

研究解説

HPC ミドルウェア

HPC Middleware

奥田 洋司*・中島 研吾**

Hiroshi OKUDA and Kengo NAKAJIMA

1. 背景

ハードウェア技術の進歩によって並列計算機、ベクトル計算機など様々な高性能の計算機が開発されている。また MPI に代表される並列計算用メッセージパッシングライブラリ、ワークステーションや PC のクラスタによって並列計算処理システムを容易に実現可能となっている。

非構造格子を使用した有限要素法および有限体積法などの科学技術シミュレーションの分野では大規模計算、高速計算への需要が高い。並列計算機、ベクトル計算機を利用する場合には計算機の性能を最大限引き出せるようなプログラミングが必要となる。従って、それぞれの計算機に対して異なったコードを開発する必要があるため：

- ・開発生産性の低下
- ・信頼性の低下
- ・可搬性、移植性の欠如
- ・保守の困難さ

を生じている。シミュレーションコード開発者（以下「コード開発者」）は計算機に依存した様々なルールを修得する必要があるが、この中には計算本体とは必ずしも直接関係のないものもある。このような現状は、科学技術シミュレーション分野におけるハイエンド計算技術の普及を阻害し、さらには複雑問題における新規アルゴリズムの開発などコード開発者の本来の専門分野の発展の停滞を招いている。

したがって、科学技術シミュレーションコード開発にあたって、並列処理、計算機依存チューニングなど計算本体と必ずしも直接関係のないプロセスを隠蔽するような仕組みが必要である。

一方、近年、広帯域ネットワーク技術の発展、インフラストラクチャの整備により、「GRID」「ITBL」など広域に分散したさまざまなリソース（計算機、ソフトウェア、デ

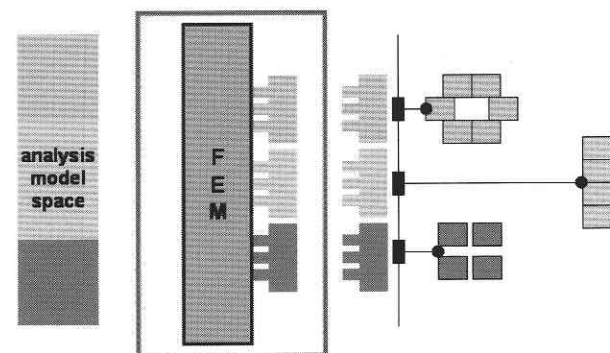


図1 Concept of 'Virtual Computer'
Parallel computers are connected on WAN via network OS.

ータ) をネットワーク経由で利用しようという試みが行なわれている。

そのなかの1つにバージニア大学で開発された「Legion」、 「AVAKI HPC」がある。これはネットワーク OS であり、特に後者は HPC 計算の効率的実施に主眼が置かれ、

- ・保守の困難さアーキテクチャの異なる計算機を連携
- ・グローバルネットワーク空間における命名機能
- ・高度なセキュリティ機能
- ・セキュアな大規模ファイルの共有
- ・ネットワークにおけるプロセス制御

などの特徴がある。このようなネットワーク OS を使用することにより複数の計算機リソースを一つの大きな仮想計算機として取り扱うことが可能となる（図1）。なお、ネットワーク OS については、その発展とともに globus への仕様準拠の動きがある。

このような「仮想計算機環境」を使用すれば、流体解析、構造解析および連成解析における大規模シミュレーション、詳細リアルタイムシミュレーションなどが可能となる。

*東京大学大学院工学系研究科

** (財) 高度情報科学技術研究機構

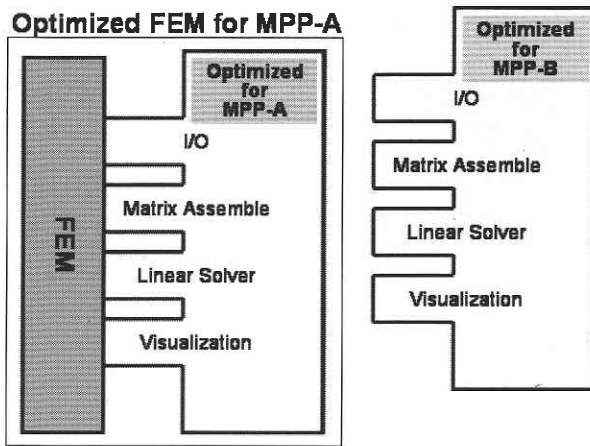


図2 Utilization of 'HPC-MW'
Optimized FE code is generated by plugging HPC-MW.

