5J-06

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

田中涼† 藤田侑弥† 福間慎司† 森眞一郎†
福井大学†

1 はじめに

我々は, 大規模なシミュレーションを遠隔地の高性能な計算機上で実行し, 複数のクライアントが操作端末からネットワークを通じてステアリング行い, リアルタイムに結果を観測できるシミュレーション環境の構築を目指して研究を行っている. その一つとして分散協調型シミュレーションのためのフレームワーク開発を行ってきた. 本研究の分散協調型シミュレーション環境とは, 遠隔地のシミュレーションサーバ (リモートサーバ) で行うシミュレーションと通信遅延の少ない操作端末の近く (ローカルサーバ) で行うシミュレーションが結果を連携させ行うシミュレーションである. リモートサーバとローカルサーバで物理シミュレーションを実行すると, 時間経過とともに各サーバでの計算結果の違いが大きくなるという問題が発生する. そのため, シミュレーションがある程度進行した時点でシミュレーション結果を他のサーバに送信する. これを一貫性制御と呼ぶ. 本研究の目的は我々の分散協調型シミュレーションのためのフレームワーク [1] に対し, 通信障害耐性を向上をさせた持続可能なシステムを開発することである. 本研究では「通信障害」を広域通信網における通信遅延の著しい増加ならびにメッセージの消失として扱い, 一定時間内に障害から復旧する可能性がある状況を考える.

2 関連研究

並列処理向け通信ライブラリとして広く用いられている MPI に関して 2018 年に実施されたユーザサーベイ [2] ならびに現在実施中のユーザサーベイ [3] では，MPI のフォールトトレランス（耐故障性）に関する課題が指摘されている．一方で、ユーザプログラムも MPI の通信障害に対する対策が十分にできていないことも報告されている．従来のスーパーコンピュータのような安定した運用が可能なシステムにおいては不十分な耐故障性が問題として顕在化することは稀であった．しかしながら，システムの大規模化にともない並列処理環境においても通信障害への耐性が重要になりつつある．

これまでにも MPI のライブラリを独自拡張し耐故障性を持たせた MPI 実装は多くの研究 [4][5] で行われてきたが標準ライブラリとして実装されるには至っていない．また，これらの実装は定期的に状態保存するチェックポイントまで戻って再実行する，あるいは，障害の影響を受けない他のノードへマイグレーションして実行を継続する手法がほとんどで、障害発生の有無にか関わらず定期的にシステム状態のバックアップをとることのオーバヘッドが問題となる．これらの既存研究に対して, 本研究では障害発生時に限り一度だけシステムの状態を保存し, 障害発生中にもシミュレーションの中断はせず, 一時的な機能の縮退を許容した上で可能な限り中断する時間を最小限に抑えたリスタートを行い, シミュレーションが持続可能な環境の設計を行う.

*Development of a distributed cooperative simulation framework sustainable against network failure
†Tanaka Ryo, Fukui University
†Fujita Yuya, Fukui University
†Fukuma Shinji, Fukui University
†Mori Shin-ichiro, Fukui University

3 実装されている通信障害対策

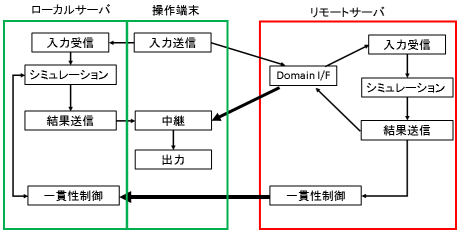

図 1: SCF の構成

3.1 障害分離型の持続可能性保障

通信障害中に遠隔地との MPI 通信を行った場合, 通信待ちや一定時間経過後に MPI によって動かされているプロセスが強制終了するといった問題が起きる. その問題に対し, 従来までのフレームワークでは, 障害検知機能 [Heartbeat] を実装し, 通信障害を検知し障害中には遠隔地との通信を行わないようにし, ローカルサーバのみで縮退運用するといった対策がなされている. この対策を行ったことで通信待ちが解消され, 障害が起きたクライアントはローカルサーバーの結果を得ることができるようになり, その他のクライアントは障害が起きたサーバーを除いてシミュレーションができるようになった. しかしながら障害が一定時間を超えた場合, リモート側のクライアントドメインプロセスが終了してしまい, 復旧が不可能であるという問題を確認した.

3.2 MPD に関する調査

先行研究では, 一定時間以上の障害に対する復旧ができない原因が MPD にあることが判明したため,MPD に関する詳しい調査を行った.

まず,MPI では事前にコミュニケータと呼ばれる通信を行うための集団を作る必要があり,MPD と呼ぶデーモンを実行する. デーモンの起動は 1 つのノード (親) で行い, 親が他ノードの MPD(子) を起動させる. 起動した後には親を含めてすべてのノードでリング状の構造となって通信路が確立される. デーモンは定期的に通信を行いコミュニケータの状態を保持する. コミュニケータ内において通信障害が発生した場合, 一時的にコミュニケータ

内からそのプロセスは外れ通信回復後再びコミュニケータ内に戻る. しかし一定時間通信を試み回復出来なかった場合コミュニケータ内から障害が発生したプロセスは完全に外されてしまい, 外れたプロセスを除いたプロセスでコミュニケータは再構築され, 外れたプロセスはコミュニケータに復帰することができなくなる. 従来の通信障害対策において一定時間を超える通信障害に対応不可であるのはこれに起因している.

