

対話型遠隔シミュレーションシステムの マルチクライアント拡張に関する検討

山本 優¹ 荒川 文貴¹ 西村 祐介² 福間 慎治² 森 眞一郎²

概要：我々は、遠隔地の高性能コンピュータ上でのシミュレーションにおいて対話的な操作を可能とする遠隔シミュレーションシステムの実現を目指して研究を行っている。このような対話的遠隔シミュレーションシステムでは、ネットワーク通信による遅延や位置透過性の問題が発生する。これまでに我々は、遅延隠蔽技術としてシミュレーションキャッシングの開発、Web ベースインタフェースを用いたユビキタスなシミュレーションシステムの開発を行ってきた。本論文では、これまでの研究を元に、シミュレーションキャッシングのマルチクライアント拡張を行い、遠隔地の高性能コンピュータ上のシミュレーションに複数人での協調作業を可能にするシミュレーションシステムの開発を行った。

キーワード：インタラクティブシミュレーション、遠隔シミュレーション、分散協調処理、シミュレーションキャッシング、一貫性制御、シミュレーション共有

1. はじめに

近年の計算機性能の急速な向上に伴い、インタラクティブな実時間シミュレーションへの期待が高まっている。フライトシミュレーション [1] や航空管制シミュレーションのようにコンピュータ上のシミュレーション結果を操作者が直接体感し、その反応として対話的にシミュレーションをステアリング可能なシミュレーションの形態は”human-in-the-loop simulation”あるいは”interactive simulation”と呼ばれる。近年盛んに研究が行われている、災害時の緊急避難シミュレーションなどもその一例である。従来、このようなシミュレータ上で行われてきたシミュレーションは主に離散事象シミュレーションであったが、これをスーパーコンピュータ上の科学技術計算のシミュレーションにも拡張する試みも進められている [2,3,4]。しかしながら、実時間での対話的なステアリングまでを考慮した研究は始まったばかりである [5,6]。これに対して我々は、遠隔地のスーパーコンピュータ上での科学技術計算に対して対話的な操作を可能とする実時間遠隔シミュレーションステアリング、ならびにネットワーク接続可能な場所であればどこからでも大規模シミュレーションを対話的にステアリング可能とするユビキタスなシミュレーション・インタフェースの開発を行っている。

本論文では、遠隔地にある高性能コンピュータ上のシミュレーション結果を複数のユーザで共有し、対話的なシミュレーションステアリングを可能にする遠隔協調シミュレーションシステムの実装法について検討を行い、その予備実装を行った結果を報告する。

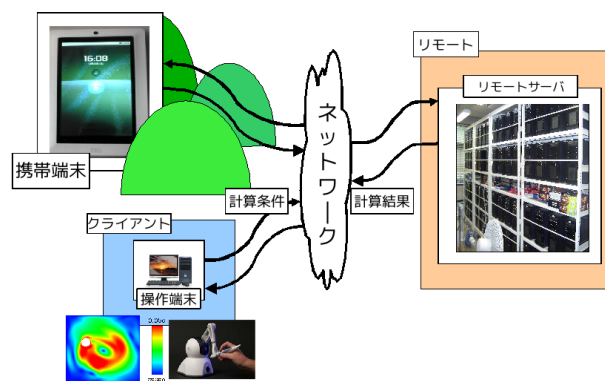


図 1 対話型遠隔シミュレーション

2. 研究背景

2.1 対話型遠隔シミュレーション

手元にある計算機では実行不可能な大規模シミュレーションを遠隔地にある高性能な計算サーバ上で実行し、ネットワークを通じて計算条件や結果を対話的に通信することで遠隔操作するシステムである (図 1)。クライアント側を操作端末としてシミュレーションをステアリングし、サーバ

¹ 福井大学工学部

² 福井大学工学研究科

上でステアリングされたシミュレーションの新しい条件から数値計算を行い、結果データを操作端末にフィードバックすることによって操作端末の負荷を軽減し、手元にある計算機だけでは不可能な大規模シミュレーションを実現するというものである。この時、複数の操作端末からリモートサーバに新しい計算条件を自由なタイミングで転送するため、対象かつ双方向の対話性を各操作端末とリモートサーバは確保することが必要である。

