションがある。実時間協調シミュレーションは2つ以上の異なるシミュレーションを連携することで実時間性を保ちながら広域かつ高解像度の結果を得る手法である。我々は、地理的に分散したシミュレーションサーバ間を連携させる実時間分散協調型シミュレーションフレームワーク「SilkRoad」[1]の開発を行ってきた。

しかし、機能追加を目的とした先行研究で行われた動作実験はローカルネットワークや研究室内部のネットワークのみで行われており、遠隔協調環境で動作することが保証されていない。昨年度に福井・大阪・九州の3拠点をを用いた遠隔連携動作実験 [2]が行われたが、全機能の動作検証には至らなかった。そのため、本研究では全機能を動作させることを目標に実装と動作実験を行った。

## 2. SilkRoad の機能

SilkRoad は実時間シミュレーションを協調実行するフレームワークであり、遠隔地を含む複数の計算機を連携してシミュレーションを行う。SilkRoad が連携に必要な通信に関する全ての処理を行うため、利用者は実時間協調のための通信を意識することなく任意のシミュレーションを組み込み連携実行できる。

先行研究を通して SilkRoad には大別して4つの機能が実装されてきた。(1) 上り結果一貫性制御 (Upward Quality Control: 以下 UQC), (2) 下り結果一貫性制御 (Downward Quality Control: 以下 DQC), (3) 入力集約制御 (Input Accumulation & Filtering control: 以下 IAF), 及び (4) 通信障害時の縮退運用及び自動復旧, である。

A wide area real machine implementation and verification of SilkRoad: a real-time distributed cooperative simulation framework capable of autonomous resilience.

Taishi Oyoshi<sup>†</sup> Ryo Tanaka<sup>†</sup> Shinji Fukuma<sup>†</sup> Shin-ichiro Mori<sup>†</sup>  
University of Fukui<sup>†</sup>

## 2.1. 連携通信

図1に示すような、広域低解像度のシミュレーション(上層サーバ)と狭域高解像度のシミュレーション(下層サーバ)の連携を考える。UQC は下層サーバから上層サーバへ高解像度の結果を送る。DQC は上層サーバから下層サーバへ境界条件を送る。これらのサーバ間連携通信により、それぞれのサーバは単独では得ることのできない情報を取得し、自らの計算精度向上を図る。IAF は下層サーバのセンサ入力にフィルタリングを施して上層サーバに送る。これにより入力の通信量や通信遅延を削減できる。

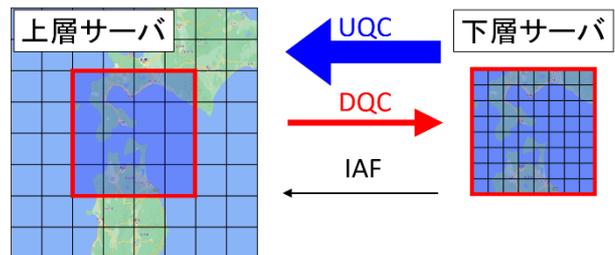


図1 サーバ間連携関連図 (例)

## 2.2. 通信障害耐性

複数のシミュレーションを連携させる実装の一つに、全体を一つの MPI ドメインとして実装する方法がある。しかし、この方法では通信障害が発生した場合にシステム全体のブラックアウトが発生し、個々のサーバの自律性・計算の持続性が担保できない。MPI 通信ライブラリの開発グループ内でも対故障性に関する議論は行われているものの、具体的な実装は進んでいない [3]。そこで、図2のようにシミュレーションと連携通信の MPI ドメインを分割し、ドメイン内の通信をノード間のソケット通信に置き換えることで通信障害耐性を保証した [4]。サーバ間の通信障害時は連携通信ドメインのみが停止することで、シミュレーションは自律的かつ継続的な動作が可能になる。また、フレームワークの機能として通信障害から復旧した場合に自動的に連携通信を再開する機能を提供する。

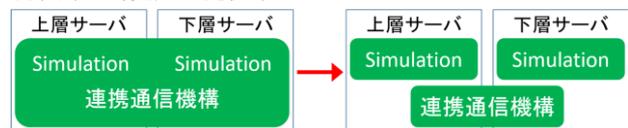


図2 MPI のドメイン分割

### 3. 動作実験

#### 3.1. 動作環境

各拠点間の接続を図3に示す。実施した動作実験におけるプロセス間の接続を図4に示す。内部ネットワークからの接続や異なる環境間の接続を想定して、各拠点間はVPN over SSHを用いて接続する。

