平成 22 年度　実績報告

「ポストペタスケールコンピューティングのためのフレームワークとプログラミング」

平成22年度実施報告書

研究代表者　　佐藤三久

筑波大学・計算科学研究センター　センター長・教授

1. 研究実施の概要

現在、最先端の計算科学に用いられる高性能計算システムの性能はペタフロップス(1 秒間に 1,000 兆回の演算能力)に達し、ポストペタスケールシステムとして、エクサスケールのシステムに向かおうとしている。我々の目標は、ペタスケールコンピューティングを超えてエクサスケールコンピューティングに到達する、超最先端の高性能コンピューティングへの道を開拓するべく、ソフトウェア技術、プログラミングモデルおよび言語を確立することである。

初年度である当該年度においては、ここの要素技術について、予定されていたタスクグループで検討を進め、全体のプロジェクトに繋がる、いくつかの成果を得た。日本側のチームにおいては、マルチコア環境における集団通信の最適化手法の設計とプロトタイプ(タスク2東大チーム)、GPU クラスタ向けの高性能通信ライブラリのための予備実験、故障検知および通知(タスク2東工大チーム)に関する設計を行った。また、大規模データ管理については、フランス側とともに、ポストペタスケール時代におけるデータ共有ソフトウェアアーキテクチャの概念設計を行った。さらに、CPU と GPU を統合的に利用するためのプログラミングインタフェースとして、XMP を拡張して検討・設計した(タスク3筑波大グループ)。また、複数GPUを用いたアプリケーションを性能モデルベースの性能チューニングをおこない、自動チューニング可能なフレームワークの検討を行った(タスク3東工大グループ)。

また、マルチコア向けの階層的並列構造を持つハイブリッドアルゴリズムによる固有値解法やFFTなどの数値の数値アルゴリズムについて、検討し、評価した(タスク4筑波大グループ)。マルチコアおよび GPU クラスタ向け科学技術アプリケーション開発基盤の設計とプロトタイプ開発を行った(タスク4東大グループ)。さらに、ライブラリ構築フレームワークの構築にむけては、基本的な要件、特にライブラリ／アプリケーションの記述方法の要件を明らかにするためのケーススタディを実施した(タスク4京大グループ)。

以上、日本側の研究においては着々と進行している。しかし、今年度においては、フランス側とキックオフのミーティングを9月に開催し、プロジェクトの意識合わせと計画の検討を行ったが、実際にフランス側でプロジェクトが開始されたのは日本側からほぼ半年おくれでスタートしている。そのため、それぞれのタスクでの成果の検討ととりまとめ、特に、プロジェクトの異なる研究項目間のインタフェースの検討については、来年度前半に行う予定である。

2. 研究実施体制

グループ名	研究代表者又は 主たる共同研究者氏名	所属機関・部署・役職名	研究題目
筑波大グループ	佐藤　三久	筑波大学・計算科学研究センター・教授／センター長	ポストペタスケールコンピューティングのためのプログラミング言語拡張および数値計算アルゴリズム、性能評価、大規模データ管理技術
東大グループ	中島　研吾	東京大学・情報基盤センター・教授	ポストペタスケールコンピューティングのためのヘテロジニアス環境アプリケーション開発基盤
京大グループ	中島　浩	京都大学・学術情報メディアセンター・教授／センター長	ポストペタスケールコンピューティングのためのアルゴリズム／技法ライブラリ構築のためのフレームワーク
東工大グループ	松岡　聡	東京工業大学・学術国際情報センター・教授	ポストペタスケールコンピューティングのためのアクセラレータ技術

3. 研究実施内容

1. ポストペタスケールコンピューティングのためのプログラミング言語拡張および数値計算アルゴリズム、性能評価、大規模データ管理技術（筑波大チーム）

(1)プログラミング言語拡張

本プロジェクトでは、ポストペタスケールコンピューティングのための並列プログラミング基盤として、並列プログラミング言語拡張仕様 XcalableMP(XMP)を用いて、他のグループで研究開発されるコンポーネントへのインタフェースを与えるとともに、上位のプログラミングパラダイムとして、ワークフロー言語である YML を用いる。本年度においては、その統合のインタフェースを検討した。