2.2 Web ベースインタフェースを用いた 対話型遠隔シミュレーションシステム

サーバ上で実行中のシミュレーションを、複数のユーザがユビキタスに共有しステアリング可能とすることを目的とした Web ベースのシミュレーションインタフェース [8] である (図 2)。クライアントでは Web ブラウザをインタフェースとして、サーバとの双方向インタラクションを行う。データの入力、表示処理は WebGL を用い、データ通信は Socket と WebSocket を用いて非同期双方向通信を可能としたシステムである。Web ブラウザから入力を与えると、サーバは入力データに応じたシミュレーションを行う。結果データを順次クライアントに配信し、Web ブラウザに表示させることでサーバにて進行中のシミュレーション結果を複数のクライアントで共有することが可能となる。しかし、このシステムでは各クライアントでのシミュレーション結果の提示は、通信路上の遅延変動の影響を受け提示間隔が変動する。

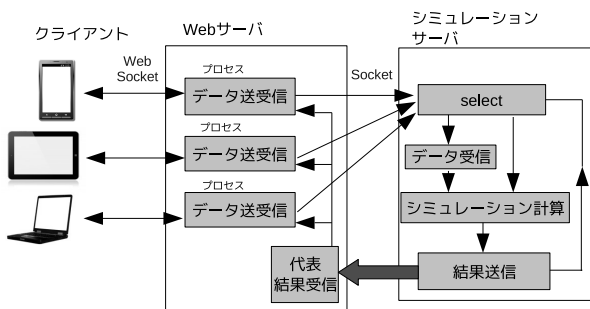


図 2 Web ベースインタフェースを用いた対話型遠隔シミュレーションシステム

2.3 シミュレーションキャッシング

シミュレーションキャッシングで想定する遠隔操作シミュレーションは、遠隔地に存在する高性能シミュレーションサーバ（リモートサーバ）と、ユーザの操作や結果の提示を行う操作端末とその近傍にあるパーソナルシミュレーションサーバ（ローカルサーバ）で構成されるクライアントサーバ型のシステム [7] である (図 3)。通信遅延が存在する環境下においても、一定間隔でのデータ出力支援を持つ。

