

---

---

# グリッド上における統合シミュレーションの 連携実行フレームワーク

(独) 日本原子力研究開発機構

---

## ■ 執筆者Profile ■



立川 崇之

2009年 (独) 日本原子力研究開発機構入社  
現在 システム計算科学センター  
高度計算機技術開発室所属  
博士研究員  
原子力情報基盤システム研究 担当



中島 康平

2004年 (株)日立東日本ソリューションズ 入社  
2004年9月 日本原子力研究所  
計算科学技術推進センター  
(現 日本原子力研究開発機構  
システム計算科学センター)  
グリッド技術研究開発支援担当  
2009年9月 (株)日立東日本ソリューションズ  
先端技術ソリューション部  
グリッド基盤技術開発担当



金 奎希

2008年 (独) 日本原子力研究開発機構入社  
現在 システム計算科学センター  
高度計算機技術開発室所属  
博士研究員  
原子力情報基盤システム研究 担当



木野 千晶

2007年 (独) 日本原子力研究開発機構入社  
現在 システム計算科学センター  
高度計算機技術開発室所属  
博士研究員  
原子力情報基盤システム研究 担当



手島 直哉

2001年 富士通株式会社入社  
グリッドミドルウェアの開発  
2007年 日本原子力研究開発機構  
グリッド基盤研究



鈴木 喜雄

2002年 日本原子力研究所入所  
可視化技術研究開発  
2004年 文部科学省研究振興局情報課 (出向)  
2005年 (独) 日本原子力研究開発機構に改組  
現在 原子力情報基盤システム研究総括



武宮 博

1989年 日立東日本ソリューションズ (株) 入社  
2003年 産業技術総合研究所グリッド研究センター  
(派遣)  
2008年7月 (独) 日本原子力研究開発機構入社  
システム計算科学センター  
高度計算機技術開発室室長

## ■ 論文要旨 ■

分散配置された計算機上に実装されたプログラムを協調動作させることにより、原子力プラントの耐力シミュレーションや核融合プラズマの制御シミュレーションなど、大規模かつ複雑な統合シミュレーションを可能とする枠組みについて紹介する。シミュレーション技術の進展に伴い多様なシミュレーション手法を統合することで、より精緻なシミュレーションが実行可能となってきた。しかしながら、個々のシミュレーション手法を実装したコードを組み合わせ、実行制御しながら動作させることは、科学者にとって大きな負担となっている。我々は統合シミュレーションにおける個々の要素プログラムの実行パターンが、ワークフロー型、パイプライン型などに分類できることに着目し、それらのパターンを設定ファイルに記述するだけで要素プログラムを連携動作可能とする機能を実現することで、既存コードを大幅に修正することなく連携動作可能とする枠組みを開発した。

## ■ 論文目次 ■

<b>1. はじめに</b> .....	《 5》
<b>2. シミュレーションの連携実行の現状と課題</b> .....	《 6》
2. 1 シミュレーションの連携実行の現状	
2. 2 シミュレーションの連携実行の課題	
<b>3. 連携実行フレームワーク SOAF の開発</b> .....	《 8》
3. 1 SOAF の目的と概要	
3. 2 原子力グリッド基盤AEGIS	
3. 3 コントローラー	
3. 4 センチネル	
<b>4. SOAF の評価</b> .....	《 12》
4. 1 3次元仮想振動台	
4. 2 地震耐力予測シミュレーションシステム	
4. 3 核融合プラズマシミュレーションシステム	
<b>5. 今後の課題</b> .....	《 16》
<b>6. おわりに</b> .....	《 17》

## ■ 図表一覧 ■

<b>図1</b> 様々な連携実行の型 .....	《 7》
<b>図2</b> AEGIS基盤の概略図 .....	《 9》
<b>図3</b> コントローラーの状態遷移 .....	《 10》
<b>図4</b> コントローラーの設定ファイルの例 .....	《 11》
<b>図5</b> センチネル実行の概略図 .....	《 11》
<b>図6</b> 3次元仮想振動台の構成 .....	《 13》
<b>図7</b> 3次元仮想振動台のファイルフロー .....	《 13》
<b>図8</b> 3次元仮想振動台の動作ダイアグラム .....	《 14》
<b>図9</b> 耐力解析システムのファイルフロー .....	《 14》
<b>図10</b> 耐力解析システムの動作ダイアグラム .....	《 15》
<b>図11</b> 核融合プラズマシミュレーションのファイルフロー .....	《 15》
<b>図12</b> 核融合プラズマシミュレーションの動作ダイアグラムの例 .....	《 16》

## 1. はじめに

独立行政法人日本原子力研究開発機構（以下、原子力機構と略す）は、原子力研究開発を総合的・一体的に実施する先端的な研究開発機関として、科学技術の水準の向上を図り、原子力の利用の高度化及び多様化の推進に貢献する役割を担っている。システム計算科学センターは、計算科学の観点から、これら貢献に資する研究開発を実施している。このため、高度計算科学技術の推進に必要な基盤技術の研究開発及び原子力分野におけるシミュレーション技術の研究開発を推進している。これら研究開発により、従来の「計算科学による理論・実験の支援」から「計算科学による予測工学確立」へのパラダイムシフトを目指している。

原子力分野では分散配置された計算機上に実装された多様なシミュレーション手法を統合することで、大規模かつ複雑な現象の解明を目指した様々な統合シミュレーションの研究開発が実施されている。例えば、3次元仮想振動台[1]、地震耐力予測シミュレーションシステム[2]、核融合プラズマシミュレーションシステム[3]などがある。以下、これら三つのシステムの概要を述べる。

3次元仮想振動台の研究では、地震発生時の原子力施設の耐震性を評価するため、施設全体の解析の実現を目指している。原子力施設が巨大地震にさらされたときの性能限界は、実大実験で実証されるべきであるが、規模やコスト面による制限から実際には不可能である。また、施設の経年変化を考慮した実験も非常に困難である。そのため、実験の代替手段として、シミュレーションを用いることが提案された。ここで、原子力施設は、内部を水やヘリウムガスなどが流れる複雑な構造物であるため、構造／流体／熱連成解析をはじめとする様々な要素技術が必要となる。また、施設の構成部品が非常に多く、扱うデータが大規模になる。したがって、一台の計算機のみでは計算資源が不足するため、構成物を部品単位に分割し、部品間の連成を考慮しつつ、複数の計算機で各部品の計算を実行する方法が用いられている。

地震耐力予測シミュレーションシステムの研究では、原子力施設が巨大地震を受けた際、地震発生直後の過渡状態での施設の機能限界（耐力）を定量的に見極めるため、地殻－表層地盤－建屋－機器間を伝播する地震波動の挙動を明らかにすると同時に、原子炉内部の構造－材料－流体－熱－炉特性の相互作用について評価することが試みられている。空間的スケールでは数百キロメートルから数ミリメートル、時間的スケールにおいても周期としてミリ秒から秒、継続時間として数十秒から数時間のものを含むため、幅広い空間的、時間的スケールを持つ対象を扱わなければならない、各々の現象に特化したシミュレーションコードを統合し、原子力施設の機能限界を評価する方法が採用されている。

