



研究技術ノート

大規模シミュレーションを中心に据えた遠隔研究システム II

菅原章博, 岸本泰明

京都大学大学院エネルギー科学研究科

(論文受付: 2010年11月9日 / 論文受理: 2011年2月1日)

近年, 理工学分野における大規模シミュレーションの重要性はますます高まっている. このような状況を受け, 我々は, 大規模シミュレーションを中心に据え, 地域的・分野的に分散した研究者が共同で研究を推進するシステム "SIMON (Simulation MONitoring) system" [1] を開発している. 本システムは, スーパーコンピュータ上で起動しているシミュレーションコード (クライアント) が, その実行中に, データ解析や可視化, Web 配信等の依頼情報を外部計算機 (サーバ) に送り, これを受けたサーバが依頼情報に従って自動的に処理を行うクライアント・サーバモデルに基づくものである.

本論文では, システムのネットワーク接続における信頼性を高めるために, SSH の login shell を動的に構築する方法や, パスワードを複数の暗号化によって変換する方法を導入し, セキュリティレベルの向上を図った. また, Web 上での効率的な解析環境を構築するため, 可視化アプリケーションの持つ機能を目的に応じて階層化する手法を考案した. そして, これらの機能を有した本システムを大規模シミュレーション研究に導入し, その有効性・有用性を検証した.

Keywords:

large scale simulation, remote collaboration, simulation monitoring, update processing, trigger method, hierarchical visualization, network security

1. はじめに

近年, 様々な物理現象を高い精度で予測できるシミュレーション研究が, 多くの科学・技術プロジェクトや基礎研究分野で重要な役割を果たしている. これは, 計算機性能が日増しに向上し, また, 様々な先進的な数値アルゴリズムや並列計算手法が進展していることからわかる. しかし, このような計算機性能や計算手法の進展だけでは, シミュレーション研究を成功に導くことはできない. 近年の大規模シミュレーションは, 装置を用いた大規模実験に近く, 膨大なデータから目的に適った現象を見出すのは研究者個人というよりは, 研究グループの総合的な能力によるところが大きい. 大規模な実験では, 多くの研究者・技術者が様々な役割分担の下に研究に参画することで目標が達成される. 同様に, シミュレーション研究でも, 計画の立案やシミュレーションの実行, データ処理や画像解析等, すべてを一人の研究者が行うには困難であり, 研究グループとしての取り組みが不可欠になる.

この視点に基づいて, 我々は, 大規模シミュレーションを中心に据え, 容易に入手可能で標準的な可視化技術やネットワーク技術を応用し, 多くの共同研究者が時間と場所を選ばず密接に連携を図りながらシミュレーション研究に参加できるシステム SIMON (Simulation MONitoring) system を考案した [1, 2]. これは, 大規模シミュレーションのモニタリングに主眼を置き, Web を用いてシミュ

レーションの途中経過を配信するシステムである. このシミュレーションモニタリングは, 研究者が実時間で向き合って解析を行う「リアルタイムモニタリング」[3-5] とは異なった概念によって行われる. それは, 長時間を要する大規模シミュレーションの進行状況や最新結果を研究者にとって最適な時間間隔で更新・入手する方法であり, 我々はこれを「アップデート処理」と呼んでいる. また, 本システムの起動やトラブルがシミュレーションの実行そのものに影響しないよう, 「トリガー送信手法」と名付けた, アップデート処理に関する依頼情報だけをシミュレーションが発行する手法を開発し, これによってシミュレーションの実行と研究活動という異なった時間スケールにおける動作の連動を図った.

また, 可視化アプリケーションの機能を利用者のニーズに従って階層的に分割し, Web の持つ相互通信機能と融合させることで効率的な可視化環境を構築する. これによってシミュレーション結果を共同研究者が OS 環境等に依存せず, 同等の可視化環境で閲覧することができる.

一方, 本システムは, クライアント・サーバ間で密接に情報交換を行うため, セキュリティに格段の配慮が要求される. このため, 本システムでは, SSH と Expect コマンドを利用して動的な接続を行う一方, 接続の際に要求されるパスワードは, RSA 暗号 [6], 乱数, 換字式を組み合わせた独自の方式で暗号化する方法を考案した.

Remote Collaboration System Based on Large Scale Simulation II

SUGAHARA Akihiro and KISHIMOTO Yasuaki

authors' e-mail: sugahara@center.iae.kyoto-u.ac.jp and kishimoto@energy.kyoto-u.ac.jp

2章ではSIMONシステムの概要を解説し、3章では、大学・研究機関におけるネットワーク環境を考慮したセキュリティ強化型のトリガー送信手法の導入経緯やその内容を述べる。またWebsiteからの再描画や、可視化アプリケーションの起動、可視化シナリオの構築等、可視化を階層的に取り扱う方法について4章で解説する。5章では、本システムを大規模シミュレーション研究に導入し、その際に現出した問題点とその解決方法等について述べる。6章ではまとめと今後の課題について議論する。