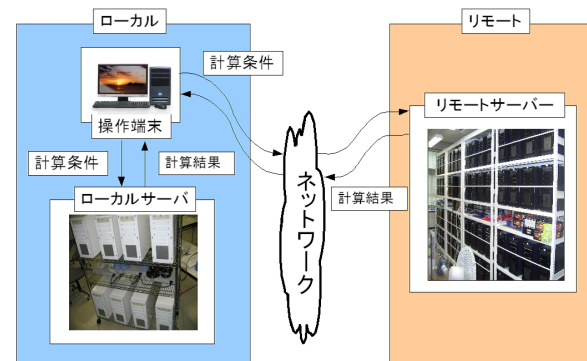


図 3 シミュレーションキャッシング

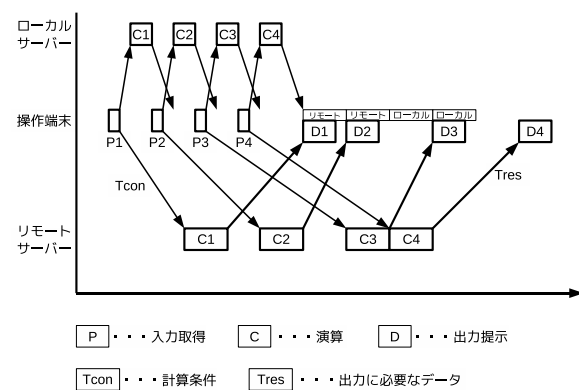


図 4 シミュレーションキャッシングの流れ

2.3.1 遅延隠蔽手法

シミュレーションにパイプライン処理を導入した上で、リモートサーバで実行中のシミュレーションの一部をローカルサーバ上でも並列に冗長実行することで、ユーザからのインタラクションに対してリモートサーバが即応できない場合でもローカルサーバが即応可能である (図 4)。2つのサーバの連携により予測不可能な通信遅延の変動による影響を隠蔽し実時間インタラクションを可能にする手法である。現在の高性能プロセッサに必要な不可欠なキャッシュ・メモリが主記憶上のデータの一部をキャッシュし、メモリアクセス遅延を隠蔽するのに対して、シミュレーションキャッシングではあたかも計算の一部をキャッシングしているかのように振る舞う。

連携の方法としては、例えばシミュレーション結果の提示において出力されるデータ群の中に計算制度や応答時間の許容量に違いがある場合、精度に対して厳しい制約がある場合は多少遅延があったとしてもリモートサーバの結果を使用し、逆に応答時間に厳しい制約がある場合は多少精度が落ちたとしてもローカルサーバの結果を提示するという方法がある。

なお、ローカルサーバで行う「シミュレーションの一部」という概念として様々考えられる。以下に示す。

- ・ リモートサーバで計算するシミュレーション領域全体を粗い解像度でシミュレーションする。シミュレーション領域が2次元の場合、解像度が縦横 $1/N$ とすると、計算量は $1/N^2$ となる。場合によっては時間方向の計算量も $1/N$ となり全体として $1/N^3$ となる場合もある。
- ・ シミュレーションする領域を限定し、リモートサーバで行っている領域の一部をについてシミュレーションを行う。縦横 $1/N$ の領域を選択しシミュレーションする場合、計算量は $1/N^2$ となる。
- ・ シミュレーション領域の一部を高解像度でシミュレーションする。注目領域内においてはリモートサーバよりも高精度のシミュレーションを行う。
- ・ 異なる数値計算モデルを使用する。打ち切り誤差を大きくする等、計算精度とトレードオフの関係にある計算モデルやアルゴリズムを使用する。
- ・ 異なる数値精度を使用する。リモートでは倍精度で計算しているところを、単精度実数を用いる。

一例を図5に示す。一番左がリモートでのシミュレーション領域であるとする、真ん中が全体を $1/2$ の解像度でシミュレーションした場合、右が一部を同解像度でシミュレーションした場合である。これらは適用するシミュレーションの内容および使用するハードウェア・ソフトウェアを加味した上で決定する必要がある。

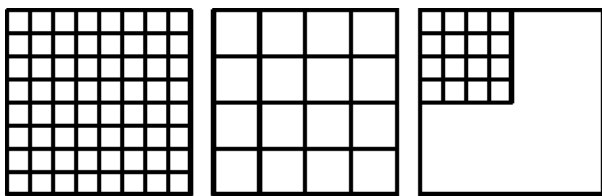


図5 各シミュレーションサーバでの実行例

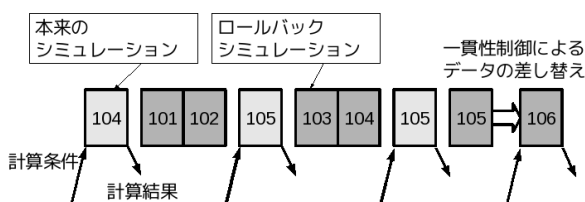


図6 ロールバックモデル

2.3.2 一貫性制御

シミュレーションキャッシングを行う場合、シミュレーションが字間的な依存関係を持つ場合、シミュレーションが進むに連れて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。その為、シミュレーションがある程度進行した時点で各サーバの結果を一致させる処理が必要になる。これを一貫性制御と呼ぶ。

一貫性制御はシミュレーションによって実施の頻度、制御

方式などをカスタマイズする必要がある。シミュレーションキャッシングフレームワークでは中断モデルとロールバックモデルの2つが実装されている。中断モデルは、一貫性制御が終了するまでの間入力を一時中断し、データの一貫性が保証されるまでの間シミュレーションをストップさせ、一貫性制御が終わってからリスタートする。ロールバックモデルはシミュレーションを継続したまま一貫性制御を見かけ上同時並列して実行し、タイムステップが一致した時点でデータを更新する(図6)。