核融合プラズマシミュレーションシステムの研究では、原子力機構那珂核融合研究所が中心となり、現在国際プロジェクトとして進められている国際熱核融合実験炉 ITER 計画での連続運転を目指して、シミュレーションの連携実行によりプラズマの制御性を理解することを目的としている。核融合プラズマシミュレーションシステムは電流拡散、プラズマの安定性、電磁波による制御など様々なコードから成り立っており、統合することで現実的なシミュレーションがなされる。プラズマの挙動を理解するためには、様々な物理過程を考慮しなければならない、その空間及び時間のスケールが非常に幅広いため、核融合プラズマのシミュレーションを、一つのシミュレーションコードで行うことは現実的ではない。

例えば、プラズマの加熱、制御に用いられる電磁波の周期は  $10^{-8}$  秒程度である一方、電流の拡散は数秒の間になされる。更にプラズマの挙動では乱流、輸送、電磁波と粒子の相互作用、プラズマとトカマク容器壁との相互作用、核反応など、複雑な物理現象をいくつも考慮しなければならない。これら各々の物理現象は、別々のコードで計算される。これらのコードを統合することにより、様々な空間スケールの物理現象を取り扱い、短時間の現象を考慮しながら長時間のシミュレーションを実行することができるようになる。

このようなシミュレーション技術を用いた解析では、複数のシミュレーションコードを協調動作させなければならないという共通課題がある。この課題は、

- ・ 計算規模が大きすぎて一台の計算機では計算資源が不足している。これは、計算機の性能が向上した場合にも、精度向上などのためにシミュレーションの規模がそれに伴って大きくなっていく場合が多い。

- ・ 個々の現象に着目したシミュレーションコードは、大学や研究機関で個別に開発されてきており、それぞれの計算機の性能を最大限発揮できるようチューニングや性能向上が行なわれる。

などの理由により、一台の超大規模なスーパーコンピュータがあれば解決されるわけではない。したがって、これらのコードを連携実行するには、大学や研究機関にまたがって安全にコード間の通信が行なわれる必要がある。また、利用者端末から手軽にこれら大学や研究機関の計算機資源を活用できる必要もある。このため、グリッドコンピューティング技術の利用が有効であり、原子力分野のみならず、これまで医療分野におけるバイオグリッド[4]や材料分野における階層的分子動力学[5]など様々な研究分野においても利用されてきている。

しかしながら、後述するとおり、様々なシナリオの連携実行に容易に対応できる技術はこれまで十分に実現されておらず、複数のコードを連携して、その実行を制御しながら動作させることは、科学者にとって大きな負担となっている。特に、シミュレーションを実行する研究者がコードの修正などに手を煩わせることなく、各々のコードを連携実行することは困難である。更に、シミュレーションコード実行中の条件分岐により、他のコードが起動されるような連携実行にも十分対応していない。

そこで我々は、各々のコードに大がかりな修正を必要とせず、科学者がコードの実行制御を行うクライアントソフトの設定ファイルを記述するだけで、様々なシミュレーションの連携実行が可能なフレームワークを開発した。各々のシミュレーションコードの連携の形態は、利用者によって動的に変更されること、また、異なる統合シミュレーションにおける連携の形態は一樣ではないことから、様々なシミュレーションコードの連携の形態に対して、柔軟に対応可能なフレームワークが望まれる。我々が開発した連携実行のフレームワークは、原子力分野に限らず、様々な分野でも共通の技術となり得る。

## 2. シミュレーションの連携実行の現状と課題

### 2.1 シミュレーションの連携実行の現状

ネットワーク上に点在する計算機同士を接続し、一つの複合システムとして活用するグリッドコンピューティング技術として、従来、複数の計算機にまたがるシミュレーションを実行するツールが開発されている。例えばワークフロー制御技術として、ITBL 基盤ソフトウェア[6]上で動作する TME[7]、UNICORE[8]に実装された UNICORE workflow engine、

UNICORE や Globus[9]などで利用可能な Kepler[10]などが開発されている。これらの技術により、研究者は、Graphical User Interface(GUI)を利用してシミュレーションコードの連携実行が行える。研究者は各々のコード実行のジョブを示すアイコンをドラッグ&ドロップで組み合わせることで、連携のシナリオを組み立てることができる。また、グリッド環境に対して拡張されたリモートプロシージャコール(RPC)を用いることにより、研究者は様々なシミュレーションコードを協調実行できる。例として GridRPC[11]がある。更に、MPI を拡張したグリッド対応 MPI を用いることにより、異機種 of 計算機間での通信が実現される。グリッド対応 MPI としては、STAMPI[12], MPICH-G[13], PACX MPI[14]などが開発されている。これらのツールを状況に応じて使い分けることにより、シミュレーションコードの連携実行が可能となる。