2. 本プロジェクトの目的

有限要素法 (FEM), 有限体積法 (FVM) など, 非構造格子を使用した科学技術シミュレーションにおける計算手法の計算処理パターン, それらの連成に必要な共通インタフェースを抽出し, ハードウェアに依存しない開発基盤, すなわち HPC ミドルウェア (HPC-MW) の機能設計, 開発, 実装を実施する. PC で開発されたソースプログラム (FORTRAN 90 または C 言語で記述) をネットワーク上の各ハイエンド計算機にインストールされた HPC-MW にプラグインすることにより, PC クラスタからベクトル並列計算機まで, それぞれのハードウェアに対して最適化されたコードが自動的に生成される. 図2にこの概念を示す.

アプリケーションプログラムのステップ数は 1000 程度, 開発工程は従来の 10 分の 1 となり, 移植性に富んだ信頼性の高いプログラムを効率的に開発できる. アプリケーションレベルで, 「地球シミュレータ」上で 5 TFLOPS の性能を達成することを目標とする.

コンパイラ型アプローチを一部採用し, 新しい計算機アーキテクチャへの柔軟な対応が可能である. 更に, 高速ネットワークで接続された計算機群による仮想計算機環境を構築し, HPC-MW 使用によるマルチスケール連成モデルを使用した大規模シミュレーション (流体解析, 構造解析および連成解析) を実施する.

すなわち, 本プロジェクトでは, 以下の機能を有するソフトウェア群を開発する:

- ・並列計算機に代表されるハイエンド計算機環境において, 各ハードウェアにおいて最適化された有限要素法による大規模シミュレーションコードの開発を円滑に行うための開発基盤, すなわち「HPC ミドルウェア (HPC-MW)」を提供する.
- ・複雑形状または新規計算機アーキテクチャに最適化さ

- れた「HPC-MW」を自動的に生成する機能を提供する.
- ・「GRID」などネットワークで接続された, ヘテロジニアスな計算機リソースを使用してシミュレーションを実施する際に, 最適化されたシミュレーションコードを開発するための「HPC-MW」を自動的に生成する機能を提供する.
- ・大規模有限要素法シミュレーションで必要となる, 大規模メッシュを生成し, 領域分割するための機能を提供する.
- ・「HPC-MW」の性能を検証するための機能を提供する.

3. 研究開発の手順

まず有限要素法 (FEM), 有限体積法 (FVM) など, 非構造格子を使用した科学技術シミュレーションにおける計算手法について, 特徴的なプロセスを抽出および体系化し, それによって得られる標準的処理パターンに基づき, 各々使用計算機に依存しない形式によるプログラムの記述方法, すなわち, ハードウェア依存部分隠蔽問題記述仕様およびデータ構造を定める. また複数プログラム連成のためのインタフェースを定める.

続いてその方法を用いて記述されたプログラムを高効率に実行するための HPC ミドルウェア (HPC-MW) をターゲット計算機環境上に構築する. ターゲットとする代表的な計算機環境は, プロセッサとしてはスカラーまたはベクトルプロセッサ, アーキテクチャとしては分散メモリまたは SMP クラスタ並列型の組み合わせによる計 4 種類を考慮する. 対象開発言語は FORTRAN 90 および C とする.

このほか, ベタフロップス計算機に代表される次世代アーキテクチャに関する技術動向にも注目し, 拡張性に配慮した HPC-MW を開発する. 問題記述の方法として「コンパイラ型」を採用することによってこのような対応が可能となる. また「GRID」「ITBL」に代表される広域分散処理環境を使用して, 高速ネットワーク接続された複数の並列計算機資源を使用する場合の最適化に関しても考慮した「ネットワーク型」の HPC-MW も併せて開発する.

さらに, 本研究で整備された HPC-MW を, 構造解析, 流体解析, および連成解析の先端的シミュレーションコードの開発に適用し, HPC-MW の援用による開發生産性, 保守性, 信頼性, 可搬性, 移植性を評価・検証する.

HPC-MW の改良のため, この評価結果をさらにシミュレーションの計算処理パターンの抽出・体系化, HPC-MW の開発にフィードバックする. また, 本研究で開発される HPC ミドルウェアを公開し, 広い分野での効果とニーズを分析するものとする.

4. 国内外の研究の現状

開発システムの詳細を記述する前に, 関連分野における

国内外の研究動向について述べる。

ハイエンド計算機環境における科学技術計算コードがハードウェアごとに個別に最適化されており、計算処理パターンを標準化してアプリケーションを多様な環境に適應させかつ普及させる試み、上位レベルのミドルウェアを設定してハイエンド計算機環境における先端的シミュレーションの普及を図る試みはない。また、有限要素法 (FEM) に関してはこれまでも、色々な試みが行なわれているが、PC クラスタからベクトル並列計算機まで、広範囲のハイエンド計算機環境を統一的なインタフェースでカバーしようという試みはない。