中断モデルでの一貫性制御は、シミュレーションが一時中断してしまうため、応答時間が大きい場合や頻繁に一貫性制御を行う場合不向きであるが、一貫性制御のタイミングを確実に保証する。対してロールバックモデルは、シミュレーションと同時新興して過去の計算条件を利用し、一貫性制御を行うためシミュレーションが中断すること避けられるが、ローカルサーバでの計算量が増え、明確にどのタイミングでデータの一貫性が保証されるか不定である。よって一貫性制御モデルの決定はシミュレーション内容に大きく依存する。

3. シミュレーションキャッシングフレームワーク

我々は、シミュレーションキャッシングをユーザが容易にアプリケーションに適用可能にするフレームワークの実装[9]を行った。これにより、ユーザは本来のシミュレーションを行う部分や入出力の部分の記述にのみ注力でき、シミュレーションキャッシングを使ったアプリケーションの開発効率を上げることができる。必要となるネットワークプログラミングの知識も最低限で済み、簡単にプログラミングを行うことができる。想定するシステム構成は、操作端末・大規模シミュレーションサーバ(リモートサーバ)・近傍の簡易シミュレーションサーバ(ローカルサーバ)の3つをネットワークで結んだものとする。このとき、リモートサーバは適用するシミュレーションに対して十分な性能を持っているものと仮定する。

3.1 搭載機能・仕様

シミュレーションキャッシングを行うため、次のような機能を関数として用意している。

- ・ 操作端末から各サーバへ計算条件の送受信
- ・ 各サーバから操作端末への計算結果の送受信
- ・ パイプライン実現のための時間管理
- ・ 一貫性制御処理
- ・ 操作端末における表示する計算結果の自動選択
- ・ 各プロセス間での同期
- ・ スロースタートアルゴリズム対策

スロースタートアルゴリズムとは、TCP/IPで輻輳制御に用いられるアルゴリズムで、通信が確立した直後のネット

ワーク通信に対して帯域幅が制限されるというものである。一貫性制御の時間間隔が大きくなると、一貫性制御が行われる前にスロースタートアルゴリズムが適用され、通信速度が低下してしまう。特に一貫性制御のデータサイズが小さい場合問題となる。

現在の実装では、対象とする言語はC言語で、各プロセス間の通信にはMPIを使用する。ユーザがフレームワークの関数を用いて作成するプロセスが、入力プロセス、出力プロセス、ローカルシミュレーションプロセス、リモートシミュレーションプロセスであり、フレームワークが提供するプロセスとして中継プロセス、クオリティコントロールプロセスがある。全体の構成図を図7に示す。

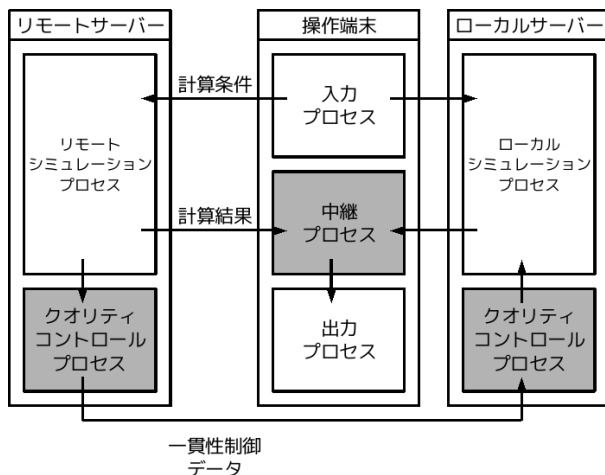


図7 システム構成図

3.2 各プロセスの機能

3.2.1 入力プロセス

入力プロセスでは、各シミュレーションサーバへ送信する計算条件の準備と各シミュレーションサーバへの一定間隔送信が主な処理となる。フレームワークではこの内、計算条件の一定間隔送信機能を関数として提供する。また、中断モデルを用いた一貫性制御時に、一貫性制御が終わるまで計算条件の送信を止める機能も関数として提供する。