## 2. 2 シミュレーションの連携実行の課題

しかしながら、これらのツールには得手不得手がある。以下、現存のツールの問題点について考察する。

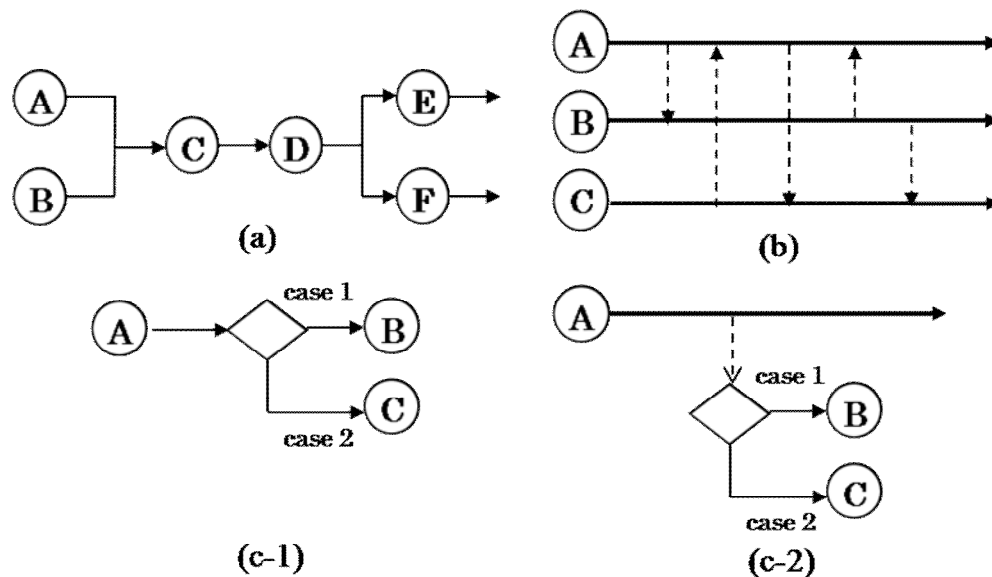


図1 様々な連携実行の型

(a) ワークフロー型. (b)パイプライン型. 破線はファイル転送を示す. (c-1)静的な条件分岐型. (c-2)動的な条件分岐型

シミュレーションコードの連携実行は、図1に示すように主に三つのタイプに分類される。一つ目はワークフロー型と呼ばれるもので、各々のジョブが順序に従って実行されるものである。二つ目はパイプライン型と呼ばれるもので、複数のコードが同時に実行され、各々のコードが実行中に互いにファイルを受送信するものである。三つ目は条件分岐型と呼ばれるもので、シミュレーションの途中での条件分岐により、次に起動するシミュレーションコードが決定されるものである。条件分岐型では、事前にどのコードがどのような順序で起動されるかを想定することができない。条件分岐型は、(c-1)に示すようなジョブ終了後に次に実行するジョブが決定される静的な条件分岐型と、(c-2)に示すようなジョブの実行中に新たに並行して実行するジョブが決定される動的な条件分岐型に細分できる。

ワークフロー制御技術では、(a)に示すワークフロー型の連携実行は容易にできる。しか

しながら、(b)に示すパイプライン型の連携実行に対しては、前の処理が終わってから次の処理を開始するため対応が困難である。(c-1)に示す静的な条件分岐型については対応する技術も存在するが、(c-2)に示す動的な条件分岐型の取り扱いは困難である。

一方で、グリッド対応 RPC や MPI を用いれば、(b)のパイプライン型や(c-2)の動的な条件分岐型の連携実行は可能である。しかし、各々のシミュレーションコードにグリッド対応 RPC や MPI を組み込む作業は大がかりなものとなり、研究者にとって負担が大きい。

### **3. 連携実行フレームワーク SOAF の開発**

#### **3. 1 SOAF の目的と概要**

我々は、研究者に大がかりなコードの書き換えを要求することなく、グリッド上に分散された計算機上の複数のシミュレーションコードを連携動作させ、統合シミュレーションを実行するためのフレームワーク Simple Orchestration Application Framework (SOAF)を開発した。ここで、特に以下の点を特に考慮した。

- 1) 様々なタイプの連携動作に容易に対応できること。
- 2) 各々のシミュレーションコードに対する修正をできる限り少なくすること。
- 3) 連携動作に際し、利用者の手間をできる限り少なくすること。
- 4) SOAF 利用による大幅な時間超過を避けること。

そこで、

- ・シミュレーションコードの修正を減らすには、コード間で行き来するファイルの情報に着目した方が、コードの実行順序に注目するよりも良い。

- ・条件分岐が発生する連携動作を扱う場合、コードの実行順序はシミュレーションの経過に依存し、事前に決められない。

- ・特に動的な条件分岐を扱う場合、コードの起動/終了を他のコードの起動の判断基準とすることができない。

ことから、SOAF では実行コードの順序ではなく、ファイルフローに着目した。我々が想定している統合シミュレーションでは、各々のシミュレーションコードは、最初に実行されるコードを除いて入力ファイルが必要とするため、他のコードによって入力ファイルが生成されることをコード実行の判断基準とする。最初のシミュレーションの実行、及びファイルフローは、クライアント PC に実装されたアプリケーション（以下、コントローラーとする）の設定ファイルに記述する。コントローラーは、各々のシミュレーションコード、及び、これらが実装されたコンピューターでファイルの出力/転送検知の役割をするアプリケーション（以下、センチネルとする）の実行を制御する。我々はこれらのアプリケーションを、原子力グリッド基盤 Atomic Energy Grid Infrastructure (AEGIS) [15]上に実装した。

SOAF 一式の開発に要した期間は、全体で約 8 ヶ月である。前述の四つの点を考慮した設計を行うための協議、構想期間が約 4 ヶ月である。その後、種々の統合シミュレーションの研究者と協議をしながらスパイラル型で SOAF を開発した。この開発期間が約 4 ヶ月である。

SOAF のソースコードの規模としては、コントローラーが約 3700 行、センチネルが約 80 行である。コントローラーについては、3. 2 で述べる認証、ジョブスクリプト作成、ジョブ投入・管理、ファイル転送を司る原子力グリッド基盤 AEGIS のクライアント API を用



いることで、ソースコードの規模及び開発期間を大幅に短縮できている。

### 3. 2 原子カグリッド基盤 AEGIS

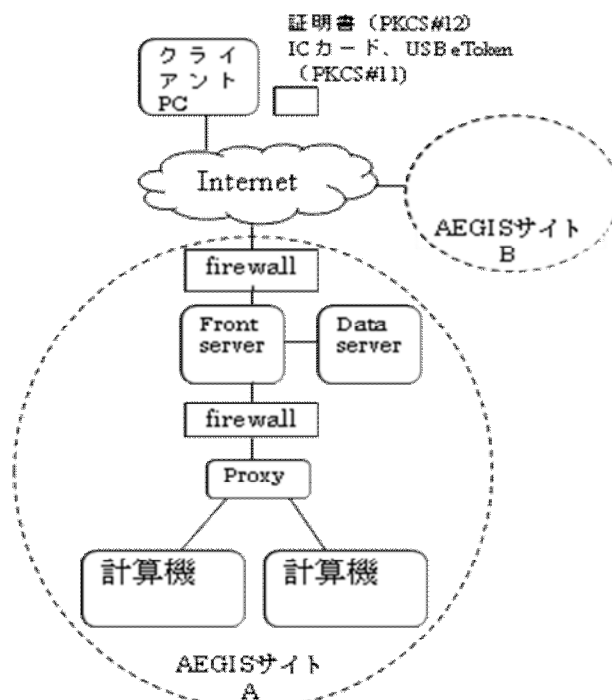


図2 AEGIS 基盤の概略図

AEGIS は、原子力分野に必要な計算科学基盤がどのようなものかを研究者の視点から捉え直し、より利便性の高いグリッド環境を構築することを目指して、原子力機構システム計算科学センターで研究開発されてきた。AEGIS の概略を図2に示す。AEGIS の特徴として、グリッド環境にある各大学、研究機関（サイトと呼ぶ）へはインターネットアクセスを基本としたアクセスを行う。クライアント PC と計算機の間ではセキュアな通信が確保され、各機関の計算機を利用できる環境が提供されている。計算機の利用に際して、利用者はまずクライアント PC から各サイトのフロントサーバにアクセスする。フロントサーバでは電子証明書を格納したハードウェアキーによるユーザー認証が行われる。計算機利用において、利用者はクライアント PC 上でアプリケーションを用いてジョブを制御できる。

