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

荒川文貴† 辺見良太†
福間慎治† 森眞一郎†

1. はじめに

我々は, 遠隔地にあるサーバ上で実行中のシミュレーションに対して, クライアント端末から対話的な操作を可能にする対話型遠隔シミュレーションシステム[1),2)]の開発を行っている. 今までは, 1 台のクライアントと Web サーバを用いて対話型遠隔シミュレーションを行ってきた. これに対し, 本研究は当該システムをマルチクライアントに対応することで, 複数のユーザによる実行中のシミュレーション結果の同時共有ならびに実行中のシミュレーションに対する対話的なインタラクションを可能にするユビキタスなシミュレーション環境を実現する.

2. 従来システムの概要

当該システムは,WebGL[3)] を用いた Web ブラウザをインタフェースとして, 動的 Web 技術を用いてサーバとの双方向通信を行う (図 1)[2)]. データ入力, 表示処理は WebGL を用い, データ通信は WebSocket[4)] を利用することで非同期双方向通信を可能とした. クライアントの Web ブラウザから入力を与えると, サーバは当該入力データに応じたシミュレーションパラメータの変更を行う. そして, シミュレーション結果データはクライアントからの入力の有無に関わらず, 逐次クライアントにフィードバックし Web ブラウザに表示させる.

3. シミュレーションコアの分離

従来システムでは,Web サーバ内にシミュレーションコアを実装していた. そのため, 高レスポンスを要求する通信処理と高スループットを要求する計算処理が同一システムに混在していた. 将来における大規模シミュレーションへの対応ならびにマルチクライアント化に伴う通信処理の増加に対応するため, 本研究の第 1 ステップとしてシミュレーションコアを Web サーバから分離させる実装を行った. シミュレーション行うためのサーバを用意し, シミュレーションサーバと Web サーバを Socket によりデータ通信を行う.

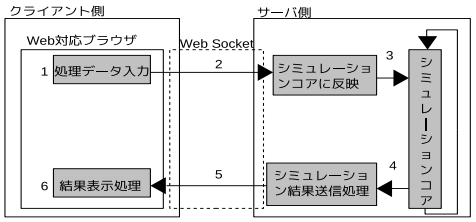

図 1 従来システムの概要図

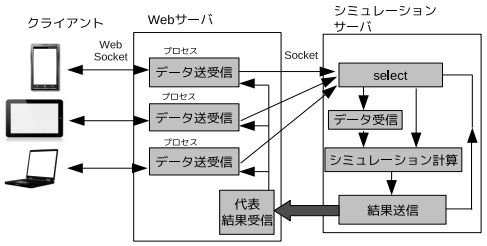

図 2 マルチクライアント対応システムの概要図

時系列シミュレーションに対応可能とするためシミュレーションサーバ側でのシミュレーションはクライアント端末からの入力の有無に関わらず継続して行うことも可能とする. そのためには,Web サーバとシミュレーションサーバ間の Socket 通信を非同期化する必要がある. 具体的には POSIXSocket の select 関数を用いて実装した. シミュレーションコアは select 関数で一旦通信待ちに入るが,select 関数のタイムアウト設定を適切に制御することで, 一定時間クライアントからの入力がない場合はシミュレーションの実行を継続する. これにより、疑似的な非同期化を行っている.

† 福井大学

4. マルチクライアント

WebSocket を用いた双方向通信では Web サーバ側のプロセスは, クライアントの間で Socket を確立しつづける. そのため, クライアント毎に Web サーバ側でサーバプロセスを起動しなければならない. この仕組みは,Apache の pywebsocket モジュールを用いることで実現した (図 2)[5].

シミュレーションサーバはマルチクライアントに対応するために, どのクライアントから入力があっても, パラメータを変更できるようにする.select 関数は, 複数のソケットを監視することも可能であり, ソケットの変化があれば, それに合わせた処理を行うことができる. 現時点では複数クライアントからの同時入力には対応しておらず、仮に同時期にパラメータ変更が発生した場合にはそれらを逐次化して順次変更があったものとして処理を行っている。

シミュレーション結果データの配信に関しては, データを代表として受け取るプロセスを Web サーバ上に作成することで, シミュレーションサーバと Web サーバ間の通信回数を 1 回に削減した. 代表受信プロセスのデータは,Web サーバ内でそれぞれのサーバプロセスに分配した後, 各々のクライアントに配信する. 現在は、シミュレーションサーバと Web サーバ間ならびに Web サーバとクライアント間で授受するデータの解像度は同じであるが、今後は利用かのうな通信帯域を考慮した解像度制御等も検討していく予定である。

5. 実 装 結 果

クライアントに様々な性能の PC または Android 携帯端末を用いて,64×64 の熱源移動を伴う熱拡散シミュレーションを行った. 従来システムのサーバと実装システムの Web サーバには Core2DuoCPU1.86GHz のデスクトップ端末, シミュレーションサーバには Core2DuoCPU2.66GHz のデスクトップ端末を, クライアント側としてデスクトップ端末 1 台, ノート PC1 台, Android 端末 2 台用いている. デスクトップ端末とノート PC は 1Gbps,Android 端末は 58Mbps のネットワークで通信を行う. Web サーバからクライアントへのデータ配信速度の上限設定は 30fps に設定し, クライアント側のデスクトップ端末で結果表示が更新される間隔に与える影響を測定した。 表 1 はデスクトップ端末が接続してから, ノート PC, 続けてノート PC から新たにブラウザを 1 つ, Android 端末 2 台を順次に接続してクライアントが増える毎の結果表示間隔を測定した結果である. その結果, クライアントが順次接続する毎に結果表示更新間隔が増加している. また,Android を接続したときの結果表示更新間隔の増加割合がノート PC 接続の時より大きくなった. クライアント数による結果表示更新間隔の増加割合は, クライアント側の通信環境によって変化するのではないかと考えた.

表 1 クライアントでの結果表示更新間隔 [ms]

	クライアント数				
	1	2	3	4	5
従来システム	34.27				
実装システム	34.38	35.52	35.76	37.26	40.31

6. ま と め

本研究では Web 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をマルチクライアントに対応させた. 現在の実装では,WebGL を用いているがクライアント端末側ではシミュレーション結果の提示だけの処理しか行っていない。 今後は、文献 1) のシミュレーションキャッシングの技術等との連携によりクライアント端末の処理能力も活用した通信の最適化、複数クライアントからの同時入力に対する一貫性制御についても検討を行っていく予定である。

7. 謝 辞

本研究の一部は JSPS 科研 基盤 (B)25280042 ならびに挑戦的萌芽 25540043 の助成による。

参 考 文 献

1) 橋本健介, 手塚俊作, 森眞一郎, 富田眞治: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, 情報処理学会論文誌：コンピューティングシステム, Vol.5, No.4, pp.76-86,2012.
2) 山元祐弥, 辺見良太, 福間慎治, 森眞一郎:対話型遠隔シミュレーションのための Web ベースインタフェースの実装, 先進的計算基盤システムシンポジウム SACSIS, 2012
3) WebGL Specification Ver.1.0, 10 Feb. 2011.
4) The WebSocket API Editors's Draft 8 Feb.2012, (http://dev.w3.org/html5/websockets/)
5) pywebsocket Project Home, http://code.google.com/p/pywebsocket (2013 年 2 月 5 日)