3.2.2 出力プロセス

出力プロセスでは、各シミュレーションサーバから結果を受け取り、出力をユーザに提示することが主な処理となる。フレームワークでは、結果の取得機能を関数として提供する。この関数では、どちらのシミュレーションサーバからの結果を受け取ったのかという情報と受け取ったデータサイズを知ることができ、ユーザはこの情報を元にして異なる処理を行うことも可能である（異なる解像度での出力など）。

3.2.3 シミュレーションプロセス

リモート・ローカルの各シミュレーションプロセスでは、計算条件の受信、シミュレーションの実行、結果の送信、一

貫性制御が主な処理となる。この内、フレームワークではシミュレーション以外の処理をそれぞれ関数として提供する。特に一貫性制御に関しては、使用するモデルに応じて複数の関数を提供し、ユーザは使用するモデルによって関数を選択することで一貫性制御を実現する。

3.2.4 中継プロセス

中継プロセスは、各シミュレーションサーバからの計算結果の受信、受信した結果のバッファリング、一定間隔での出力プロセスへの計算結果送信がある。このプロセスが必要となる理由は、各シミュレーションプロセスからの受信処理と出力に必要な処理を簡単に並列して実行可能になるためである。使用するバッファの構造は、送信元のシミュレーションサーバ識別番号(RorL)、シミュレーションステップおよびデータ格納先ポインタを持つ。データ格納領域サイズとバッファのエントリ数は、シミュレーションキャッシング用ヘッダファイルで変更できる。

3.2.5 クオリティコントロールプロセス

一貫性制御の際に一貫性制御データの送信、通常時はスロースタートアルゴリズム回避のため常に非常に小さなデータをやりとりをしている。ただし、一貫性制御データが大きい場合や一貫性制御間隔が比較的小さい場合、スロースタートアルゴリズムの影響は比較的小さいものと思われるため、このプロセスを使用するか、使用せず直接リモートシミュレーションプロセスとローカルシミュレーションプロセスで通信するかをユーザが簡単に変更できる仕様になっている。

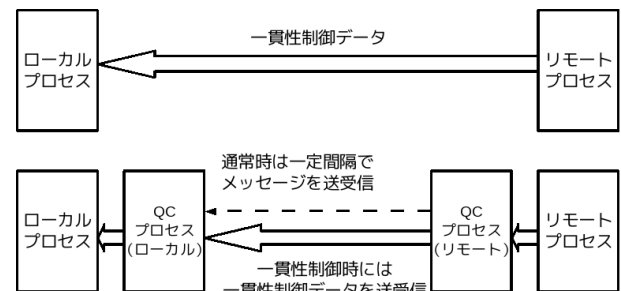


図8 QCプロセスの処理

3.3 一貫性制御

一貫性制御は、リモートサーバのQCプロセスからローカルサーバのQCプロセスに一貫性制御データを送り、ローカルサーバにおいてそのデータが反映されることで実行される。中断モデルの場合、リモートサーバとローカルサーバの一貫性が確保されるまで入力プロセスからの入力をストップさせる。ロールバックモデルでは、ローカルサーバ内においてロールバックに関する計算を行う必要があるが、入力プロセスからの入力はストップされない。フレームワークではロールバックモデルのためのシミュレーション