AEGIS では、利用者がグリッドに対応したアプリケーションを開発可能とするため、クライアント PC 上で利用可能なクライアント API [16]を開発してきた。クライアント API は Linux, Windows のいずれの C 言語のプログラムでも使用できる。クライアント API は低レベル、中レベル、高レベルの API から構成されており、機能によってレベルが分類されている。低レベル API は、クライアント PC と AEGIS 上の計算機との接続、計算機上のジョブ制御、クライアント PC と計算機双方のリソース管理などを司る。利用者が AEGIS 上の計算機に接続する際には、証明書（PKCS#12）あるいは IC カードまたは USB e トークン（PKCS#11）のいずれかと、パスワードの二要素認証を行う。中レベル API としては、コンピューター名、ジョブクラス、使用 CPU 数、コードのパス、作業ディレクトリなどの情報を入力とし、様々な種類の計算機に対応したスクリプトを作成可能な、ジョブスクリプトジェネレータ API が整備されている。高レベル API としては、パラメータサーベイやフォ

ールトトレラントなど、シミュレーションの高度な実行に必要とされる機能を実現する API が整備されている。

### 3.3 コントローラー

我々は、クライアント PC からシミュレーションコードの連携実行を可能とするため、クライアント API を用いてコントローラーを開発した。コントローラーの状態遷移を図 3 に示す。

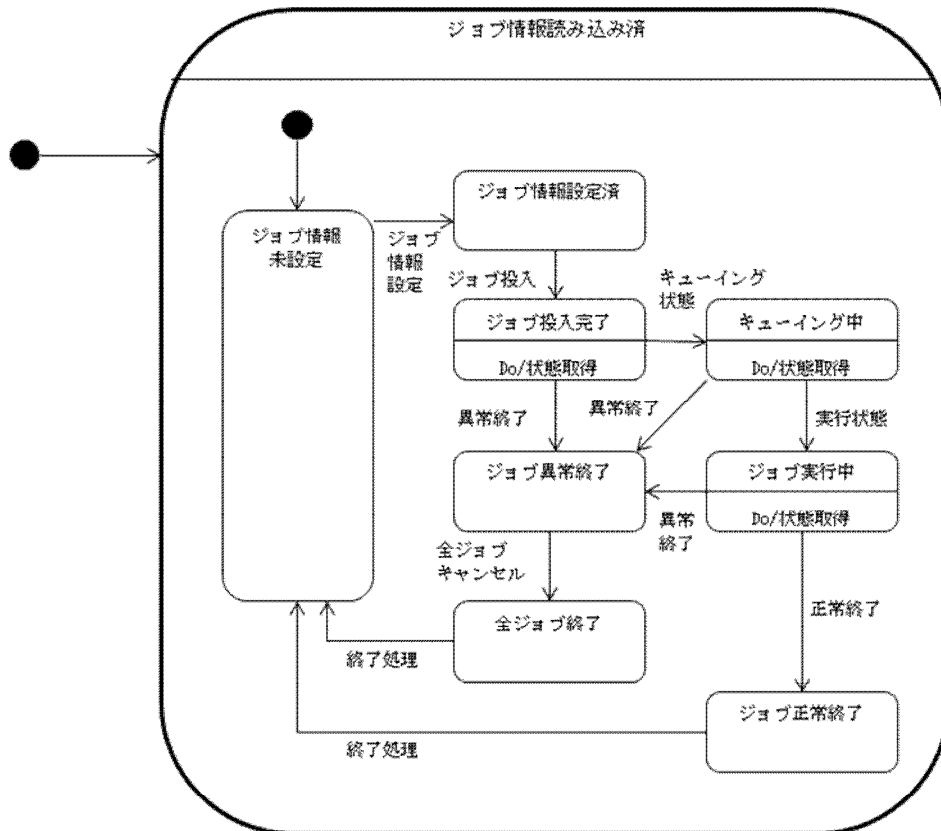


図 3 コントローラーの状態遷移

図 3 から分かるとおり、コントローラーは設定ファイルから読み込んだ情報を基に、ジョブの情報設定、投入、実行、終了までの処理を行う。ジョブ投入から終了まではジョブ状態を監視する。設定ファイルの情報不足している場合、あるいは誤りがある場合にはエラーとし、異常終了とみなして全ジョブを終了させる。ここで、シミュレーションコードのファイル生成、コード間のファイル転送を制御するアプリケーションとして、後述のセンチネルをシミュレーションコードとともにジョブとして実行し、センチネルの終了処理の後にシミュレーションコードのジョブ投入を行うよう設計することにより、利用者がコントローラーの設定ファイルを作成するだけで、シミュレーションコードを連携実行できるようにした。

設定ファイルは、図 4 に示すように、シミュレーションコード (PROGRAM) 及びファイルフロー (FLOW) の情報からなる。PROGRAM での "name", "path" はそれぞれ、シミュレーションコードの名前とパスを表す。FLOW での "type" は、シミュレーションコードのタイプを表す。一度だけ実行するコードを type 1, 他のコードからファイルを受け取って複数回

実行するコードを type 0 としている. FLOW の AtoB\_a, AtoB\_b はコード A の出力ファイルを表し, コード A の作業ディレクトリからコード B の作業ディレクトリに転送される. ファイル転送後, コード B が実行される. BtoA はコード B の出力ファイルで, コード B の実行後にコード B の作業ディレクトリからコード A の作業ディレクトリに転送される.

```

PROGRAMNUM 2
PROGRAM
  program CodeA
  name CodeA_0
  server computer1.node.site
  path /home/foo/bin/CodeA
  work /home/foo/workdir
  queue TSS
  para 1
  kind serial
END

PROGRAM
  program CodeB
  name CodeB_0
  server computer2.node.site
  path /home/foo/bin/CodeB
  work /home/foo/workdir
  queue TSS
  para 1
  kind serial
END

FLOWNUM 2
FLOW
  alias CodeA_0
  exec 1
  type 1
  send AtoB_a CodeB_0
  send AtoB_b CodeB_0
  recv BtoA CodeB_0
END

FLOW
  alias CodeB_0
  exec 1
  type 0
  recv AtoB_a CodeA_0
  recv AtoB_b CodeA_0
  send BtoA CodeA_0
END

```

図4 コントローラーの設定ファイルの例

### 3. 4 センチネル

ファイル転送及びジョブ投入のタイミングを定めるため, コントローラーはセンチネルを実行する. センチネルはシミュレーションコードの出力ファイルを検知し, シミュレーションコード間のファイル転送を制御するアプリケーションである. コントローラーによるジョブ投入の前後で, センチネルはその役割を果たす.