アメリカにおいて「PITAC (President Information Technology Advisory Committee)」の提言を受けて 2000 年頃から開始された「IT² (Information Technology for the 21st Century)」では、ソフトウェアの生産性、信頼性、安定性を高めるための基盤整備が重視されている。具体的には：

- ・大規模複雑ソフトウェアの設計、実証試験、保守等の基準を確立する
- ・エンドユーザーが手軽にプログラミングできるような環境を開発する
- ・コンポーネントを利用したソフトウェア開発手法を促進し、開發生産性、信頼性を高める
- ・超並列計算機などハイエンド計算機の効率を最大限引き出せるプログラムを容易に開発できるような開発基盤を整備する

などがあげられており、本研究の趣旨ともよく合致している。アメリカにおいても本研究が目指すような動きはまだ緒に就いたばかりであるといえる。

それら類似研究のなかに、「ASCI (Accelerated Strategic Computing Initiative)」プロジェクトの中の「DOE Distributed Computing Research Team」, 「ALICE (Advanced Large-Scale Integrated Computational Environment)」(Argonne National Laboratory, ANL) などが挙げられる。しかしながら、これらの多くは線形方程式ソルバーなど一部の機能に限定されている。また、多くは計算機科学サイドからの研究であり、シミュレーション技術者、研究者の立場からは必ずしも使いやすいものになっていない。

「CORBA (Common Object Request Broker Architecture)」に代表される「ORB (Object Request Broker)」は空間的に分散している計算機群上のオブジェクト間の通信サポートを実施するためのソフトウェアシステムである。個々のアプリケーション (オブジェクト) の並列化、最適化とは無関係であり、またアプリケーションの結合、連成という観点からも下位レベルの (ハードウェア、OS に近い) ミドルウェアと考えられる。「Ninf」(産業技術総合研究所), 「Globus」(ANL), 「Legion」(University of Virginia) はともに広域分散プログラムを記述、実行するための開発基盤

である。Ninf は RPC (Remote Procedure Call) ベース、Globus はメッセージパッシングベースの通信機能、および、ファイルアクセスなどをサポートしている。これらについても、CORBA と同様、より下位レベルに位置付けされるミドルウェアである。「Legion」は広域分散環境上の計算機ハードウェア群を統合する OS であり、本研究で利用することも考えられる。

また、ネットワーク OS を使用して、広域分散環境上の計算機群を仮想的な計算機環境として扱った事例はあるが、異なったアーキテクチャを持つそれぞれの計算機に対してアプリケーションレベルで最適化された例はない。

5. 本プロジェクトで開発する HPC-MW の優位性

本プロジェクトで開発する HPC-MW は、多様なハイエンド計算機環境において標準化された計算処理パターンのもとにハードウェア依存部分を隠蔽する世界初のミドルウェアであり、また、1,000 ステップコード、開発工程 10 分の 1 のソフトウェア開発環境、「地球シミュレータ (ピーク性能 40 TFLOPS)」においてアプリケーションレベルで 5 TFLOPS 以上の性能、を目指している。

本研究のアプローチのユニークさは、共通化できる一部の計算部分をライブラリ化する従来のアプローチとは異なり、解析アルゴリズムに依存する部分以外をミドルウェアがカバーしている点である。これによって従来の数値ライブラリとは異なり隠蔽可能な部分が増加し、コード開発者によるコーディング量を従来と比較して減少させ、コードの保守性、信頼性、可搬性、移植性を格段に向上させることが期待される。またインタフェースをよりアプリケーションに近い部分においているため、利用者にとって非常に使いやすいものである。

6. 設計指針

以下の 4 項目の観点について、本システムの基本的な設計戦略を示す。

- ①処理パターン抽出および体系化
- ②ハードウェア依存部分隠蔽問題記述仕様策定
- ③各種ハードウェア環境における HPC-MW 実装
- ④ HPC-MW を使用した大規模シミュレーション

①処理パターン抽出および体系化

非構造格子を使用した連続体手法 (有限要素法, 有限体積法) に代表される科学技術計算手法およびそれらの連成計算における標準的な計算処理パターンを抽出し、研究項目②における HPC-MW 基本設計のための必要項目を決定する。さらにこれら抽出された計算処理パターンを体系化し、他の手法による様々な種類のアプリケーションに対しても本課題の成果を容易に転用できるようにする。

有限要素法に関してはその特徴である要素単位処理の特性を考慮して研究を実施する。また、要素の次数、混合補間法、安定化手法、適応格子、動的負荷分散、マルチレベル解法、連立一次方程式解法などの項目にも注目して検討を実施する。