ステップ管理やロールバックのためのシミュレーション回数の提供を行うが、ロールバックシミュレーションについては提供していない。ユーザが別途記述する必要がある。

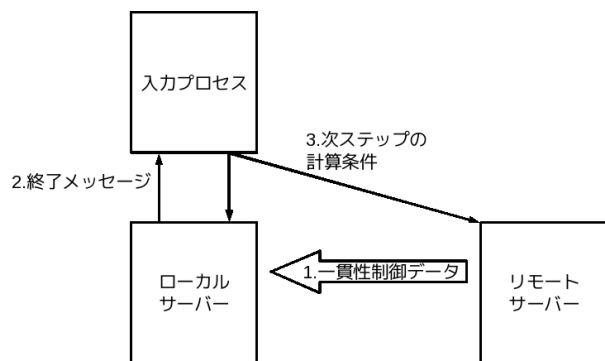


図9 一貫性制御 中断モデル

4. 対話型遠隔シミュレーションシステムのマルチクライアント拡張

4.1 実装方針

我々が開発した Web ベースインタフェースを用いた対話型遠隔シミュレーションシステム・シミュレーションキャッシングフレームワーク、2つの研究を元に大規模シミュレーションにて遅延環境下においても一定間隔でのデータ出力を保証し、複数のクライアントでシミュレーションの共有ができるシステム(図10)の構築を行う。

また、マルチクライアント拡張を行う上でマルチクライアント対応可能なサーバ間連携モデルがいくつか挙げられる。一人のユーザがステアリングを行い複数のクライアントがモニタリングをする SOMM(Single Operator Multi monitor) モデルと複数のユーザが交互にステアリングを行う MOMM(multi monitor multi operator) モデル、複数のユーザが相互干渉が十分に少ない状況で同時にステアリングを行う MOMM モデル、複数のユーザが同時にステアリングを行う MOMM モデルなどである。本来、MOMM モデルでは各クライアントのローカルサーバ間の一貫性制御が必要になる。しかし、今回の実装では、ローカルサーバ間の厳密な一貫性制御は行っていない。そのためシステムの利用形態として以下の制約を与える。

- ローカルサーバではリモートサーバの一部領域を計算するシミュレーションモデルとする
- 複数クライアントが接続されている場合、同時入力はあるが互いに疎干渉である
- クライアントで表示する結果は自身の入力を行う領域のみとする

この条件下では、SOMM モデル、交互 MOMM モデル、疎干渉 MOMM モデルに対応できる。各クライアントで表示する結果には、他クライアントの入力による影響がすぐには反映されないため、2.3.2 章で述べたローカルサーバとリ

モートサーバ間の一貫性制御を行うタイミングでそれまでの入力情報がローカルサーバ上のシミュレーションに反映される。

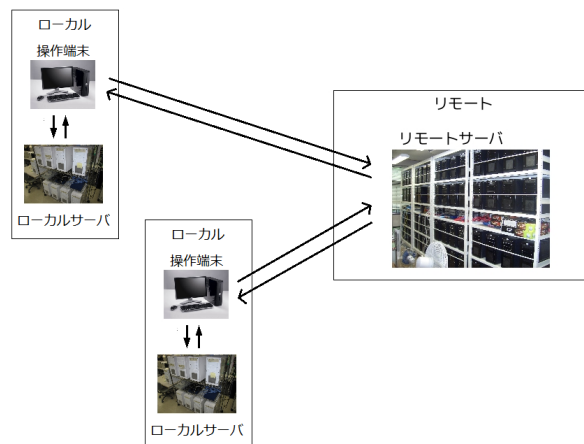


図10 マルチクライアント拡張

4.1.1 機能要件・仕様

マルチクライアント拡張を実現するためには、以下の機能を追加しなければならない。

- 不特定多数のクライアント間でのシミュレーションの共有
- 任意のタイミングでのクライアントの参加および退去
- 特定クライアントの故障、通信回線切断の影響が全システムに波及しない枠組

シミュレーションキャッシングフレームワークでは、全てのプロセスは MPI で通信を行い同期がとられていた。しかし、今回のマルチクライアント拡張によって不特定多数のクライアントが任意のタイミングで接続される。今回の実装では、クライアントやローカルサーバのプロセスは従来同様 MPI を用いて通信を行うが、クライアントとリモートサーバ間の通信のみ Web ベースインタフェース同様、Socket を導入した。これにより、不特定多数のクライアントが任意のタイミングでサーバに接続することが可能となる。また、Socket を導入することでリモートサーバを独立して運用可能になる。特定のクライアントが予期せぬタイミングで接続切れを起こした際も接続していた Socket を close するだけでよいので、全体に影響が波及することはない。各プロセスの関係を図11に示す。