2. SIMONシステムの概要

SIMONは、1章でも述べたように、シミュレーションの実行中にデータ転送や解析、可視化作業などの依頼を外部計算機に送るSIMON-Clientと、その依頼内容を実行するSIMON-Serverが対になってシミュレーションの実行から解析までを遠隔環境で行う「クライアント・サーバモデル」を用いて制御するシステムである。図1はSIMONで構成されたシミュレーション研究環境を模式的に示したものである(参考文献[1]の図3参照)。大型計算機(図1中のSuper Computer)で実行されているSIMON-Clientを組み込まれたシミュレーションコードは、指定されたタイミングで外部計算機に「依頼」(図1中のTrigger)を行う。この際、「依頼」はシミュレーションの実行に影響を与えないトリガー送信手法で行い、依頼を受けたSIMON-Serverはデータ転送(図1中のData transfer)や、そのデータを基にした解析・画像処理(図1中のSimulation dataからVisual dataへの工程)などを自動的に実行する。

この一連の動作は、「依頼」のタイミングと連動した擬似的なリアルタイムで行われ、一般的に呼ばれる「リアルタイム処理」や、シミュレーションの終了後に行う「ポスト処理」とは区別して「アップデート処理」と呼んでいる[1]。これは、本体プログラムに画像処理等のルーチンを

直接組み込んでシミュレーション実行中に解析作業を行うリアルタイム可視化とは基本的に異なった概念である。実際、数時間から数10時間、数日から数カ月に及ぶ大規模シミュレーションでは、これらの解析作業をリアルタイムで行う意味はなく、シミュレーションが進行する典型的な時間スケールと連動した適切な時間間隔で行うことが重要である。SIMONは、シミュレーションから送信されるトリガーのタイミングを調整することでこれを実現する。また、同様のタイミングで解析・可視化結果をWeb Serverによって配信することで(図1中右側)、地域的に分散した共同研究者への情報共有を図り、大規模シミュレーション全体の最適化・効率化を図る。

3. SIMON-Client・SIMON-Server間のセキュリティ

トリガー送信を行う際のクライアント(SIMON-Client)からサーバ(SIMON-Server)へのアクセスや、SIMON-Serverから計算機へのファイル転送等の通信はSSHで行う。これは、現在の各大学・研究機関におけるセキュリティポリシーに対応するためであり、従来の計算機の接続方法と同じ方法を利用することで特殊な許可を求めることなくSIMONシステムが利用できるように考慮した。当初は、TCP/IPのSOCKETを用いたデータ通信を利用する手法も検討したが、いずれの大学・研究機関も一般的に必要とされるPORT以外は開放を認めない場合が多いため、現在はこの方法を採用していない。

一方で、ポートを固定したSSHによる接続方法はセキュリティに対して解決すべき課題が存在する。SIMONは動的な自動接続が必要であり、手法として2種類の方法が考えられる。一つは、SIMON-ServerとSIMON-Clientの間の接続許可を設定し、常時、パスワードを必要としないloginのアクセス方法が考えられる*1。しかし、この場合、SI-

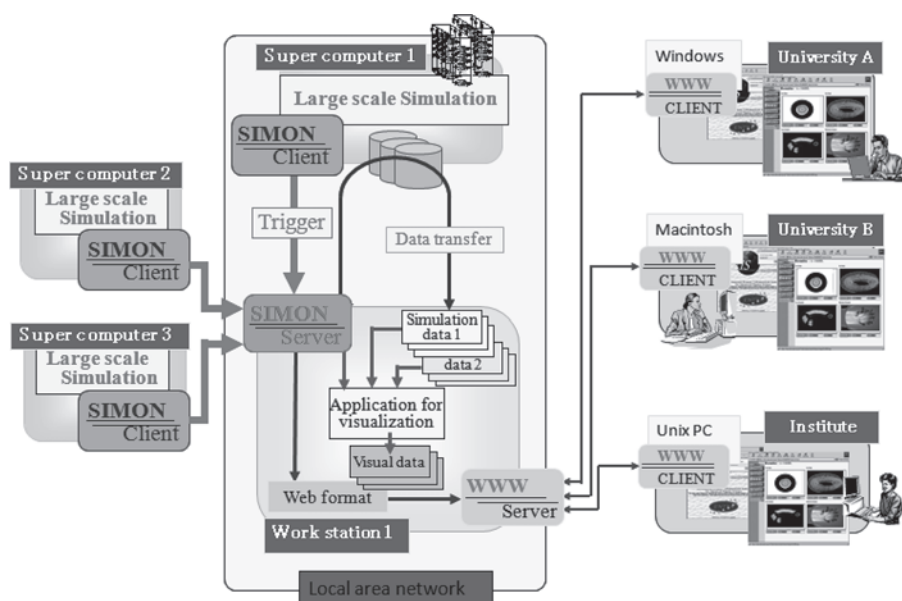


図1 SIMONシステム概要図。

*1 一般的には、Non Password SSH Login と呼ばれる手法やパスワードをメモリ上に記憶させる ssh-agent を利用する手法がある。

MONシステムが利用する以外にも設定が有効になってしまい、SIMON-Serverを設定する外部計算機とコードを運用する計算機のどちらか一方でもパスワードが盗まれた場合などに被害の拡大する恐れがあり、運用方法として好ましくない。したがって、ここでは次に述べる2つ目の方法を選択する。