先端的シミュレーションに必須のマルチフィジックス／マルチスケール・シミュレーションに関連した連成問題に関しては、複数のアプリケーション間でデータ交換を行うための標準的な手続、様々な空間離散化手法および時間離散化手法に基づいたデータ変換方式および複数プログラムの実行スケジューリングなどの典型例を導き出し、HPC-MW 開発のために必要な標準的な計算処理パターンの抽出および分類を実施する。各手法間の相互作用のほか、同じ手法を使用している対象とするモデル、領域、物理現象が異なるような複数アプリケーション間の相互作用についても考慮する。

②ハードウェア依存部分隠蔽問題記述仕様策定

計算アルゴリズムとデータは密接に結びついており、HPC-MW の効率を高めるためにはデータ構造に配慮することが必要となる。有限要素法、有限体積法およびそれらの連成について、①で得られた知見に基づき、ハイエンド計算機の性能を最大限引き出すことが可能となるように、スカラー／ベクトルプロセッサ、分散メモリ／SMP クラスタ並列型などのハードウェアの種類に依存しないデータ構造を検討する。対象開発言語は FORTRAN 90 および C とする。

並列計算機を使用して大規模な問題を計算する場合、局所化された大量のデータセットに関する入出力を実施する必要がある。開発するアプリケーションソフトウェアの信頼性、可搬性、移植性を高めるにはこのような計算本体に直接関係の無い部分の処理は隠蔽することが必要である。有限要素法、有限体積法およびそれらの連成について入出力に関する標準的処理パターンを抽出する。さらに HPC-MW のプロトタイプを開発する。データ入出力も含めて以下のプロセスに関して注目するものとする：

- ・データ入出力
- ・領域分割
- ・適応格子
- ・動的負荷分散
- ・並列可視化
- ・線形ソルバ
- ・有限要素処理（マトリクス計算、コネクティビティ生成等）
- ・連成カップリング

線形ソルバとしては有限要素法に特有な疎行列ソルバのほか、第一原理分子動力学に現れる、密行列固有値ソルバについても検討するものとする。

また、ペタフロップスに代表される次世代アーキテクチャに関する技術動向にも拡張可能な仕様とする。問題記述の方法としては既存言語をそのまま使用する場合のほか、新規記述言語開発、プラグマ記述について検討する。

③各種ハードウェア環境における HPC-MW 実装

研究項目②で決定された問題記述仕様に基づき、HPC-MW を代表的計算機環境に構築、実装、最適化を実施する。②における問題記述仕様、データ構造設定に関するフィードバックも実施する。

ターゲットとする代表的な計算機環境は、プロセッサとしてはスカラーまたはベクトルプロセッサ、アーキテクチャとしては分散メモリまたは SMP クラスタ並列型の組み合わせによる計 4 種類を考慮する。「GRID」「ITBL」に代表される広域分散処理環境を使用して、複数の並列計算機資源を使用する場合の最適化に関して検討を実施する。

本研究開発では、(1) ライブラリ型、(2) コンパイラ型および (3) ネットワーク型、の 3 種類の HPC-MW を開発する。ライブラリ型は既存の特定のハードウェアに対して最適化されたライブラリとして提供されるものである。コンパイラ型は次世代ハードウェアや新規開発のハードウェアに関して、演算性能、通信性能、SMP ユニット当りのプロセッサ数などのパラメータを入力することにより、それぞれのハードウェアに関して最適化されたライブラリを自動的に生成する。ネットワーク型は「Legion」, 「Globus」などの既存のフレームワーク (Grid Middleware) を使用してネットワーク接続された並列計算機群 (異機種の場合も含む) を対象としたものである。基本的にライブラリ型ミドルウェアをネットワーク上の各並列計算機にインストールして使用する。

また、開発した HPC-MW の信頼性向上、効率的な保守を実現するため、効果的な開発環境、コード記述手法についても検討するものとする。

④ HPC-MW を使用した大規模シミュレーション

本サブテーマでは、サブテーマ①～③で開発される HPC-MW を、構造解析、流体解析、および連成解析の先端的シミュレーションコードの開発に適用する。そして、HPC-MW の援用による開發生産性、保守性、信頼性、可搬性、移植性を評価・検証する。とくに、コード開発者が記述するコードを 1,000 ステップ程度にすることができるかどうか、ハイエンド計算機環境におけるシミュレーションコード開発工程を従来の 10 分の 1 以下に短縮できるかどうかを評価・検証する。

HPC-MW の改良のため、この評価結果をさらにシミュレーションの計算処理パターンの抽出・体系化、HPC-MW の開発にフィードバックする。地球シミュレータを使用した実効 5 TFLOPS 以上の計算性能の実現を目指す。

さらに、開発されたシミュレーションコードを、これまで不可能であったような大規模で複雑なシミュレーションに適用する。そして、開発されたコードの有効性を評価する。また、ハイエンド計算機上での性能評価を HPC-MW の開発へフィードバックする。