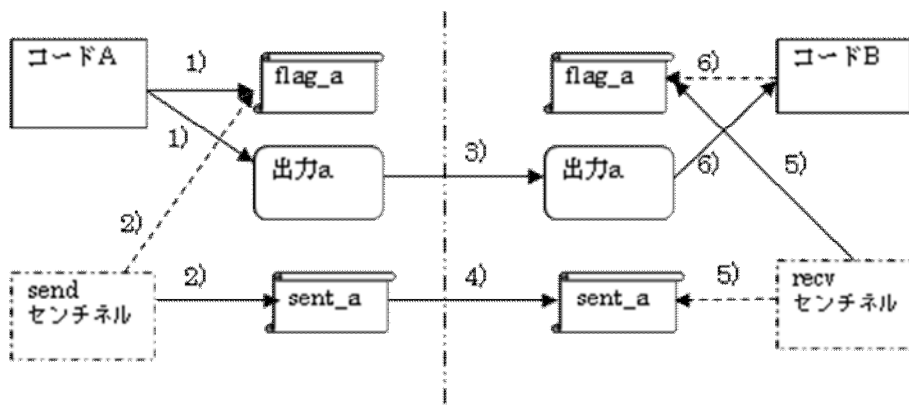


図5 センチネル実行の概略図

センチネルは send, recv, void の3種類のスクリプトから構成される. send スクリプトは出力ファイルの flag ファイルを検知し, sent ファイルを作成する. recv スクリプトは sent ファイルを検知し, flag ファイルを作成する. send, recv スクリプトはファイル

検知後終了する。その後、再び起動され、次のファイル検知に備える。void スクリプトはファイルを生成せず、最初に行われるシミュレーションコードとともに動作する。

センチネルの動作について説明する。例として、シミュレーションコード A とコード B があり、コード B はコード A の出力ファイルを入力として使うと想定する。動作順序は図 5 のようになる。番号が付与されている各過程では、以下の処理が実施される。

- 1) コード A が出力ファイルと、それに伴う flag ファイルを出力する。
- 2) コード A に伴う send センチネルが flag ファイルを検知し、削除。その後 send センチネルは出力ファイルの sent ファイルを作成し終了する。
- 3) コントローラーが出力ファイルをコード A の作業ディレクトリからコード B の作業ディレクトリに転送。
- 4) コントローラーが sent ファイルをコード A の作業ディレクトリからコード B の作業ディレクトリに転送。
- 5) コード B に伴う recv センチネルが sent ファイルを検知し、削除。その後 recv センチネルは出力ファイルの flag ファイルを作成し終了する。
- 6) コントローラーによりコード B が実行される。コード B は flag ファイルを検知し削除。コード A から転送されたファイルを入力ファイルとして使用する。

図 5 で示すように、コード A は出力ファイルに伴い flag ファイルを作成し、コード B は flag ファイルを検知してそれを削除する必要があるため、これらファイルの作成と削除のためにシミュレーションコードを変更することになるが、それ以外の変更は必要としない。

SOAF 使用による所要時間の増加を可能な限り低減するため、センチネルをマルチスレッドプロセスで動作するように開発した。type0 のコードに付随するセンチネルは逐次起動し、type1 のコードに付随するセンチネルは同時並行的に起動する。この結果、センチネルによるファイル検知、削除、作成に要する時間は 1 秒以内におさえることができた。SOAF 自身の所要時間については、具体例により 4 章で示す。

## **4. SOAF の評価**

本章では、SOAF の動作試験を行い、その有用性を評価した結果を示す。なお、本論文での SOAF の評価に際しては、実際のコードではなくダミーコードを用いた。ダミーコードは、連続動作してファイルと flag ファイルを生成し続けるものと、flag ファイルを検知して新たな出力ファイルと flag ファイルを生成するものから成る。前者が type1、後者が type0 のコードに対応する。出力ファイルの大きさは数十バイト、flag ファイルの大きさは 0 バイトである。type1 のコードは 10 ステップ動作し、ファイル生成の間隔は 10 秒、また各々のファイル送受信の待ち時間は最大 10 秒とした。

テストは原子力機構システム計算科学センターの上野サイトで実施した。クライアント PC には LinuxOS がインストールされた PC を用いた。ジョブを投入する計算機には IBM pSeries690 を用いた。AEGIS への認証には、IC カードを用いた。

我々は SOAF を用いて、三つのシナリオの統合シミュレーションを実行した。三つのシナリオとは、1 章に記述した 3 次元仮想振動台、地震耐力予測シミュレーションシステム、核融合プラズマシミュレーションシステムである。以下、それぞれについて詳細を述べる。

#### 4. 1 3次元仮想振動台

3次元仮想振動台の実現に向け、構造解析に用いるコードとして FInite Element SStructural analysis for Assembly (FIESTA)が開発されている[17]. FIESTA は構造物を部品単位で処理し、部品間の相互作用を考慮することで原子力施設の全体解析を行う. 組立構造解析手法が考案され、複雑な大規模構造物の静・動弾性解析が可能となっている. 3次元仮想振動台では、現在、原子力機構大洗研究開発センターに設置された高温工学試験研究炉 High Temperature engineering Test Reactor (HTTR)[18]を対象とした解析を実施している. HTTR は、核分裂で生じた熱を外に取り出すために、冷却材としてヘリウムガスを用いた新型の原子炉である. 発電以外にも水の熱化学分解による水素の製造など、化学工業の様々な分野での応用が見込まれている.

本シミュレーションでは、HTTR は、高温ガス炉 (RPV), 加圧水冷却器 (PWAC), 補助冷却器 (AWAC)の三つの大型機器、及びこれらをつなぐ三つのパイプから構成されるとし、まず大型機器での弾性解析を実施し、その結果として得られる境界条件をパイプのシミュレーションに随時送信し、パイプの弾性解析を行っている. 大型機器とパイプの関係は図6のようになっている.



図6 3次元仮想振動台の構成

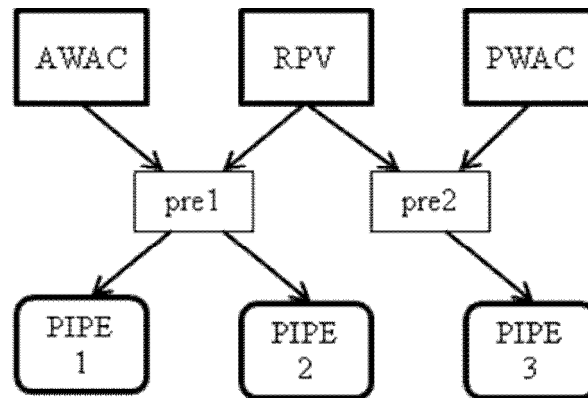


図7 3次元仮想振動台のファイルフロー

仮想振動台のファイルフローは図7のようになる. シミュレーション中に AWAC, RPV, PWAC を解析するコードから三つのパイプを解析するコードにファイルが随時転送される. これら六つのシミュレーションはそれぞれ FIESTA により実行される. また、六つの FIESTA のほかに、大型機器からパイプへの境界条件の引き渡しを行う前処理のコード pre が二つ動作する. PIPE1, 2 への pre1 と、PIPE3 への pre2 である. ここで六つの FIESTA はすべて type 1 に、pre は type0 に分類される.