それは、SIMONシステム内で、パスワードを含んだExpectスクリプトを動的に作成し、これを実行することによって計算機へのアクセスを行う方法である。この場合、パスワードのファイル保存が必要であるが、一般的にパスワードをファイル等の媒体に保存すべきではない。このため、SIMONでは、パスワード文字をそのままファイルに保存するのではなく、パスワード文字にSIMON特有の暗号化を行い、利用直前に復元して接続する方法を採用している。これにより、信頼性の高いセキュリティが保証される。以下にSIMONシステムにおけるパスワードの暗号化および復号化について解説する。

現在、暗号化は共通鍵暗号方式（秘密鍵暗号方式）と公開鍵暗号方式の2種類が主流であり、安全性や高速性、様々な用途等によって使い分けられる。前者の方式で有名なのはDES暗号であり、後者の方式はRSA暗号である。SIMONではRSA方式の基本的な機能を利用する。以下は、暗号化（(1)式：平文 m を c に暗号化）および復号化（(2)式：暗号文字 c を平文 m に復号化）に用いられる式である。

$$c = m^e \pmod{n} \quad (1)$$

$$m = c^d \pmod{n} \quad (2)$$

一般的なRSA暗号化では、 e に65537（ $=2^{16}+1$ ）を利用しているのに対して、SIMONでは複数の e を導入する。つまり、 e に対応した $n(=p \times q)$ や $d(=e^{-1} \pmod{(p-1)(q-1)})$ ： p 、 q は素数）も合わせて用意し、それらをユーザは暗号化パスワードの作成の際に選択できるようにした。また、この他に、暗号化は文字（英数字（62文字）・特殊文字（30文字））の換字による数値化および乱数によるスライド等を行っている。乱数の導入は、一般的に暗号化にとって脆弱とされる文字（1文字だけでもしくは、同一文字）からの読解を防ぐためである。

また、これら暗号化・復号化プログラムにおいては、セキュリティホールの有無を確認するため、オーバーフロー等による調査を行った。プログラムは異なる言語の結合によって構成されているため、変数引き渡しの都合上、取り扱うパスワード文字を40字以内としている。暗号化の際、対象とする文字が既定の40字を超えていると、暗号化プログラムが起動する前に強制的に停止する仕様となっているため、基本的にはバッファオーバーラン等の誤作動は起きない。また、復号化の際には40文字以上の暗号化パスワード（本システムにおいて復号可能な40文字以上の暗号化文字は存在しない）を与えると文字数を確認するルーチン内で復号化対象から外れ、正しく復号できずにトリガー送信に失敗する手続きと同様の動作を行うことを確認した。以上の評価結果から暗号化・復号化プログラムのセキュリ

ティホールはないと判断している。

4. 可視化の階層化

2章で説明したように、SIMONは、シミュレーションの実行に影響を与えることなく外部計算機（SIMON-Server）で自動的に可視化作業を行うことができる。しかし、そのためには、基本データの他にいくつかの制御パラメータが必要になる。例えば1次元グラフを作成するには、縦軸・横軸の最大値・最小値等のパラメータが、2次元・3次元の可視化で等値面表示やベクトル表示を行う場合には更に多くのパラメータが必要になる。自動可視化を行う場合、状況に応じて経験的に設定することもできるが、新しい現象の発見をめざしたシミュレーションでは、これらパラメータを予想することは困難である。これは、可視化アプリケーションを起動してインタラクティブな環境で可視化作業を行えば克服できる問題である。これらの問題に対して柔軟に対応するため、我々は可視化アプリケーションの機能とWebの持つ相互通信機能を合わせることで、ユーザの利用環境に応じた可視化機能の構築を試みた。以下に、可視化作業の役割を階層別に述べる。

4.1 第1階層「基本解析階層」

第1階層はトリガー送信によって行われる画像や動画の作成、表示のためのフォーマティング（Webページの設定）を行うものであり、これを「基本解析階層」とする。ユーザは事前に予測した可視化パラメータをSIMON専用のインプットファイルに記入しておく。この際、各項目を空白指定することで可視化アプリケーションの持つ自動スケラビリティ機能を利用して、有限数値を表示することは可能である。

この結果についてはWeb Browserから確認することができる。利用するHTML（本システムではPerlによって作成）は、文字、画像、動画といった異なるデータを容易にリンクすることが可能であり、導入予定のデータベースについても一般的に相性が良いとされている。

また、情報共有を目的としているため、シミュレーション結果をWeb上に公開しているが、Website閲覧や可視化処理のWeb Serverに掛かる負荷等を考慮し、研究プロジェクトの規模は多くとも100人以下を想定している。このため、閲覧に必要なIDとパスワードの配布や、Web Serverの設定によってWebsiteへの接続制限を設けて対応している。

4.2 第2階層「Web解析階層」

可視化アプリケーションの持つインタラクティブ性を発揮するため、Web Browserから可視化パラメータを変更する機能を持たせ、画像の再描画等を行えるようにする。この機能を「Web解析階層」とする。Web解析階層ではグラフの各軸における最大値・最小値の設定やタイトルの追記等、一般的に変更する必要があるパラメータについて操作できるようにした。図2はWeb解析階層によって画像を再描画させた例である。例えば、図2右上で示すグラフでは、シミュレーション実行前に縦軸・横軸の最大値・最小値を予想して描画した。しかし、このグラフから現象を十分に