また、本研究で開発される HPC-MW を公開し、広い分野での効果とニーズを分析するものとする。本サブテーマで開発されるシミュレーションソフトウェアの開発技術を基に、本課題で開発される HPC-MW を利用した多様なシミュレーションソフトウェアシステムが開発される。

⑤システムの拡張性について

HPC-MW の最大の利点は、日進月歩のハードウェアの変革に関わらず、移植性、信頼性、演算性能に富んだプログラムを効率的に開発できることである。当然のことながら、この利点を維持し続けるには、本システムが常に新たな状況に対応して更新されなければならない。すなわち、本システムの拡張性については、特に以下の点に留意する：

- ・新規アーキテクチャへの対応
- ・メッシュデータに対するフレキシビリティ

前者に関しては、「コンパイラ型」アプローチによって、現在の SMP クラスタ型アーキテクチャの延長にあるものについては：

- ・メモリ階層の深さ
- ・プロセッサ速度
- ・メモリバンド幅
- ・キャッシュ容量

をパラメータとして、入力メッシュデータの複雑さに応じた、最適なプログラムを自動的に生成する仕組みとして「コンパイラ型 HPC-MW および生成プログラム」を開発する。「HPC-MW 原型ソースファイル」から、各アーキテクチャ、メッシュデータに最適な「HPC-MW」を自動的に生成する。

後者については、「NASTRAN」、「ABAQUS」、「FEMAP neutral」および「GeoFEM」については、一領域のメッシュデータを直接読みこんで、内部的に「HPC-MW」共通データに変換するコンバート機能を開発する予定であるが、より一般的なデータ構造に対応するためのインタフェースを開発する必要がある。基本的には、On-Demand な対応をとる予定であるが、「HPC-MW」共通データ構造を XML によって体系化しておき、新たなインタフェース開発を効率的に実施可能にしていく予定である。

また、「戦略的基盤ソフトウェアの開発」プロジェクトの中では、「次世代流体解析」および「次世代構造解析」グループの共通データ構造をサポートしていく予定である。

7. 基本設計

前述した目的の機能を達成するために、本システムは、以下の各機能ブロック (5つの大項目) から構成される：

- ①ライブラリ型 HPC-MW
- ②コンパイラ型 HPC-MW および生成プログラム
- ③ネットワーク型 HPC-MW および生成プログラム
- ④検証用シミュレーションコード
- ⑤並列メッシュ生成、領域分割用ユーティリティソフトウェア

図3に本システムの全体構成を示す。また、各機能ブロックの相互関係を以下に述べる。

①ライブラリ型 HPC-MW

「ライブラリ型 HPC-MW」は、並列有限要素法の代表的な計算プロセスである：

- ・データ入出力
- ・適応格子
- ・動的負荷分散
- ・並列可視化
- ・線形ソルバ
- ・有限要素処理 (マトリクス計算、コネクティビティ生成)

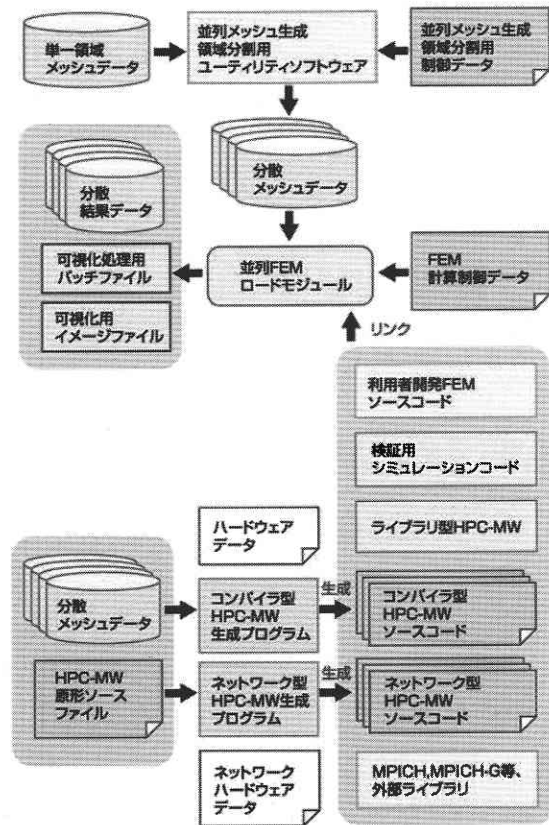


図3 Configuration of the Whole System of HPC-MW

・連成カップリング

について、既存のハイエンド並列計算環境（ベクトルまたはスカラープロセッサから構成されるSMPクラスタ型および分散メモリ型並列計算機）に関して最適化されたライブラリである。これにユーザーがPC上で開発した単独CPU用のFEMコードをリンクすることによって、様々な並列計算環境において最適化されたコードが生成される。「ライブラリ型HPC-MW」を使用することによって、様々なハイエンド並列計算環境において信頼性の高いシミュレーションコードを効率的に開発できる。また「GRID」上で利用可能な様々な計算機リソースのパワーを最大限に利用することが可能となる。