4 通信障害時の持続性確保

4.1 要求定義

大規模な計算を行うシミュレーションにおいては実行時間が長時間にも及ぶ場合がある. 障害発生時に最初から実行を行うのは望ましくなく機能を縮退させても動作し続けることが重要である. 障害が復旧する可能性を考慮しているため, 復旧時の対応も必要である. よって, 障害時でも動作し, 復旧時にはフレームワーク本来の機能が復旧可能であることをもってして持続性を持ったシステムであるとする.

4.2 対策検討

障害からの復帰を可能にする最も直接的なアプローチは MPI を機能拡張することであるが, その拡張が MPI の標準機能として公式にサポートされなければ過去の先行研究と同様に通常の一般ユーザはその機能を利用することができない. そこで, 本研究では,MPI レベルでの障害は回避できないものとして, コミュニケータの再構築を許容し, フレームワークレベルでの障害復帰を行う. そのため復旧時にフレームワーク内で何らかのリスタートが必要となる. しかしながら, 従来のチェックポイント・リスタート方式でシミュレーションの再実行を行うと, 障害から復帰したサーバでの再実行が障害とは無関係の他のサーバでのシミュレーションに追いつくまでの間, 時刻 (シミュレーション上のタイムステップ) のずれが継続してしまう. この時刻のずれを最小限に低減するため, 孤立しつつも継続して縮退運用していたローカルサーバーでの最新のシミュレーション結果を用いて再実行する方式を検討する. 前述の障害分離型の SCF で実装した heartbeat ベースの通信障害検知機能は障害からの復旧も検知することが可能である.

5 ドメイン分割による障害対策

MPD の改変は他の MPI を使用するプログラムへの影響を与える可能性があるため望ましくない. そのため, 対策検討したリスタートを用いた方法を提案する.

5.1 対策・実装

提案する対策方法は, ローカルシミュレーションをクライアントドメインから切り離し, 残ったクライアントドメインのみをリスタートさせるという方法である.. 障害の影響を受ける MPI ドメインからシミュレーションを分離することで MPI ドメインの再構築の影響がシミュレーションに波及するのを防ぐ. 想定している構造を図 2 下側に示す. 現在の実装 (図 2 上側) としては,Local シミュレーションもクライアントドメイン (Domain2,MPDRing2) で起動している状況であり, 通信回復後の中断・リスタートはクライアントドメイン全体に渡って行われる. この状況であるとリスタート時にシミュレーションが中断されてしまう. ローカルシミュレーション, クライアントドメインから切り離すことで中断・リスタートするのは,MPDRing3 のみでよくなり, ローカルシミュレーションを中断させずに協調シミュレーションに復帰することができる. ローカルシミュレーションをリモートシミュレーションと同様の構造とすることで、従来のクライアントドメインをローカルサーバードメインとクライアントドメインに分割した.

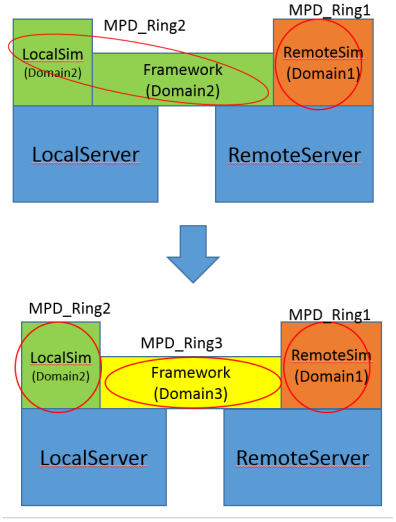

図 2: MPI のドメイン分割

6 まとめ

分散協調型実時間シミュレーション環境において問題となっている通信障害のについて述べ, その対策を検証し実装案を示し, 提案法を実装し, 長期的な通信障害に対しても持続可能な SCF を構築することができた.

参考文献

[1] 藤田侑弥, 福間慎治, 森眞一郎: 通信障害耐性を持った分散協調型実時間シミュレーション環境の構築, 研究報告ハイパフォーマンスコンピューティング (HPC),2019-HPC-168, No2

[2] David E. Bernholdt , Swen Boehm , George Bosilca , Manjunath Gorentla Venkata , Ryan E. Grant , Thomas Naughton , Howard P. Pritchard, Martin Schulz, Geoffroy R. Vallee : ECP Milestone Report A Survey of MPI Usage in the U. S. Exascale Computing Project WBS 2.3.1.11 Open MPI for Exascale (OMPI-X) (formerly WBS 1.3.1.13), Milestone STPM13-1/ST-PR-13-1000

[3] Atsushi Hori, George Bosilca, Emmanuel Jeannot, Takahiro Ogura, Yutaka Ishikawa:Is Japanese HPC another Galapagos?- Interim Report of MPI International Survey -, 研究報告ハイパフォーマンスコンピューティング (HPC),2019-HPC-180,No34

[4] 堀田 勇樹, 田浦 健次朗, 近山 隆:耐故障並列計算を支援する自律的な故障検知機構, 情報処理学会論文誌,Vol.46,pp236-244(2005).

[5] 酒井 将人, 石川 裕:クラスタ監視機能付き MPI 通信ライブラリ, 研究報告ハイパフォーマンスコンピューティング (HPC),2005-HPC-103,pp.175-180