動作ダイアグラムは図8のようになる. パイプライン型の連携で、RPV を解析するコード、pre1, pre2 からは同じファイルが複数のコードに転送される. AWAC, RPV, PWAC を解析するコードからはファイルが10個ずつ生成される.

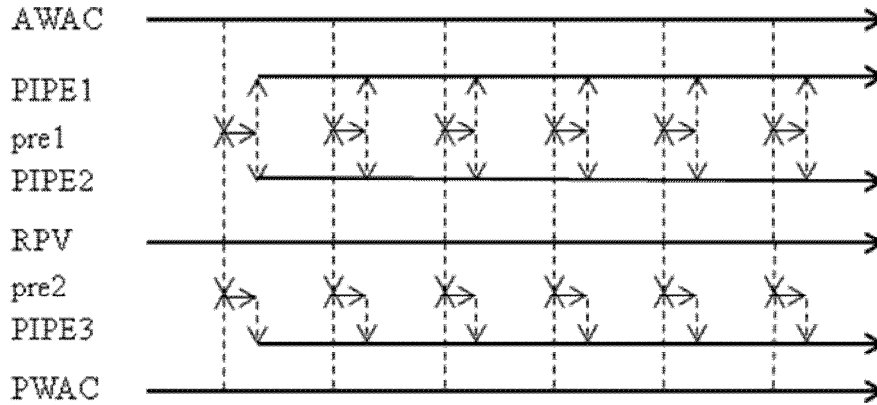


図8 3次元仮想振動台の動作ダイアグラム。

FIESTA の代わりにダミーコードを用いてテストした結果，全動作時間は約8分であった．ファイルの送信元，発信元の対応が1対1ではなく，なおかつ同じファイルが複数のコードに転送されるやや複雑な状況であるが，ファイルフローのみの記述でシミュレーションは連携動作した．SOAF 使用による所要時間の増加は約3分であった．

#### 4.2 地震耐力予測シミュレーションシステム

地震耐力予測シミュレーションシステムに用いられるシミュレーションコードはそれぞれ，東京大学地震研究所，東京大学大学院システム量子工学専攻吉村研究室，原子力機構東海サイトに設置された計算機上に実装されている．それぞれのシミュレーションコードは，地盤解析を行う Macro-micro Analysis Method (MMA) [19]，建屋・プラント・機器解析を行う ADVANCED ENGINEERING analysis Tool for Ultra large REal world (ADVENTURE) [20]，機器内の核熱解析を行う BWR 3D-Neutronic Thermal-Hydraulic Code (TRAC-SKETCH) [21]，熱流動解析を行う Advanced Code for Evaluation of 3-Dimensional constitutive equation in two-fluid model (ACE-3D) [22]である．これらのコード間で入出力データを転送し，ファイルベースでの連携を実現する．

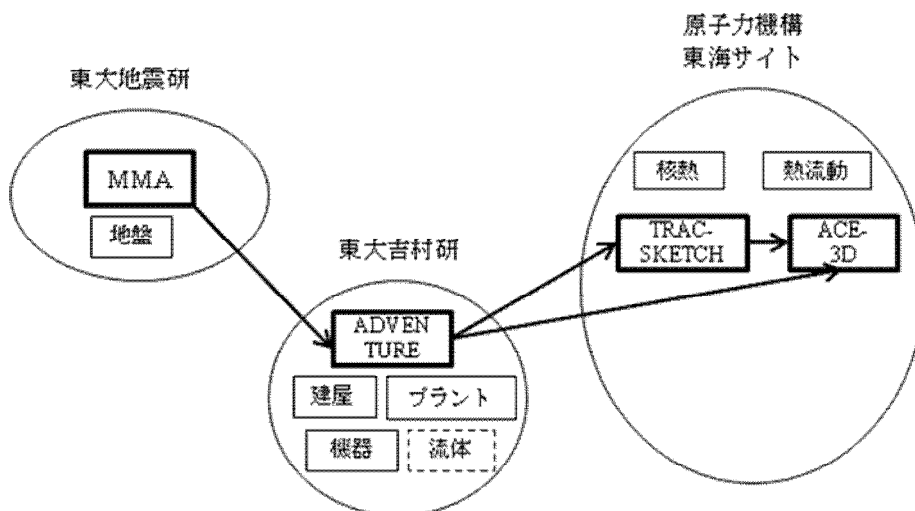


図9 耐力解析システムのファイルフロー

地震耐力予測シミュレーションシステムのファイルフローを図9に示す。MMA の出力ファイルが ADVENTURE に転送され、ADVENTURE の出力ファイルが TRAC-SKETCH と ACE-3D に転送される。ACE-3D は ADVENTURE, TRAC-SKETCH の両方からの出力ファイルを受け取る。以上のようにパイプライン型の連携で逐次解析を実施していく。

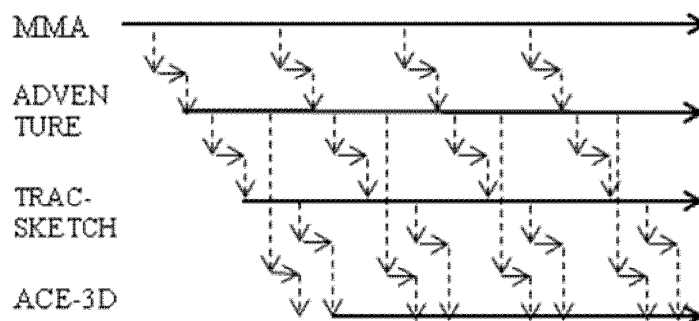


図10 耐力解析システムの動作ダイアグラム。前処理のコード名は省略した。

動作ダイアグラムは図10のようになる。本シナリオのテストでは、四つのコード及びコード間のファイルの前処理を行う四つのコードの計八つのコードを組み合わせ実施した。MMA, ADVENTURE, TRAC-SKETCH, ACE-3D は type1, ファイルの前処理を行うコードは type0 である。MMA から 10 回ファイルが出力され、このファイルを他のコードが逐次受信して新たなファイルを生成し、次のコードに引き渡すという処理を行った。

テストの結果、全動作時間は約 11 分であった。MMA から ACE-3D まで最大で七つのコードを経由しているが、SOAF ではファイル生成後に逐次転送を実施しており、本テストでファイルフローの設定だけでパイプライン処理が成功していることが分かる。また、SOAF 使用による所要時間の増加は約 2 分であった。

#### 4.3 核融合プラズマシミュレーションシステム

核融合プラズマシミュレーションシステムで用いられるシミュレーションコードは、トカマク内のプラズマの時間発展を予想する Tokamak Prediction and Interpretation Code System (TOPICS) [23], 2次元の磁気安定性解析を行う two-dimensional magnetic stability analysis code (MARG2D) [24], そしてプラズマを電磁波で制御することを仮想的に行う Electron-Cyclotron Current Drive code (ECCD) [25] と Lower-Hybrid Current Drive code (LHCD) である。