研究員を派遣し、XMLとYMLをどのように統合するかについての調査を行った。YMLは、コンポーネント間のデータの流れを記述するワークフロー言語であり、現在のところ、このコンポーネントは逐次プログラムに限られている。このプロジェクトでは、このコンポーネントを XMP 言語で記述することにより、並列プログラムとすることである。XMP 言語は基本的に MPI による並列プログラムにコンパイルされ、並列実行されるために、このためには MPI で記述された並列プログラムをコンポーネントとする拡張が必要である。現在の YML は、グリッド RPC(遠隔手続き呼び出し)ミドルウエアを使って、実装されている。その一つの実装として、筑波大で開発された OmniRPC を用いた実装がある。一つの解決方法としては、OmniRPC の遠隔手続きをMPIの並列プログラムにすることによって、これが実現できることが分かった。今後、次年度にかけて、詳細について議論を進め、統合するための設計・実装を行う予定である。また、この派遣では、日仏の双方のサイトに、XMPとYMLを利用できる環境を整備した。また、フランス側の1年目の終わりに全体の設計についての報告書を作成する予定になっており、次年度の前半で、この報告書をまとめる。

また、本チームにおいては、演算加速機構(GPU)と CPU の並列環境を統合する言語拡張仕様について検討

した。並列記述言語 XcalableMP（以後、XMP）に対し、マルチコア／マルチ GPU ノードを対象としたプログラミングを実現するための言語仕様拡張、実行支援ライブラリ設計、想定されるコンパイルコードに対する性能評価を行った。従来の XMP は分散メモリ環境に対する均質なデータ及びプロセス分散のみが可能であり、マルチコアノードにおける共有アドレス空間を含むハイブリッドプログラミングに対応していなかった。また、ホモジニアスな CPU 環境のみが対象であり、GPU 等の加速装置を含むヘテロ環境にも対応していなかった。

本年度の研究により、各ノード上に複数コア及び複数 GPU が存在するような一般的なマルチコア／マルチ GPU 環境がポスト PFLOPS 時代の一つの典型的大規模クラスタの姿であることを想定し、XMP におけるデータ配列分散表現に、各ノードの分散メモリ環境上に複数コアによる共有メモリと、個別のアドレス空間を持つ複数 GPU に対応する記述機能を追加した。これにより、従来明示的に記述されていた CPU／GPU 間データ交換が XMP の枠組みで自然に記述できるようになる。また、従来は CPU と GPU を機能分散的に用いて来たことに対し、この拡張 XMP の枠組みでは巨大な配列に対するループ分割処理を CPU と GPU に負荷分散し、両者を論理的に等価な計算リソースとしてみなせるようにした。コンパイラの実装は H23 年度以降の計画となるが、コンパイル後コードを想定したランタイムライブラリ設計と、数ノードの実システム上での負荷分散テストを行い、N 体問題や行列計算等の典型的 HPC ベンチマークにおいて、一定の CPU/GPU 分散比率の下で、GPU のみを用いる場合より最大で2倍程度までの速度向上が得られる事を確認した。これにより、拡張 XMP における CPU/GPU 混合演算が有望であることが示された。

（2）数値計算アルゴリズム・性能評価

ポストペタスケールに向けた計算アルゴリズムの研究においては、階層型の並列構造を持つ固有値解法を対象として、その計算カーネルの一部であるシフト方程式計算およびブロック方程式計算について、マルチコア向けの効率的な実装方法の検討を行い、格子 QCD で現れる問題を対象としてその性能評価を行った。また、密型と疎型のアルゴリズムのハイブリッド型解法の開発を行い、密度汎関数法で現れる問題を対象として性能ボトルネックの評価を行った。解法において必要とされる効果的なパラメータの選択を統計的手法によって行う方法についても開発を進めた。

当該年度においては、本プロジェクトがターゲットとするアプリを規定するためにベンチマークの策定を開始することになっており、そのための検討・議論の訪仏を3月に予定していたが、震災のために中止となった。このベンチマークに関する報告書も、フランス側の1年目の終わりに作成する予定になっており、次年度の前半で、フランスに訪問し、議論の上、この報告書をまとめる。

（3）大規模データ管理技術

大規模データ管理技術について、日本側で、広域分散ファイルシステムのメタデータを管理するためのマルチマスター型分散メタデータサーバ HGMDS の設計を行った。フランス側で研究開発を行っているロックフリーの並列ストレージシステム BlobSeer と組み合わせることにより、大規模データへの高速アクセス、広域環境における低遅延アクセス、クライアント数、ファイルサーバ数に対しスケールアウトする性能を実現することが可能と考えられ、そのための概念設計を行った。

研究員を派遣し、BlobSeer のアーキテクチャを解析し、BlobSeer の基本的な性能評価を行った。BlobSeer と HGMDS を組み合わせた広域ファイルシステムの設計を行った。ファイルシステムの設計段階で、BlobSeer 自身のメタデータが、広域環境へ配置した際に問題となる事が明らかになった。BlobSeer を広域環境に対応するため、BlobSeer のメタデータ I/O を高遅延通信路を介して行わないよう、アーキテクチャの改良設計を行った。今