②コンパイラ型HPC-MW

有限要素法では非常に複雑な形状を扱うが、ハードウェアの性能を最大限引き出すためには、形状データに応じたプログラミングが必要となる。特にスカラープロセッサの場合、キャッシュ容量に応じたプログラミングは効果的である。並列計算の場合は各プロセッサあるいはノードごとに異なるプログラムが必要となる場合もある。また「ライブラリ型HPC-MW」は、既存のハイエンド計算機環境に対して開発されたものであるが、新しいプロセッサが開発された場合、また、新しいアーキテクチャの並列計算機が開発された場合、また新たに「HPC-MW」を開発しなければならない。今後しばらくは、SMPクラスタ型アーキテクチャをベースに、共有メモリユニットの階層が深くなっていくような傾向が続くと考えられることから：

- ・メモリ階層の深さ
- ・プロセッサ速度
- ・メモリバンド幅
- ・キャッシュ容量

をパラメータとして、入力メッシュデータの複雑さに応じた、最適なプログラムを自動的に生成する仕組みがあれば便利である。本システムではこのような仕組みを「コンパイラ型HPC-MW」と呼び、「HPC-MW原型ソースファイル」から、各アーキテクチャ、メッシュデータに最適な「HPC-MW」を自動的に生成する。

③ネットワーク型HPC-MW

「ネットワーク型HPC-MW」は「コンパイル型」とよく似た仕組みであるが、GRID等、ネットワーク上で共有されているヘテロジニアスな並列計算機群を同時に利用して大規模有限要素シミュレーションを実施する場合に、利用する計算機リソースそれぞれに最適化された「HPC-MW」を各計算機リソースの性能に基づくパラメータをもとに自動生成するものである。

④検証用シミュレーションコード

「検証用シミュレーションコード」はこれら開発された各種「HPC-MW」の性能、有用性を検証するために、HPC-

MWを使用して開発される並列有限要素法によるプログラム群である。HPC-MWのユーザーは、これら検証用シミュレーションコードとHPC-MWとの結合のさせ方を参考にすることで、各自のプラグイン作業の効率化を図ることができる。もちろん、これら検証用シミュレーションコードはそのまま高性能な有限要素法解析アプリケーションであるため、ブラックボックス的に使うこともできる。

⑤並列メッシュ生成、領域分割用ユーティリティソフトウェア

「並列メッシュ生成、領域分割用ユーティリティソフトウェア」は、並列有限要素法の計算に使用する、大規模非構造メッシュの生成、および領域分割を効率的に実施するためのユーティリティである。「NASTRAN」、「ABAQUS」、「FEMAP Neutral」など、商用汎用コードのメッシュデータに関するインタフェースも備えており、「HPC-MW」を使用して開発したFEMコードでは、これらのコードで使用したメッシュファイルをそのまま利用することが可能である。

上記の①～③はそれぞれ、ベクトルプロセッサ、スカラープロセッサを使用したSMPクラスタ型および分散メモリ型の両アーキテクチャに対応したバージョンを開発する。具体的なターゲットマシンとしては以下を考慮している：

- ・ベクトルプロセッサ，SMPクラスタ型：「地球シミュレータ」、NEC-SXシリーズなど
- ・ベクトルプロセッサ，分散メモリ型：富士通VPPなど
- ・スカラープロセッサ，SMPクラスタ型：COMPAQ, Hitachi SR 8000, IBM-SPなど
- ・スカラープロセッサ，分散メモリ型：PCクラスタ

8. 本年度の成果「反復法並列線形ソルバー」

ここでは、ライブラリ型HPC-MWの中核をなす反復法並列線形ソルバーについて、ベクトル、スカラールの両プロセッサに関する最適化、チューニングについての検討結果について述べる^{1,2)}。

有限要素法を始めとする科学技術計算のプロセスの大半は、線形化された微分方程式の離散化によって得られる大規模連立一次方程式の求解に費やされる。連立一次方程式の解法としてはガウスの消去法などの直接法（Direct Method）が広く使用されてきたが、大域的（Global）処理が生ずるため、並列計算には適していない。並列計算に適した手法として共役勾配法（Conjugate Gradient）などの反復法（Iterative Method）が注目されている。反復法の収束特性は係数行列の固有値分布に依存するため、実用的な問題に適用するためには前処理（Preconditioning）を施し、固有値分布を変えた行列を解く手法が一般的である。反復法の前処理手法としては不完全LU分解（Incomplete LU