4.2 追加・変更プロセスの説明

4.2.1 シミュレーションプロセス

ローカルサーバのシミュレーションプロセスでは、計算条件の受信、シミュレーションの実行、結果データの送信、一貫性制御が主な処理となる。これは変更はない。

リモートサーバのシミュレーションプロセスは、独立したサーバプロセスである。予め起動しておいたリモートサー

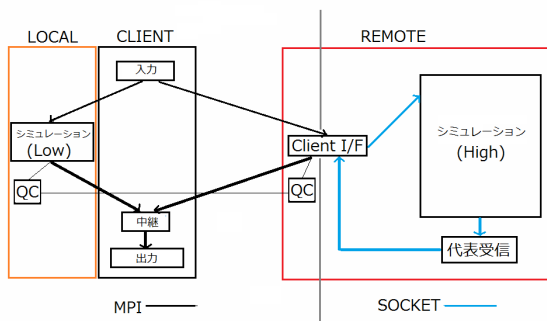


図 11 マルチクライアント拡張時の各プロセスの関係

バのシミュレーションプロセスに各クライアントが接続する方式になっている。複数クライアントの動的な接続を可能とするため Socket が導入されており、Select 関数にて接続 Socket を管理している。同タイムステップ内で複数の入力があった場合、入力データをクライアント毎に管理し、シミュレーションを実行する。主な役割としてはクライアントインタフェースプロセスから計算条件の受信、シミュレーションの実行、結果データを代表受信処理に送信することである。

4.2.2 代表受信プロセス

代表受信プロセスもリモートサーバのシミュレーションプロセスと同様に独立したプロセスとなっている。プログラム実行の際は、リモートサーバのシミュレーションプロセスとセットで予め起動しておく必要がある。主な役割として、シミュレーションプロセスから結果データを受け取り、各クライアントへ配信する処理を行っている。シミュレーションプロセスと分割することで、接続クライアント数が増えた場合でも、シミュレーション本体に通信コストによる影響が出にくくなる。

4.2.3 ClientI/F プロセス

クライアントインタフェースプロセスは、主にクライアントとリモートサーバとの中継、QC プロセスへの一貫性制御データの送信を行う。主な処理としては、クライアントからの計算条件を MPI で受け取り Socket を用いてリモートサーバの Simulation プロセスに送信し、結果データを Socket を用いて受信し、MPI で RECV プロセスへと送信する。

4.3 適用例

4.3.1 熱拡散シミュレーション

このアプリケーションは、熱伝導方程式ベースの差分法により熱の拡散をシミュレーションするものである。入力は2次元領域内の熱源の位置 (XY 座標)、出力はシミュレーション領域の温度を RGB に変換し OpenGL を用いて表示する。ローカルサーバでのシミュレーション解像度はリモートサーバの 1/2 で行っており、計算量は 1/4 である。具体的には、シミュレーション解像度をリモートにおいて 128x128、ローカルでは 64x64 としている。クライアントを

2つ接続した際のクライアントでの表示結果とリモートサーバでのシミュレーション内容を図 12,13 に示す。図 12 は一貫性制御を行う前、図 13 は一貫性制御後である。