把握することは困難である。そこで図2左上のように、空欄だった各項目(縦軸、横軸の最大値や最小値や、X軸のlogスケールおよび各タイトル)を埋めることで(図2左下)、図2右下に示すような詳細なグラフを作成することができる。

一方、シミュレーション研究では、複数の入力パラメータに対するシミュレーション結果の解析を行う場合や、一つのシミュレーション結果に対して複数の解析コードを用いて解析を行う場合がよくある。このような場合には、可視化パラメータを統一して結果を比較・検討しなければならない。通常、この種の問題に対応するため、可視化アプリケーションにはマクロ機能が搭載されており、この機能を用いて作業効率を向上させることができる。本システムでも、可視化の制御パラメータや工程を Visualization ファイルとして保存することで(シナリオ化)、可視化アプリケーションの持つマクロ機能と同様の効果を引き出している。また、シナリオ化された Visualization ファイルは Web 上で選択することが可能であり、共同研究者による共同作業を行うこともできる。

4.3 第3階層「詳細解析階層」

前章で述べたように、Web 解析階層では Web Browser から可視化結果の再描画が可能である。しかし、更に細密な画像を作成する必要がある場合や3次元描画における視点の変更などは、Web 解析階層における再描画機能では対応しにくい。また、端末 PC の計算能力が向上している昨今、可視化アプリケーションをエンドユーザ側で直接扱った方がよい状況も考えられる。このような場合を考慮して、SIMON では出力データと各アプリケーション対応のマクロファイルを Web Browser からダウンロードできるようにしており、端末 PC で起動する可視化アプリケー

ションと合わせれば、使い慣れた環境で解析作業を行うことができる。この他にも、ユーザは X window system の環境を整えることで、可視化アプリケーションを遠隔利用することができる。また、Paraview[7]のような Client-Server 機能を有するアプリケーションを起動することで状況に応じた解析作業を行うことができる。このようにSIMON は、Web 上からの可視化アプリケーション起動を「詳細解析階層」としてサポートしている。なお、現在、「詳細解析階層」での対応可能な可視化アプリケーションは、Gnuplot と Paraview である。ただし、マクロ処理が可能であり、かつ、UNIX OS 対応の可視化アプリケーションであれば本システムへの導入は可能である。

5. SIMON システム導入の手順と留意点

ここでは、これまでに概説した SIMON システムをシミュレーション研究に適用した例について述べる。シミュレーションコードには、高強度レーザーと物質との相互作用などの研究をするための実空間三次元の相対論的粒子コード:EPIC3D[8]を使用する。またハードウェア環境は、大型並列計算機・T2K(京都大学学術情報メディアセンター)[9,10]と SIMON-Server を設置した Linux の PC を用いる。いずれの計算機も学内に設置されているが、ネットワークはグローバルアドレスを持ち、学外からでも接続が可能である。

5.1 SIMON-Client の導入

SIMON-Client は、Fortran サブルーチン(ライブラリ: libSIMON.a)の形で提供する。そのため、シミュレーションコードのメインソースに SIMON-Client ルーチンの呼び出し文(CALL SIMONsystem_****)を挿入し、ソースのコンパイル時にライブラリを指定するだけで導入できる。また、最新システムで用意されている SIMON-Client の各ルーチンは、表1のとおりである。ユーザは状況に応じて、これらルーチンを使い分けることができる。図3は、メインソースに対して SIMON-Client ルーチンの呼び出しを挿入した一例である。(図3中15行目が時系列データの可視化に関するルーチン"SIMONsystem_plot"の挿入例および25行目がスナップデータの転送・可視化に関するルーチン"SIMONsystem_snap"の挿入例)

また、導入するシミュレーションコードが MPI (Message Passing Interface) 対応の場合、SIMON ルーチンは1 CPU で実行されなければならないため、図3中の2行目から4行目に示したような MPI 関数を用いた計算 CPU の番号付けおよび図3中の9行目で示したような計算 CPU によって動作を限定することにより(図3においては、計算 CPU に割り振った番号が0番であった場合のみ、SIMON