Factorization, ILU) あるいは不完全コレスキー分解 (Incomplete Cholesky Factorization, IC) などがよく使用される。ILU/IC (以下 ILU) は一種の不完全な直接法であり、大域的な処理が生じるために、並列化に適した局所的な前処理手法が必要となる。

前処理つき反復法における計算プロセスは以下の4種類に分類される：

- ・行列ベクトル積
- ・ベクトル～ベクトル内積
- ・ベクトル (およびその実数倍) の加減
- ・前処理

このうち「ベクトルの加減」を除く各プロセスでは領域間の通信が発生する。「行列ベクトル積」は計算前に通信を実施すれば局所的な処理が可能である。「内積」は MPI_ALLREDUCE などのサブルーチンを使用して容易に達成可能である。最後の「前処理」については前処理手法によって異なる。例えば ILU などの手法は前進/後退代入により大域的な変数の依存性が生じるため、並列化が困難である。単独プロセッサを使用した計算の場合、Fill-in のない ILU (0) 法を前処理として使用すると、前処理計算部分が全体の 50% 程度を占めるため、前進/後退代入部分の並列化は計算効率の向上のために不可欠である。

本研究では局所前処理法 (局所 ILU (0) 法, Localized ILU (0)) を中心に検討を進める。局所 ILU (0) 法は一種の「擬似」ILU (0) 法である。局所 ILU (0) 法では前進/後退代入計算時に領域外からの影響 (すなわち外点の影響) を 0 とすることによって、前処理の局所化を行い、並列性の高いアルゴリズムを実現している (図 4)。

本研究では、反復法としては：

- ・ Conjugate Gradient (CG)
 - ・ Bi-Conjugate Gradient Stabilized (BiCGSTAB)
 - ・ Generalized Minimal Residual (GMRES)
 - ・ Generalized Product-type Bi-Conjugate Gradient (GPBiCG)
- の使用を検討する。

1 節点に 1 自由度のスカラーソルバーのほか、1 節点に多自由度が存在する場合に関するブロックタイプのソルバーも用意する。ブロック内の自由度が変化する場合、一つのプログラムの中でブロックサイズが変化する場合に対しても対応する。

前処理手法としては、前述の局所 ILU (0) / IC (0) 法の他に、Block Scaling, Point Jacobi, SAI (Sparse Approximate Inverse, Algebraic Multigrid (Algebraic Multigrid)) を用意する。

ハードウェアに依存したチューニングは、科学技術計算コードの移植性を高める上で非常に重要なプロセスである。本研究では、ベクトル、スカラーの両プロセッサを対象としたミドルウェアを開発するため、それぞれに関す

る最適化、チューニングが必要である。

一般に、非構造格子を使用した計算において高いベクトル/並列性能を得るためには以下の3点が重要である：

- ・局所的な処理
- ・連続メモリアクセス
- ・充分なループ長 (ベクトルプロセッサの場合)

さまざまな局面において、これらが実現されるよう、節点あるいは要素のリオーダーリングを実施する。図 5 は剛性行列成分を依存のないようにオーダーリングする 3 様式を示している。図 5(a) における同じ番号の Hyperplane 上にある成分は並列/ベクトル実行が可能である。図 5(b) では、同じ color 番号に属する成分は並列/ベクトル実行が可能である。また、図 5(c) では、ベクトル長を確保するために Hyperplane がグループ化されている。図 6 はこれら互いに非依存の剛性行列成分を格納する 3 通りの格納様式、アクセス順を示している。図 6 の 3 様式がアーキテクチャの違いに及ぼす影響を図 7 に示す。図 7(a), 7(b) はそれぞれ、SMP クラスタ型ベクトル計算機、スカラプロセッサにおける弾性解析の演算性能を示している。アーキテクチャの違いに応じて適切なオーダーリングが必要であることがわかる。HPC-MW では、ユーザーであるコード開発者からできるだけこれらのプロセスを隠蔽できるような仕様を検討する。

スカラプロセッサについてはキャッシュ容量を考慮したブロッキングが必要である。キャッシュ容量に関する特性を計測するためのベンチマークコード群を整備し、さまざまなプロセッサに関して特性を評価する。キャッシュ容量によって最適なブロックサイズは異なるため、その変化に応じて別々のミドルウェア (サブルーチン) を使用する必要が生じる場合がある。このような場合を想定して、フレキシブルでかつコンパクトなミドルウェアの実装法についても検討する。

9. 応用分野

本 HPC-MW の開発により、ローエンドからハイエンドまで様々な計算機環境において、科学技術計算コードの新規開発が効率化され、また既存コード資産の有効利用を図ることができる。また、ひいては、下記のようなさまざまな波及効果をもたらすものと期待される。