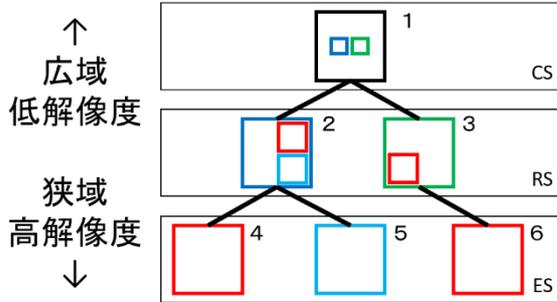


図3 シミュレーション接続図

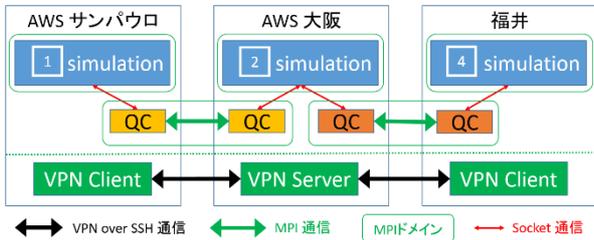


図4 拠点間接続図

サーバは3階層で構成され、上層から順にCentral Server (CS), Regional Server (RS), Edge Server (ES)と名付ける。上層と比較して下層の時間分解能と空間分解能が2倍、すなわち詳細かつ高頻度でシミュレーションを行う。

遠隔協調環境として、CSをAWS EC2サーバ(ブラジル サンパウロ地域)、RSをAWS EC2サーバ(大阪地域)、ESを福井の研究室内に配置した環境を用いた。サーバ間の通信状況をベンチマークプログラム iperf3 [5]と ping により測定した結果を表1に示す。なお iperf3の実行時オプションは -V -t 5, pingの実行時オプションは -c 100である。

表1 各サーバ間の通信速度と往復遅延

サーバ	通信速度 [Mbps/s]	平均遅延 [ms]
サンパウロ-大阪	TX 1.33 / RX 1.02	269
大阪-福井	TX 40.5 / RX 37.0	5.91

#### 3.2. 実験方法

シミュレーション対象としては、動作解析が容易かつ連携効果の確認が容易な2次元熱拡散シミュレーションを用いる。

全サーバを稼働し連携させた状態で数時間運用し、IAFを含めた連携通信が安定して動作し続けることを確認する。SSH接続を十数分程度切断したのち再接続することで通信障害を仮想的に再現する。これを複数回行い、自動復帰機能が安定して動作することを確認する。

#### 3.3. 実験結果

動作実験の結果、この環境において全機能が正常

に動作することを確認した。動作時の各サーバのシミュレーション結果のスナップショットを図5に示す。各サーバの計算領域は図3の番号と対応している。Server3やServer6に注目すると、左下の陸地で囲まれた領域はServer3及びServer6のみの情報では熱が拡散し得ない。この領域への熱の拡散はDQCによってServer1から伝達された境界条件によるものである。

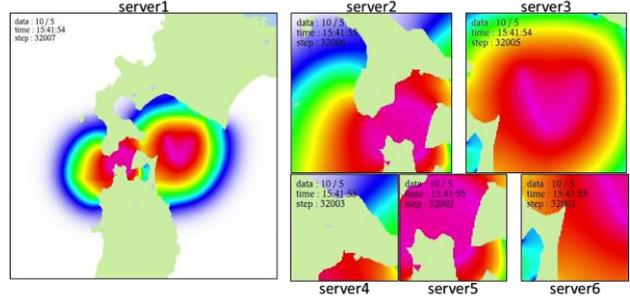


図5 動作時のスナップショット

### 4. まとめ

実時間分散協調型シミュレーションフレームワーク SilkRoad の全機能を用いた動作検証を行った。先行研究 [2]において動作検証を行っていない IAF 及び通信障害復旧時の自動復帰について、正常な動作を確認した。

今後はより大規模なシミュレーションを用いた動作検証を行う予定である。

謝辞：

本研究は一部 JSPS 科研費 JP20K11744 の助成により実施した。

参考文献

- [1] JiaChao Zhang, Shinji Fukuma, Shin-ichiro Mori, "Silk Road: A Framework for Distributed Collaborative Simulation," J. of IPSJ, Vol. 26, No. 3, pp. 237-246, 2018.
- [2] 森 眞一郎, 南里 豪志, "通信障害耐性を備えた分散協調型シミュレーションフレームワーク SilkRoad の 3 拠点連携実験," IPSJ 全国大会, Vol. 86, No. 4A-06, 2023.
- [3] "MPI 4.0," <https://www.mpi-forum.org/mpi-40/>
- [4] 田中涼, 藤田侑弥, 福岡慎司, 森 眞一郎, "通信障害に対して持続可能な分散協調型シミュレーションフレームワークに関する研究," IPSJ 全国大会, Vol. 82, No. 5J-06, 2020.
- [5] "iPerf - The ultimate speed test tool for TCP, UDP and SCTP," <https://iperf.fr/>.