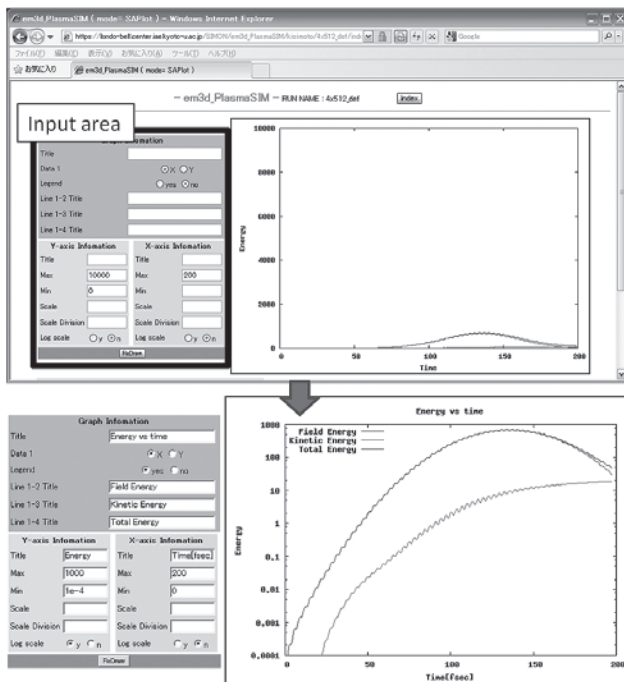


図2 Web 解析階層における可視化の一例。

表1 SIMON-Client ルーチン。

SIMON ルーチン名	用途
SIMONsystem_init	SIMON 利用の初期設定: 必須
SIMONsystem_plot	時系列データの可視化
SIMONsystem_snap	スナップデータの転送・可視化
SIMONsystem_anim	アニメーション作成

```

01 C---MPI control
02   call mpi_init(ierr)
03   call mpi_comm_size(mpi_comm_world, nnn, ierr)
04   call mpi_comm_rank(mpi_comm_world, me, ierr)
05 C--- Calculate
06   do 100 i_time=1, iend_time, idt
07     call e_magnetic(a(kall), i_time)
08
09     if (me. eq. 0) then
10 C----- Plot data out for SIMON
11       SIMONE(1)=i_time
12       SIMONE(2)=energy_f
13       SIMONE(3)=energy_k
14       SIMONE(4)=energy_total
15       call SIMONsystem_plot(SIMONE)
16 C----- Snap data out for SIMON
17       DO i=1, Nx_d
18         DO j=1, Ny_d
19           SIMONS(i, j, 1, 1)=Ele_density(i, j)
20           SIMONS(i, j, 1, 2)=Ion_density(i, j)
21           SIMONS(i, j, 1, 3)=Ele_Temp(i, j)
22           SIMONS(i, j, 1, 4)=Ion_Temp(i, j)
23         enddo
24       enddo
25       call SIMONsystem_snap(
26 +     i_time, SIMONS, Nx_d, Ny_d, 1, 4)
27     endif
28 100 continue
    
```

図3 プログラムソースへの SIMON-Client 導入例.

ルーチンが実行されるようプログラミングされている), SIMON-Client ルーチンの実行を制限する必要がある.

5.2 並列計算機におけるトリガー送信

並列計算機で SIMON-Client を起動する場合, トリガー送信は外部計算機にアクセスできない可能性が高い. これは計算を行う CPU (計算 CPU: 図4中の CPU 1~n) がプライベート IP アドレスを使用しており, 外部計算機への接続を考慮していないためである. しかし, このような計算機では, 通常, グローバル IP アドレスを持った Front CPU が設定されており, 計算機を利用する際, ユーザはこの Front CPU に接続する. したがって, トリガーも Front CPU を介して外部計算機に送信するように構築した.

並列計算機は, 通常, MPI で複数の CPU が協調動作を行うよう設定されている. この際のネットワーク構成は MPI 構築上 (例えば MPICH[11] で構築した場合), プライベート IP アドレスと RSH, もしくは SSH のようなシェルコマンドでリングを構築している. (図4). この特徴を利用して, 並列計算機上でトリガー送信を行う場合, SIMON-

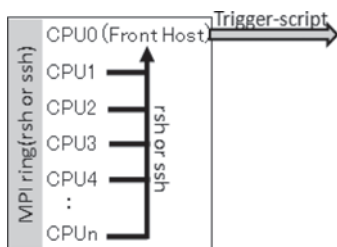


図4 並列計算機におけるトリガー送信.

Client は Expect スクリプトを二つ作成し, トリガーの二段送信を行う. スクリプトの一つは, プライベート IP アドレスを使って計算 CPU から Front CPU へのトリガー送信を行うものであり, もう一つは, グローバル IP アドレスを使って外部計算機へのトリガー送信を行うスクリプトである. ただし, SIMON-Server から実行するデータ転送は, 上記のような二段実行に対応していないため, NFS 等のファイル共有設定により Front CPU からデータ格納位置が特定できなければならない.

5.3 データ出力タイミングへの適応

シミュレーションコードにおける出力データ作成のタイミングは, 図5に示すような3つのタイプが一般的に考えられる. 図5中の Type A は一般的なファイルの出力方法を示し, 計算ループが終わると密度や温度等の変数別にファイルが出力される. この場合, 解析作業が行いやすい一方, ファイル数が膨大になる. 図5中の Type B は, 計算ループ毎に一つのファイルが出力される方法であり, Type A と比べるとファイル数が軽減する. しかし, 解析作業の際には Filter 等でファイルを分割する必要がある. 図5中の Type C は, 並列計算機で実行するシミュレーションコードによく見られるデータ出力方法であり, 並列書き出しの観点から有効である.