- ・科学技術シミュレーションの代表的な標準ソフトウェア体系の確立、インタフェースの共通化によるソフトウェア開発環境の飛躍的な改善。
- ・マルチディシプリナリーな研究・交流の活性化。
- ・HPC-M の公開、普及により大学、産業界における科学技術シミュレーションを担う人的資産の充実化
- ・本プロジェクトでとりあげたシミュレーションをモデルケースとした、他のシミュレーション分野の「GRID」

「ITBL」広域分散処理環境上へのシミュレータの構築。
 ・ハイエンド計算機のハードウェア・ソフト市場と先端基礎研究の相乗的發展，技術的・経済的国際競争力の強化。
 さらには，ソフトウェア開発基盤の充実による，さらなる高性能ハードウェアの新規開発を促進。

(2003年4月2日受理)

参 考 文 献

- 1) 中島研吾, 奥田洋司, ハイエンド計算機環境における科学技術計算コード開発基盤 (HPC ミドルウェア), 計算工学講演会論文集, Vol. 7, No. 1, pp. 199-202, 2002.
- 2) Kengo Nakajima and Hiroshi Okuda, HPC Middleware-Infrastructure for Development of Scientific Applications on Various Types of HPC Environments-First Step for ACES/GRID, Proc. 3rd ACES International Workshop, p. 95, 2002.

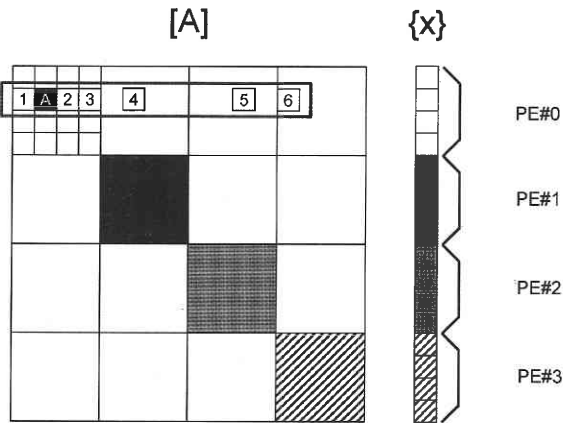
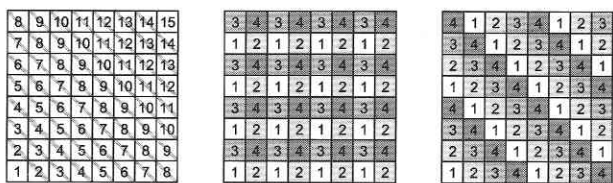


図 4 Forward/Backward Substitution for Localized ILU(0)
 Component 'A' on PE#0 is related to six non-diagonal components. Components '4', '5' and '6' are ignored at the forward/backward substitution for the localized ILU(0), since they belong to other PEs.



(a) Hyperplane/RCM (b) Multicoloring: 4 colors (c) CM-RCM: 4 colors

図 5 Ordering for Vector Processors and SMP clusters

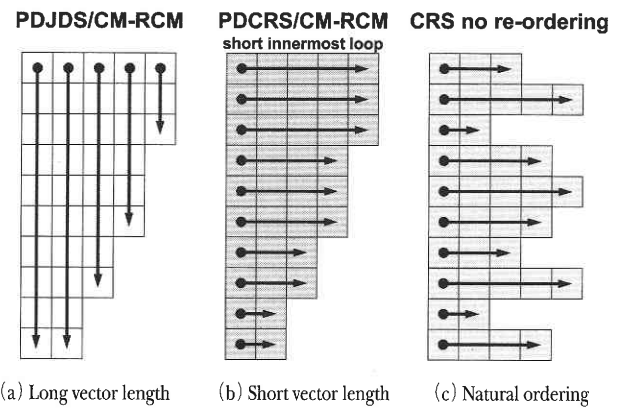


図 6 Storing Styles and Accessing Directions of Independent Components of Matrix

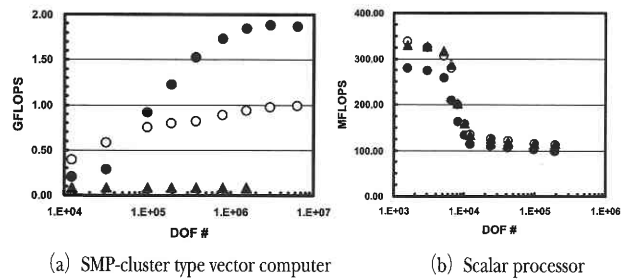


図 7 FLOPS Rate of 3 D Elastic Analysis Under Various Ordering (see Fig.6)

1 node of Hitachi SR8000 (8 CPU, 8 GFLOPS/peak) with pseudo vector option is used.
 ● : PDJDS/CM-RCM ordering, ○ : PCRS/CM-RCM ordering, ▲ : Natural Ordering