後の研究のために、フランスの広域計算機環境である Grid5000 の利用についての環境整備も行った。今後は、フランス側が BlobSeer を改良し、広域環境への対応を行った後に、HGMDS と改良 BlobSeer を組み合わせた広域ファイルシステムの設計と実装を行う。

2. ポストペタスケールコンピューティングのためのヘテロジニアス環境アプリケーション開発基盤（東大グループ）

（1） 統合数値ライブラリ

平成 22 年度はマルチコアおよび GPU クラスタ向けに、基本設計とプロトタイプの開発を実施した。

有限要素法、差分法、境界要素法の各離散化手法について、ユーザ定義による形状データ構造への I/O インタフェースの基本的な検討を行った。

マルチコアおよび GPU クラスタにおいて、ハイブリッド並列プログラミングモデルを適用した場合の領域分割手法として、Nested/Extended HID 法（Hierarchical Interface Decomposition）を提案した。HID 法は、オーバーラップ領域を持たない Nested Dissection 法を厳密に適用した並列計算向け領域分割手法であり、グローバルな依存性を有する計算プロセス、特に ILU 前処理の並列化に適している。Extended HID は本グループによって提案された手法で、セパレータ厚さを変えることによって、深いフィルインを有する前処理にも対応している。本研究では、各領域内のスレッド並列化にも HID を適用した Nested/Extended HID を提案し、従来の Multicoloring に基づくスレッド並列化と比較して、並列化効率の高い計算を実施することが可能となった。提案手法を規則正しい形状に基づく有限要素法アプリケーションに適用し、悪条件問題、フィルインの深い前処理への適用性をマルチコアクラスタ（T2K（東大））を使用して実施した。

GPU は CPU と比較して高いメモリバンド幅を有し、memory-bound な科学技術計算への適用が期待されているが、現状の適用範囲は規則正しいデータ構造を有する差分法にほぼ限定されている。本研究では、不規則なデータ構造に基づく有限要素法への GPU の適用の可能性を検討するため、シングル CPU 向けに開発された三次元固体静力学有限要素法コードの GPU 向けの実装、性能評価を実施した。対象コードは対称正定な疎行列を前処理付共役勾配法によって解いている。本年度はブロック対角 LU 分解による簡易な前処理を用いて、様々なブロック化、疎行列格納法による性能評価を実施した。有限要素法は要素マトリクス生成、全体マトリクス生成、疎行列計算など様々な計算プロセスを含んでいる。マルチコアーGPU によるヘテロジニアスな環境における、各プロセスの各計算デバイスに対する最適なデータ構造に関する検討を実施した。2月に助教を約 2 週間ボルドー大学に派遣し、GPU における有限要素法プログラムの実装、最適化についての情報交換を実施した。

（2） 実行時環境

平成 22 年度は、マルチコア環境における集団通信の最適化手法の設計とプロトタイプ、故障検知および通知に関する設計を行った。マルチコア環境における集団通信の最適化手法の設計とプロトタイプでは、MPI 通信ライブラリ 3.0 規格で導入予定の非同期集団通信機構の最適化手法を設計しプロトタイプを実装した。非同期通信では、アプリケーションの実行スレッドとは別に通信処理をしていくための別スレッドが必要になる。現状のナイーブな実装では、各プロセスにユーザには見えないスレッドが生成されている。マルチコア環境において、ユーザアプリケーションが一つのコアに一つのプロセスを生成するようなことをすると、コアの数の 2 倍の数のスレッドが生成されることになり、非効率となる。そこで、ネットワークデバイスの割り込み処理時に非同期通信処理を

行う手法を提案しプロトタイプを実装した。本手法は、EuroMPI2010 国際会議で発表した。さらに、フランスボルドー大学で開発されている NEWMadeline 通信ライブラリ上を用いてユーザレベルで非同期通信処理を一括管理する機構を設計し、現在（H23 年 1 月 28 日）、実装している。このために後期博士課程の学生を約 4 カ月間ボルドー大学にインターンシップとして派遣した。