Type A や Type B は計算ループが終了すると一定の間隔

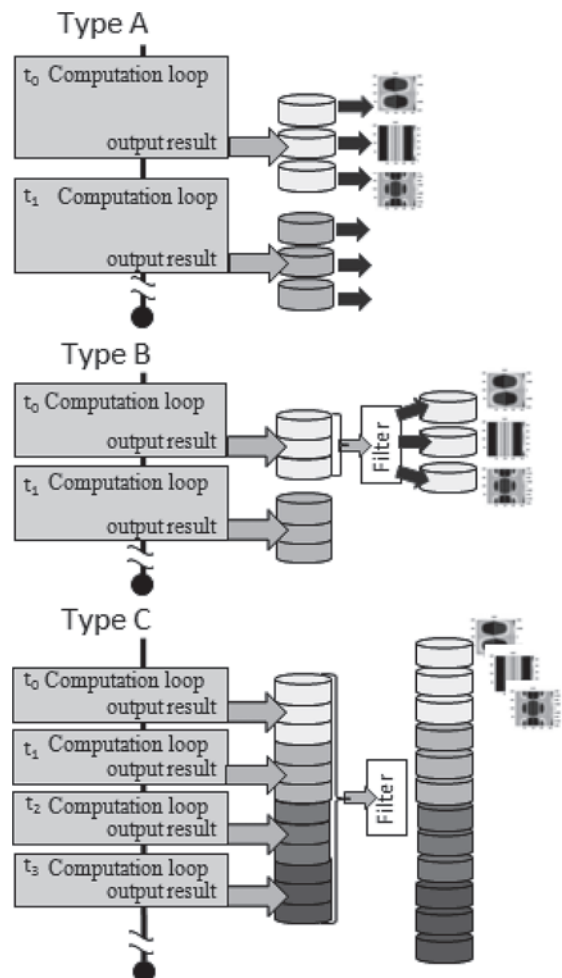


図5 シミュレーションコードにおけるファイル出力形式.

で計算結果をファイルに出力し、以後、ファイルは追記されることなく書き込みを終了する。この場合、SIMON は、作成されたファイルを転送し、可視化作業等を行うことができるため、シミュレーションの疑似的なモニタリングを行うことができる。ただし、シミュレーション実行中において、write 文（ファイルへの書き込み命令）を実行したにもかかわらず、メモリ内にデータを保存したままで、ディスクへの書き込みが完了していないか、あるいは、書き出しが行われていない状況が実際に確認されている。この場合、ファイル作成の完了を待たずにトリガー送信を行ってファイル転送を開始してしまう可能性があるため、SIMON 内では、Fortran ライブラリとして用意された flush を呼び出すことでこの問題に対応している。ただし、実行環境によっては、稀に flush の呼び出しだけでは不十分な可能性が残されているが、現在、この問題には対応していない。また、ファイル出力形式が Type A や Type B である場合、システム導入には、図 3 中 25 行目に書かれたような

SIMONsystem_snap における各引数 (i_time, SIMONS 等) の設定は必要ない。この場合、ファイル名やタイミングの指定はすべて SIMON 専用インプットファイルで行う。

一方、Type C のように計算が終了するまでファイルへの書き込みが終了しない場合、書き込み中のファイルを転送することはできないため（転送は可能であるが、ファイルが壊れてしまう可能性が高い）、モニタリングを行うことができない。そこで Type C のような、ファイルの出力形式を持つシミュレーションコードに対してもモニタリングが可能となるように、SIMON-Client は解析対象となるデータ変数を Call 文の引数として指定することで、時間毎のファイル作成を行えるようにした。これによって SIMONsystem_snap が起動される毎にファイルの作成・転送・可視化までの工程を自動的に行うことができ、シミュレーションのモニタリングが可能となった。図 3 は、シミュレーションコードの出力形式が Type C である場合の導入例を示しており、図 3 中 19 行目～22 行目に示したように各データ変数の内容（電子密度、イオン密度、電子温度、イオン温度の 4 変数）を変数名 SIMONS に代入した後、SIMONsystem_snap に SIMONS と格子数（図 3 中では $N_x _d, N_y _d$, 1 と記載）、データ変数の数（図 3 中で 4 と記載）を引数として渡している。なお、ここで示したような導入例は、ファイル出力をまったく行わないようなシミュレーションコードに対しても用いることができる。

5.4 システムトラブルによるシミュレーション実行への影響

SIMON 実行時において、考え得るトラブルとして以下を想定し、意図的にこれらのトラブルが発生するような環境を作って実行した。

- A) ネットワークの停止 (SIMON-Server 設置 Work Station の停止およびシミュレーション実行中に復帰し

た場合を含む

- B) SIMON 専用インプットファイルの有無
- C) SIMON 専用インプットファイルの作成ミス（トリガー情報：Server および Client のユーザ名やパスワード、ホスト情報等）
- D) SIMON 専用インプットファイルの作成ミス（可視化情報：データ保存位置（ディレクトリ）や、格子サイズ、可視化パラメータ等）
- E) ファイル出力形式 Type C におけるファイル作成時におけるオーバーフロー

実行した結果、仮定した A), B), C) のトラブルについては、トリガー送信が正常に行われませんが、シミュレーションの実行が停止することはなかった。また、シミュレーションは正常に終了したため、解析に必要なファイルはすべて出力された。したがって、シミュレーション終了後に、再度、正常な環境下で手動によるトリガー送信*2を行うと、シミュレーション結果の Web 配信等が正常に行われることを確認した。