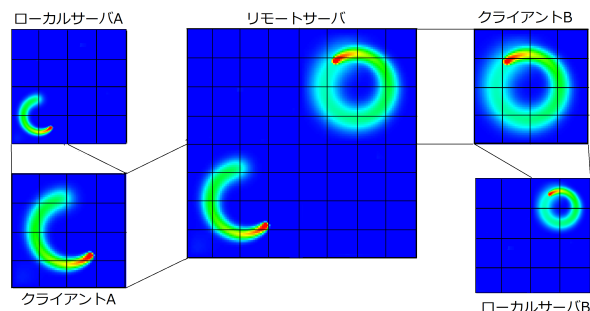


図 12 各サーバでのシミュレーション

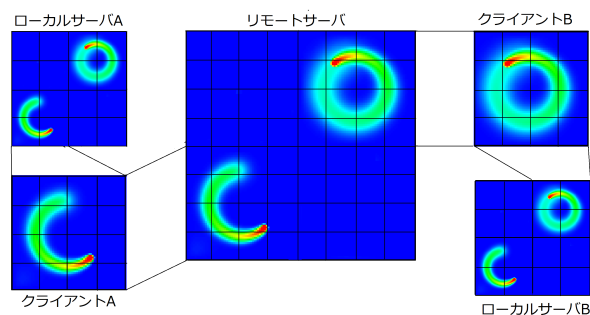


図 13 一貫性制御実施後

5. まとめ

本稿では、我々が開発した遠隔地にある大規模シミュレーションサーバを用いたインタラクティブシミュレーションに対する通信遅延隠蔽手法である、シミュレーションキャッシングのマルチクライアント拡張を行い、各ローカルサーバ間の一貫性制御手法についての検討を行った。その結果として限られたシミュレーションモデル上ではあるが、マルチクライアント環境でシミュレーションキャッシングを実行できるシステムを構築することに成功した。

謝辞 本研究の一部は、JSPS 研究費 25280042 ならびに 25540043 の助成を得て実施しました。また、数々の有力な御指導、御意見を頂いた松山幸雄技術職員に心から感謝致します。上記の皆様方をはじめ、日頃様々な面でお世話になった本学森・福間研究室の皆様様に深く感謝致します。

参考文献

- [1] W.D. McCarty, S. Sheasby, P. Amburn, M.R. Stytz, C. Switzer, "A virtual cockpit for a distributed interactive simulation," IEEE Computer Graphics and Applications, Vol. 14, Issue 1, pp.49-54, 1994.
- [2] R. Marshall, J. Kempf, S. Dyer, "Visualization Methods and Simulation Steering for a 3D Turbulence Model

- of Lake Erie,” Proc. Symp. on Interactive 3D Graphics, pp.89–97, 1990.
- [3] C.Johnson, S.G. Parker, C. Hansen, G.L. Kindlmann, Y. Livnat, “Interactive Simulation and Visualization,” IEEE Computer, Vol. 32, No.12, pp.59-65, 1999. Dec. 1999.
 - [4] Q. Zhu, G. Agrawal, ”Supporting Fault-Tolerance for Time-Critical Events in Distributed Environments”, Proc. of Supercomputing, Paper Nr. 32, pp.1-12, 2009.
 - [5] P. Malakar, V. Natarajan, S.S. Vddhiyar, ”An Adaptive Framework for Simulation and Online Remote Visualization of Critical Climate Applications in Resource-constrained Environment”, Proc. of Supercomputing, Paper Nr. 295, pp.1-10, 2010.
 - [6] P.R. Woodward, D.H. Porter, J. Greensky, A.J. Larson, M. Knox, J. Hanson, N. Ravindran, and T. Fuchs, “Interactive Volume Visualization of Fluid Flow Simulation Data,” PARA’06 Workshop on the State-of-the-Art in Scientific and Parallel Computing, 2006.
 - [7] 橋本 健介, 手塚 俊作, 森真一郎, 富田 真次:”シミュレーションキャッシングと遠隔インタラクティブ流体シミュレーションへの応用, IPSJ Symposium Series(SACSIS ’09), Vol.2009, No5, pp.229-238.
 - [8] 荒川 文貴, 福間 慎治, 辺見 良太, 森 真一郎:”WebGL を用いた対話型遠隔シミュレーションシステムのマルチクライアント対応”, 先進的計算基盤システムシンポジウム 2013
 - [9] 西村 祐介, 森 真一郎:”シミュレーションキャッシングフレームワークの実装”, 福井大学工学研究科情報・メディア工学専攻修士論文