また、エクサスケール環境における、故障検知、通知に関する設計を検討し、成果発表を行った[2]。この検討では既存記法である例外処理に類似した記述法を提案し、目的である外部モジュールのエラーハンドリング、さらにチェックポイント／リスタートやデータ破損によるサイレント故障のハンドリング等の基本的な耐故障機能が容易に実装可能であることを示した。提案した記述法は、既存アプリケーションコードに適切なディレクティブを挿入することにより、プログラマにハンドリングコードを記述可能とする。このハンドリングコードはモジュール間のイベント送受信を提供するレイヤにより呼び出され、プログラマは情報の受信やハンドリングコードの呼び出しなどの実装をする必要がなく、純粋にアプリケーションの修復に集中したハンドリングコードを記述できる。

3. ポストペタスケールコンピューティングのためのアルゴリズム／技法ライブラリ構築のためのフレームワーク（京大グループ）

平成23年度は、ライブラリ構築フレームワークの基本的な要件、特にライブラリ／アプリケーションの記述方法の要件を明らかにするために、具体的なアプリケーションへの適用に関するケーススタディを実施した。具体的なアプリケーションとして、電磁場解析、プラズマシミュレーション、および格子ボルツマン法による流体シミュレーションを取り上げ、アプリケーションに応じた線形ソルバーの選択・適応化と、アプリケーション固有のループコードに対する最適化・並列化技法ライブラリの適用法について検討した。

まず線形ソルバーについては、大規模な電磁場解析に対するマルチグリッド法の適用についてさまざまな検討を行った。特に、幾何マルチグリッド法をアプリケーション固有のデータ構造に応じて適用する方法の検討や、新たな並列スムーザの開発・評価を行った。またマルチグリッドソルバー自体をターゲットとして、後述の最適化・並列化技法の適用に関するケーススタディを行った。

最適化・並列化技法のライブラリ化と適用法については、プラズマシミュレーションのための負荷分散ライブラリ OhHelp の適用法、互いに依存関係を持つ複数の空間ループのタイリングによるメモリアクセス向上、および多次元配列の境界データ通信の最適化について検討した。その結果、これらの並列化・最適化に共通する事項として、アプリケーション固有のループ構造や複数のループの依存関係などが技法の適用法と深く関係し、ループ構造の変形などアプリケーションコードの大幅な変更が必要となるケースがほとんどであることが判明した。

そこで多様なアプリケーションと技法を包含する共通的なフレームワークとして、主に空間的なループを局所的な視点で記述することで、ループボディとループ構造とを分離可能なプログラミングパラダイムが必要であると言う結論を得た。具体的には、アプリケーションコードの本質的な意味をループボディの列で記述し、ループ構造については上下限などの基本的な情報のみを分離して記述することで、タイリングなど複雑なループ構造が生じるような最適化・並列化技法が容易に適用可能となる。またタイリングと境界データ通信最適化の組み合わせのように、直行的な複数の技法の同時適用や、適用技法選択の試行錯誤なども、この分離記述により容易に実施可能となる。

今後このフレームワークについて、具体的な記述仕様を詳細化し、アプリケーション（ドメイン）固有の知識・情報の表現・利用を含めたプロトタイピングを実施する。

4. ポストペタスケールに向けたアクセラレータ技術（東工大グループ）

平成 22 年度　実績報告

平成 22 年度は、大規模 GPU システムのためのスケーラビリティ向上技術および耐故障技術の提案・評価を行った。評価環境として主に、本グループが中心となって本年度導入した東工大 TSUBAME 2.0 スパコンを主に利用した。TSUBAME 2.0 は約 4,200 デバイスの NVIDIA M2050 GPU を搭載し、理論性能 2.4PFlops のペタスケールシステムである。

(1) 強スケーリングモデルの提案および強スケーリングアプリAuto-Tuning

スケーラビリティ向上技術については、チップあたりの演算性能の向上につれ相対的にコストが上昇する通信時間の隠ぺいについて、CFD カーネル、線形演算カーネルなどにおけるケーススタディを行った。ノードにまたがった GPU プログラムにおいては、ノード間通信、ホスト-GPU 間通信の双方が問題となる。CFD カーネルについては、本グループで作成中の GPU クラスタ向けステンシル演算コード生成システムに境界領域通信と演算のオーバラップ処理を組み込み、マルチ GPU 環境において効果の評価を行った。線形演算カーネルにおいては、GPU クラスタ向け Linpack ベンチマーク実装において、DGEMM(行列積)演算と MPI 通信とホスト-GPU 通信全てのオーバラップ処理を組み込んだ。評価としては TSUBAME 2.0 スパコンのほぼ全体にあたる、1357 ノード 4071GPU を用いた。その結果 1.192PFlopsの性能が得られ、国内では初のペタフロップスの達成および Top500 スーパーコンピュータランキングにおいて世界 4 位を達成した。他に疎行列演算カーネルを GPU クラスタ上で実装し、hyper-graph partitioning に基づく通信コスト削減を行った。以上の通信コスト隠ぺい・削減処理の効果について実証を行った一方で、その処理の多くは手作業によるソフトウェア変更に頼っており、その自動化は今後の課題の一つである。