次に、D) については、Web 上でシミュレーション結果の適切な表示が行われなかったが（設定によっては画像ファイルの作成が失敗）、大半は Web 解析階層による再描画によって画像が正常に作成され、表示することができた。画像が作成できなかった事例は、SIMON 専用のインプットファイルに誤ったメッシュサイズを与え、Filter 起動時（図 5 中 Type B および Type C にあるような Filter によるファイル分割作業）に正常なファイル分割が行われなかった場合である。ただし、これは 6 章で述べる今後導入予定のインプットファイルの簡略化で対応できると考えられる。

最後に、E) については、ファイル出力が SIMON 専用のインプットファイルで設定した範囲を超えないようにプログラムしているため、一つの変数に対して宣言領域を超えるようなデータを与えても、指定した範囲内しかでファイル出力が行われず、出力エラー等は発生せず、シミュレーションが停止することもなかった。

以上の結果から、SIMON のトラブルによってシミュレーション実行が停止することは高い確率でないと判断される。ただし、SIMON 専用インプットファイルに誤りがある場合、Web 配信への適用等が困難になることを確認した。

5.5 適用例 (EPIC3D)

図 6 は、EPIC3D における放電現象の解析結果を示したものである。これは、高圧ネオンガス中（密度： 1.2×10^{20} [cm^{-3}], 粒子数 900 万個）に高電圧 (2.5×10^6 [V/cm]) を印加したとき、初期に配置した Ne^{+2} の微小な電離スポットから放電が進行する様子をシミュレーションした結果である。シミュレーション領域は $(x, y) = (32, 128 [\text{mesh}]) = (0.2 \times 10^{-3}, 8 \times 10^{-3} [\text{cm}])$ である。計算時間は、64CPU で 20 時間程度である。

* 2 トリガー送信は、SSH によって実行可能なため、ssh user@SIMON-Server-host "/trigger optionA, optionB, optionC" というような状況に応じた正しいオプション群を用いて実行することで、ユーザが手動で行うことができる。

図6①は、異なった電荷を持つイオン密度の時間発展を示しており、シミュレーションの実行中に、これらの値が一定時間間隔で更新されている。これは、シミュレーションの正常動作の目安になる。また、スナップデータについても、図6②で示しているような画像がおよそ2時間毎に追加されWeb上に表示されている。図6③は、作成されたスナップ画像を連続的に並べたものである。計算領域中心に置いた電離スポットが、ある時点から爆発的に増加している様子が確認できる。図6④は、同時刻における電子密度、イオン密度、電子温度を並べてWeb上に表示したものである。

なお、本論文で導入対象としたEPIC3Dは、既に投稿した論文[1]と同様のものを採用しているが、シミュレーションの利用環境やコードのファイル出力形式の大幅な変更等があり、前回の対象としたコードとは異なったものと考えてよい。

6. まとめと今後の課題

前稿[1]および本稿では、科学・技術研究において重要性を高めているシミュレーション研究を効率的に進めるため、大規模シミュレーションを中心に据え、地理的および分野的に分散した多くの共同研究者が参加できる新しいシミュレーション研究環境「SIMONシステム」の提案を行った。このシステムはシミュレーションコードに組み込んだSIMON-Clientが指定されたタイミングでデータ転送・解析、および可視化作業等に関する依頼を外部計算機に送

り、それを受け取ったSIMON-Serverが依頼内容を処理するものである。これを実現するため、開発したトリガー送信手法は、ネットワークのセキュリティや、並列計算機を使用した場合の対応について格段の配慮が必要である。このため、セキュリティ強化型のトリガー送信手法や並列計算機に対応した二段送信手法を考案し、実装した。また、共同研究の効率化を目的として可視化機能を階層化し、Webを積極的に利用することで情報の共有化を図った。

これらのシステムは、前稿[1]でも述べたように、システムの汎用性を高めるため、既存技術の応用やそれらの組み合わせによって構築している。したがって、システムの導入やWeb解析階層等、システムの利用については、特別な知識を必要とすることなく利用できる。また、5.4節でも述べたように、本システムの導入や関連トラブルの発生によるシミュレーション実行の停止はなく、システムの有用性・有効性を確認した。

今後は、前稿[1]から未解決であるシミュレーション結果のデータベース化やデータマネージメントに関連したシステムの拡張を行う必要がある。また、トリガー送信手法の拡張や可視化の階層化を行ったため、SIMON専用インプットファイルには、多くの情報が必要になってきている。このため、容易に本システムの導入を行うためには、インプット情報の簡略化が望まれる。したがって、今後SIMON-Client内で入力値のチェックを行う機能（チェックユーティリティの導入）や、シミュレーションの実行状態（トリガー送信に必要な計算機のホスト名・アカウント