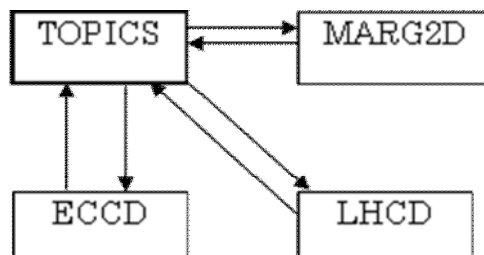


図11 核融合プラズマシミュレーションのファイルフロー

TOPICS を最初に起動して、TOPICS の出力ファイルを MARG2D, ECCD, LHCD の入力ファイル

ルとして用いることでプラズマの挙動，安定性が解析される． TOPICS 以外の三つのコードは，シミュレーションの状況次第で実行されるかどうかが決まるので，このシミュレーションの連携は，2章に記述した「動的な条件分岐型」に分類される．ファイルフローを図11に示す．また，動作ダイアグラムの例を図12に示す．

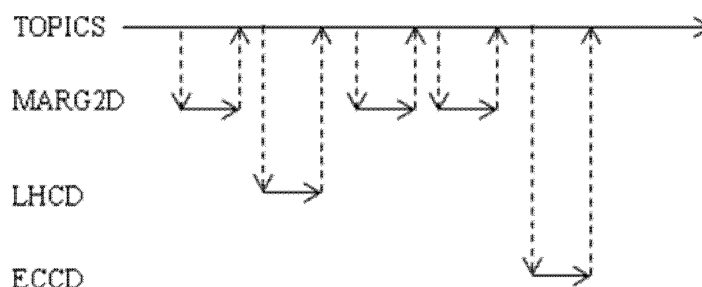


図12 核融合プラズマシミュレーションの動作ダイアグラムの例

本シナリオのテストでは TOPICS が 10 回ファイルを生成し，その都度 MARG2D が送受信を行う． ECCD は 2 回に 1 度で計 5 回， LHCD は 5 回に 1 度で計 2 回のファイル送受信を行う． TOPICS は type1，他の三つのコードは type0 に分類される．

テストの結果，全動作時間は約 16 分であった． TOPICS が生成したファイルをもとに， TOPICS 以外のコードがファイル受信，実行，ファイル返信を実施する連携動作が，設定ファイルの記述だけで実現できた．つまり，ファイルフローの記述だけで，動的な条件分岐を伴うシミュレーションの連携実行が実現できた．本テストでは TOPICS 以外のコードが実行しているときは， TOPICS は停止する．他のコードの実行，及びファイル送受信の際の待ち時間のため，本シナリオは三つのシナリオで一番ファイル送受信が少ないにもかかわらず，他のシナリオよりも全動作時間が長くかかった．しかしながら，実際に想定しているシミュレーションでは，各々のコードの実行時間は 10 秒よりもはるかに長いいため， SOAF は核融合プラズマシミュレーションの連携実行に対し十分に実用的であると考えられる．

## 5. 今後の課題

SOAF の開発により，グリッド分散された計算機に実装されたシミュレーションコードを，科学者の手を煩わせることなく連携動作させることを可能とした．想定している三つのシナリオについて連携動作が可能であり，十分に実用的であることを，ダミーコードを用いて示した．

今後は， SOAF を用いて，統合シミュレーションの長時間実行を実現する．ダミーコードは 10 秒間隔でファイルを生成し数分で終了するが，実際のシミュレーションコードは，数時間，あるいはそれ以上にわたって実行される．例えば，国際熱核融合炉実験計画 ITER では， 1000 秒以上のプラズマ燃焼が考えられている [26] ため，この計画を反映する核融合プラズマシミュレーションを実施しようとする，数週間あるいは数カ月を要する．更に，ダミーコードの出力ファイルは数十バイトであるが，実際のコードはより大容量のデータを出力する．例えば，3次元仮想振動台の FIESTA が出力するデータの量は数百メガバイトに及ぶ．

現在，長時間実行への対応のために， SOAF へのフォールトトレラント機能の組み込みに着手している．大型計算機では一般にジョブの実行時間に制限がつけられている．このた



め、長時間のシミュレーションをそのまま実行すると、実行時間制限のために途中で打ち切られてしまう。他にも、計算機環境の動作不良によりシミュレーションが中断されてしまうこともある。シミュレーションが中断した場合、途中からシミュレーションを再開するための、ジョブの再投入機能やファイルの再転送機能などが必要になる。現在、これらの機能の SOAF への実装に取り組んでいる。この実現により、SOAF の適用範囲がますます広がることを期待される。

## **6. おわりに**

我々は、分散した計算機に実装されたシミュレーションコードを連携して実行することができるフレームワークSOAFを開発した。利用者は、シミュレーションコードの所在とファイルフローを記述した設定ファイルを使って、容易にシミュレーションの連携実行ができる。また、SOAFはシミュレーションコードに大規模な修正を加えることを必要とせず、シミュレーションを連携動作するように制御できる。もし連携の仕方が変わった場合には、設定ファイルを書き換えるだけで対応できる。

SOAF開発に際し、我々は4章で述べたように四つの問題について考えた。SOAFを用いると、三つのシナリオでシミュレーションの連携実行が容易にできることを説明した。これにより、第1の問題は解決した。また、シミュレーションの連携実行においては、シミュレーションコードにflagファイルの生成と検出機能を備えさえすればよく、利用者はコントローラーの設定ファイルを作成するだけでよいことを述べた。このため、第2、第3の問題を解決した。最後に、実際のシミュレーションの連携実行を想定すると、SOAFだけが動作している時間は全シミュレーション時間に比べると些細なものであることが分かった。これにより第4の問題を解決した。以上により、開発時に考慮した点はSOAFで解決されたと考えられる。

本論文の4.3章で示したように、シミュレーションの途中で生成される出力ファイルに応じて、新たに起動されるシミュレーションコードが決まるという、動的な条件分岐シナリオにもSOAFは対応している。この連携は、従来のワークフロー型の連携実行では実現が困難であった。動的な条件分岐でもコードに大がかりな変更は不必要で、ファイルフローを記述しさえすれば実現できる。グリッド対応のRPCやMPIの実装に比べて利用者の負担を大きく減らすことができるため、様々な分野で多様なシミュレーション手法の統合が容易にできると期待される。

## **謝辞**

本研究の一部は、(独)科学技術振興機構 戦略的創造推進事業 CRESTの研究課題「原子力発電プラントの地震耐力予測シミュレーション」の予算を使用して行いました。研究の遂行に関しまして、原子力機構 システム計算科学センター 高度計算機技術開発室 耐震計算科学技術開発チーム、原子力機構 大洗研究開発センター 高温工学試験研究炉部、原子力機構 那珂核融合研究所 核融合研究開発部門にご協力をいただきました。ここに記して感謝の意を表します。