(2) 大規模 GPU 耐故障機能の開発

大規模 GPU システムでの耐故障性実現に向けて、開発中の CUDA GPU プログラム向けチェックポインタと MPI プログラム向けチェックポインタの統合を行った。GPU プログラムにおいては、GPU がホストメモリと別のメモリ空間を持っていることから、通常のチェックポイント処理に加え GPU メモリ空間の保存、復帰が必要となる。この処理と MPI プロセスをまたいだチェックポイントの両立のためにシグナルハンドラの有効区間の設定などの対処を行った。なおスケーラブルな耐故障性の研究の推進のために、H23 年 3 月より約 9 カ月間、博士課程学生を米国・NCSA-INRIA Joint Laboratory およびフランス・INRIA に派遣している。

4. 原著論文発表

[1] Yasunori Futamura, Hiroto Tadano, and Tetsuya Sakurai, Parallel stochastic estimation method of eigenvalue distribution, JSIAM Letters, Vol. 2, pp. 127–130, 2010.

[2] Akihiro Nomura and Yutaka Ishikawa, "Design of Kernel-level Asynchronous Collective Communication," Proceedings of Euro MPI, September 2010.

[3] 中島研吾，並列プログラミングモデルと自動チューニング，日本応用数理学会誌 20-4，2010.

[4] T. Mifune, Y. Takahashi, and T. Iwashita, "New Preconditioning Technique to Avoid Convergence Deterioration due to the Zero-Tree Gauge Condition in Magnetostatic Analysis," IEEE Trans. Magn., Vol. 46, No. 7, pp. 2579-2584, 2010.

[5] Y. Takahashi, T. Tokumasu, A. Kameari, H. Kaimori, M. Fujita, T. Iwashita, and S. Wakao, "Convergence Acceleration of Time-Periodic Electromagnetic Field Analysis by Singularity Decomposition-Explicit Error Correction Method," IEEE Trans. Magn., Vol. 46, No. 8, pp. 2947-2950, 2010.

[6] 廣谷迪，美舩健，岩下武史，村山敏夫，大谷秀樹，「並列幾何マルチグリッド法による大規模高周波電磁場有限要素解析」，電子情報通信学会論文誌 B, Vol. J93-B, No.9, pp. 1331-1341, 2010.

[7] 美舩健，廣谷迪，岩下武史，村山敏夫，大谷秀樹;「マルチコアプロセッサシステムによる高速有限要素電磁界解析」，情報処理学会論文誌:コンピューティングシステム（ACS）, Vol. 3, No. 3, pp. 189-198, 2010.

[8] 南武志，岩下武史，中島浩，「キャッシュメモリを考慮した 3 次元 FDTD カーネルの性能改善」，情報処理学会論文誌 コンピューティングシステム, Vol. 4, No. 1, 2011.

[9] Ali Cevahir, Akira Nukada, and Satoshi Matsuoka. “High Performance Conjugate Gradient Solver on Multi-GPU Clusters Using Hypergraph Partitioning” In Proceedings of the 2010 International Supercomputing Conference (ISC’10), Hamburg,Germany, June 2010.

[10] Leonardo Bautista Gomez, Akira Nukada, Naoya Maruyama, Franck Cappello and Satoshi Matsuoka. Low-overhead diskless checkpoint for hybrid computing systems. In Proceedings of 2010 High Performance Computing Conference (HiPC 2010), Goa, Dec. 2010

[11] Koichi Shirahata, Hitoshi Sato, SATOSHI MATSUOKA. Hybrid Map Task Scheduling for GPU-based Heterogeneous Clusters, First International Workshop on Theory and Practice of MapReduce (MAPRED'2010), First International Workshop on Theory and Practice of MapReduce (MAPRED'2010), Jan. 2011.

5. 主催したワークショップ等

年月日	名称	場所	参加人数	概要
平成22年8月4日	日仏 FP3C キックオフミーティング（国内）	金沢文化ホール	25人	日本側でのプロジェクトの開始にあたっての意識合わせと検討
平成22年9月6－7日	日仏 FP3C キックオフミーティング	INRIA Saclay（1日目）、ベルサイユ大学（2日目）	30人	フランス側とのプロジェクトの開始にあたっての意識合わせと検討

以上