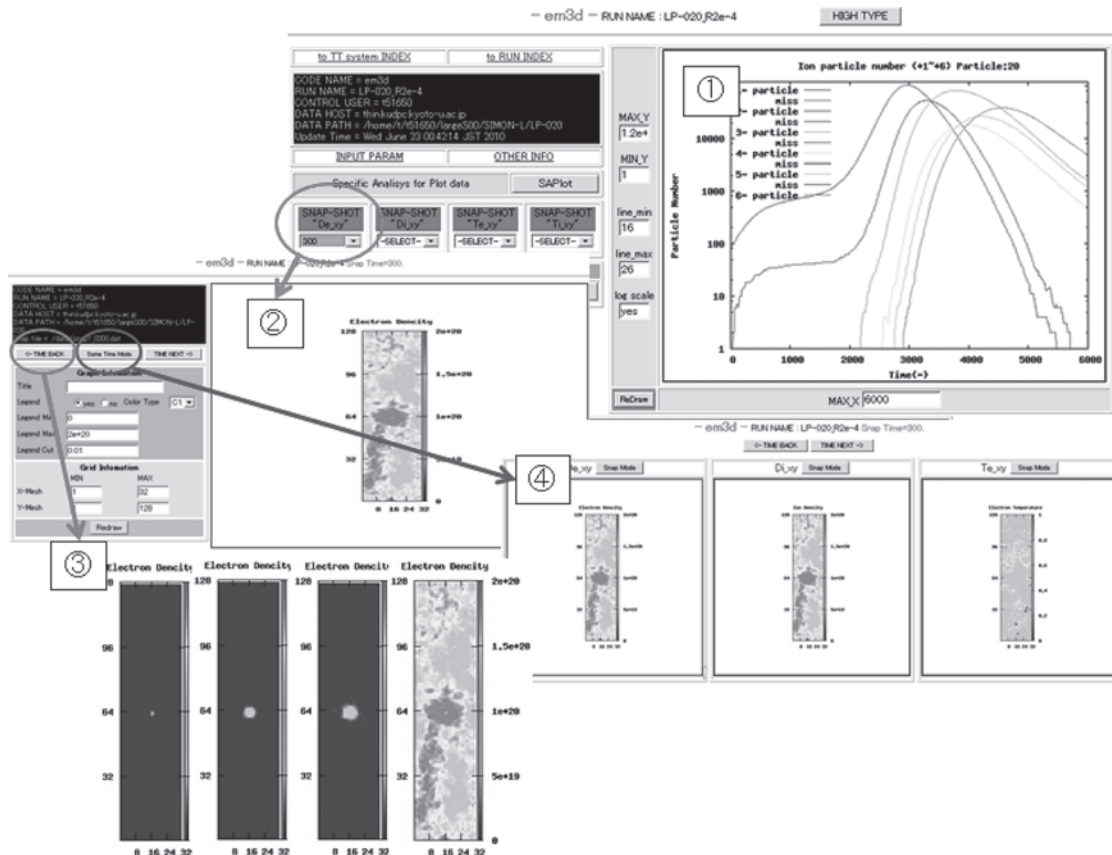


図6 EPIC 3D適用例 (Lightning 解析)

情報や、可視化に必要なメッシュ数等のシミュレーション規模に関する情報)を把握できる機能を導入することで、ユーザの負担を軽減したい。

さらに、より汎用的で多様な研究環境に対応するため、トリガー送信手法についてはSSHが許可されていないネットワークエリアへの接続手法を確立しなければならない。例えば、ITBL[12]を利用する方法や、SIMON-Serverを各大学・研究機関で分散的に配置し、Serverが連動することで各研究者に必要な情報を共有できるシステム等の構築が考えられる。本研究は科学研究補助金(No.21340171)の一環として実施されました。

謝 辞

本論文におけるシステム開発について貢献および有益な助言をいただいた今寺賢志博士に感謝します。また、本文5章におけるシステム導入に際しては、正木知宏博士に支援していただき、感謝いたします。

参考文献

- [1] 菅原章博, 岸本泰明: プラズマ・核融合学会誌 84, 51 (2008).
- [2] Y. Kishimoto, A. Sugahara and J.Q. Li, Fusion Eng. Des. 83, 434 (2008).
- [3] “リアルタイム可視化システム RVSLIB”, 武井利文, 松本秀樹, 土肥 俊: 第10回計算力学講演会講演論文集 (1997) p.413.
- [4] 小笠温滋, 南多善, 小澤亮夫: 計算工学講演会論文集 4, 337 (1999).
- [5] “ライブラリレス・リアルタイム可視化システム” 小笠温滋, 森屋光弘, 中島拓之, 藤井孝藏, 田村善昭 <http://www.kgt.cybernet.co.jp/viz-journal/contents/122/index.html>
- [6] "A Method for Obtaining Digital Signature and Public-key Crypt systems" R.L.Rivest, A.Shamir, and L.Adleman, MIT Laboratory for Computer Science Technical Memo LCS/TM82 April 4, 1977.
- [7] Paraview ホームページ <http://paraview.org/>
- [8] Y. Kishimoto and T. Masaki, J. Plasma Phys. 72, 971 (2006).
- [9] T2K ホームページ <http://www.open-supercomputer.org/>
- [10] 京都大学学術メディアセンターホームページ Open Super computer <http://web.kudpc.kyoto-u.ac.jp/hpc/super-computer>
- [11] MPICH2 ホームページ <http://www.mcs.anl.gov/research/projects/mpich2>
- [12] ITBL ホームページ <http://www.itbl.jp/>