## **参考文献**

[1] 西田明美, 松原仁, 田栄, 羽間収, 鈴木喜雄, 新谷文将, 中島憲宏, 谷正之, 近藤誠,

「原子カプラントのための3次元仮想振動台の構築 組立構造解析法による巨大施設解析システムの提案」, 日本原子力学会和文論文誌, Vol. 6, No. 3, p. 376-382 (2007)

- [ 2 ] S. Yoshimura, "Overview of Adventure Project", Technical Report of ADVENTURE Project, ADV-99-1 (1999)
- [ 3 ] N. Hayashi, T. Takizawa, T. Ozeki, N. Aiba, and N. Oyama, "Integrated Simulation of ELM Energy Loss Determined by Pedestal MHD and SOL Transport", Nucl. Fusion, vol. 47, pp. 682-688, (2007)
- [ 4 ] 伊達進, 奥村利幸, 秋山豊和, 下條真司, 松田秀雄, 中村春木, 「バイオグリッドプロジェクト「スーパーコンピュータネットワークの構築」」, 情報処理, vol.44(6), pp.601-607 (2003)
- [ 5 ] 武宮博, 田中良夫, 中田秀基, 関口智嗣, 「MPIとGridRPCを利用した大規模Gridアプリケーションの開発と実行:Hybrid QM/MDシミュレーション」, 情報処理学会誌: コンピューティングシステム, vol. 46, No. SIG 12(ACS 11), pp.384-395 (2005)
- [ 6 ] 福田正大, 「ITBL: 新しい研究環境の構築に向けて」, 学術月報, vol. 55(2), pp.19-23 (2002)
- [ 7 ] T. Imamura, Y. Hasegawa, N. Yamagishi, and H. Takemiya, "TME: A Distributed resource handling tool.", Recent Advances in Computational Science & Engineering, International Conference on Scientific & Engineering Computation (IC-SEC) (3-5 December 2002, Raffles City Convention Centre, Singapore), pp. 789-792, (2002)
- [ 8 ] D. Erwin (Ed.), "UNICORE Plus final Report - Uniform Interface to Computing Resources", Forschungszentrum Jülich, (2003)
- [ 9 ] I. Foster, "Globus Toolkit Version 4: Software for Service-Oriented Systems", IFIP International Conference on Network and Parallel Computing, Springer-Verlag LNCS 3779, pp 2-13, (2006)
- [ 1 0 ] B. Ludäscher, I. Altinta, C. Berkley, D. Higgins, E. Jaeger, M. Jone, E. A. Lee, J. Tao, Y. Zhao, "Scientific workflow management and the Kepler system", Concurrency and Computation: Practice and Experience, vol.18, pp. 1039-1065, (2005)
- [ 1 1 ] K. Seymour, H. Nakada, S. Matsuoka, J. Dongarra, C. Lee, H. Casanova, "Overview of GridRPC: A Remote Procedure Call API for Grid Computing", Proceedings of 3rd International Workshop on Grid Computing (M. Parashar eds.), pp 274-278, (2002)
- [ 1 2 ] T. Imamura, Y. Tsujita, H. Koide, and H. Takemiya, "An Architecture of Stampi: MPI Library on a Cluster of Parallel Computers", Proceedings of the 7th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, pp. 200-207, (January 2000)
- [ 1 3 ] I. Foster and N. T. Karonis, "A grid-enabled MPI: message passing in heterogeneous distributed computing systems", Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM), pp.1-11, (1998)

- [14] E. Gabriel, M. Resch, T. Beisel, and R. Keller, "Distributed Computing in Heterogeneous Computing Environment", Proceedings of the 5th European PVM/MPI Users' Group Meeting on Recent Advances in Parallel Virtual Machine and Message Passing Interface, pp.180-187, (September 1998)
- [15] Y. Suzuki, K. Nakajima, N. Kushida, C. Kino, T. Minami, N. Matsumoto, T. Aoyagi, N. Nakajima, K. Iba, N. Hayashi, T. Ozeki, T. Totsuka, H. Nakanishi and Y. Nagayama, "Research and Development of Fusion Grid Infrastructure Based on Atomic Energy Grid InfraStructure (AEGIS)," Sixth IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research (4-8 June 2007, Inuyama, Japan), Fusion Engineering and Design Vol. 83, pp.511-515 (2008)
- [16] 鈴木喜雄, 「GRID環境利用のためのアプリケーションプログラミングインタフェースの研究開発」, 2007年度FUJITSUファミリ会論文 (2007)
- [17] Y. Suzuki, A. Nishida, F. Araya, N. Kushida, T. Akutsu, N. Teshima, K. Nakajima, M. Kondo, S. Hayashi, T. Aoyagi, and N. Nakajima, "Development of Three-dimensional Virtual Plant Vibration Simulator on Grid Computing Environment ITBL-IS/AEGIS", Journal of Power and Energy Systems, vol. 3, No. 1, pp.60-71 (2009)
- [18] S. Saito et al., "Design of high temperature Engineering Test Reactor (HTTR)", Report of the Japan Atomic Energy Research Institute, JAERI 1332 (1994).
- [19] T. Ichimura and M. Hori, "Macro-Micro Analysis Method for Strong Motion Distribution", J. Struct. Eng./Earthquake Eng., JSCE, I-52:654, 51-62 (2000).
- [20] 山崎一郎, 吉村忍, 「有限要素解析ソフトの利用環境整備－ADVENTUREシステム利用手引書－」, JAERI-Tech 2003-050 (2003)
- [21] V. G. Zimin, H. Asaka, Y. Anoda et al., "Verification of J-TRAC code with 3D neutron kinetics model SKETCH-N for rod ejection analysis", NURETH-9, San Francisco, California (1999)
- [22] 大貫晃, 加茂英樹, 秋本肇, 「多次元二流体モデル構成方程式評価用コード ACE-3Dの開発」, JAERI-Data/Code 96-033 (1996)
- [23] N. Hayashi, A. Isayama, K. Nagasaki, and T. Ozeki, "Numerical Analysis of Neoclassical Tearing Mode Stabilization by Electron Cyclotron Current Drive", J. Plasma Fusion Res., vol. 80, pp.605-613, (2004)
- [24] S. Tokuda and T. Watanabe, "A new eigenvalue problem associated with the two-dimensional Newcomb equation without continuous spectra", Phys. of Plasmas, vol. 6, pp.3012-3026, (1999)
- [25] K. Hamamatsu and A. Fukuyama, "Controllability of driven current profile in ECCD on ITER", Fusion Eng. Des., vol. 53, pp.53-58, (2001)
- [26] ITER Physics Basis Editors et al., "Chapter 1: Overview and summary", Nucl. Fusion, vol. 39, pp.2137-2174, (1999)