様　式　Ｆ－１９－２

# 科学研究費助成事業　　研究成果報告書

令和　　６ 年　　９ 月　　５ 日現在

機関番号：１２６１２
研究種目：国際共同研究加速基金（国際共同研究強化(A)）
研究期間：2023～2023
課題番号：２２ＫＫ０１８２

研究課題名（和文）ポストペタスケールのための革新的アプリケーション解析基盤の展開

研究課題名（英文）Extension of Innovative Frameworks for Application Analysis in Post-Peta Scale Systems

研究代表者

三輪　忍（Miwa, Shinobu）

電気通信大学・大学院情報理工学研究科・准教授

研究者番号：９０４０２９４０

交付決定額（研究期間全体）：（直接経費）　　7,800,000円
渡航期間：　10ヶ月

研究成果の概要（和文）：2023年4月初旬から2024年3月末まで米国メリーランド州立大学を訪問し，高性能計算に関する複数の共同研究を行った．具体的には，並列アプリケーションのメモリアクセストレース予測に関する研究，GPUの性能ばらつきを考慮した負荷分散に関する研究，CPUとGPUの電力ばらつきを考慮したジョブスケジューリング手法に関する研究の3つを海外共同研究者と共同で行った．これらの研究の一部は本研究課題の計画段階では想定していなかった研究であるが，上記の訪問研究期間中に海外共同研究者と広範囲なテーマについて議論する中で生まれた研究課題であり，本訪問研究の成果と位置付けることができる．

研究成果の学術的意義や社会的意義
海外共同研究者と共同で行った研究はいずれも高性能計算分野においてホットなトピックであり，研究成果をまとめた論文は著名な国際会議や論文誌に採択されることが期待される．特にCPUとGPUの電力ばらつきを考慮したジョブスケジューリング手法に関しては，膨大なエネルギーを消費するスーパーコンピュータの消費エネルギー削減は社会的にも重要な課題であり，将来のカーボンニュートラル社会の実現に貢献することが期待される．

研究成果の概要（英文）：I visited the University of Maryland from April 2023 to March 2024 and conducted various collaborations with a researcher in the university on the field of high performance computing. More specifically, I collaborated with him on studies of predicting memory access traces for parallel applications, load balancing while considering the performance variation between GPUs, and job scheduling while considering the power-efficiency variation among CPUs and GPUs. A part of these studies were not assumed when planning this research project, but have been produced through the discussion with my collaborator on a wide range of research topics with respect to high performance computing. Thus, the results of these studies can be regarded as the outcome of this research project.

研究分野：高性能計算

キーワード：高性能計算　プロファイル　トレース　スケーラビリティ予測

様　式　Ｆ－１９－２

## １．研究開始当初の背景

並列アプリケーションの解析は，性能分析や性能チューニングなどを目的として，高性能計算分野において広く行われている．並列アプリケーション解析は，アプリケーション全体，あるいは，関数単位の実行時間，演算回数，通信関数の呼び出し履歴などの情報をもとに行われるが，これらの情報は，通常，解析対象のアプリケーションを解析対象のシステム上で実行することによって得られる実行時情報である．

この実行時情報を取得する手段としてプロファイリング/トレーシング（PR/TR）とモデリングの 2 つ（表 1）がこれまでに開発されており，用途に応じてこれらのいずれかが利用されていたのが研究開始当初の状況であった．PR/TR は，コンパイラや外部ライブラリの支援によって計測用コードを解析対象のアプリケーションに挿入し，解析対象のシステム上で上記アプリケーションを実行することによって，アプリケーションの挙動に関する詳細情報を取得する．一方モデリングは，少数のノードにおけるプロファイリング結果からアプリケーションのスケーラビリティモデルを構築し，構築したモデルを用いて上記アプリケーションを多数のノード上で実行した場合のアプリケーションや関数単位の実行時間を予測する．これら 2 つの技術は得られる情報量（＋精度）と情報の取得コストとの間にもともとトレードオフが存在していたが，両者のギャップは近年の並列アプリケーションの複雑化・大規模化によって拡大しつつある（図 1）．

表 1．並列アプリケーションの実行時情報の取得方法

| | PR/TR | モデリング | 実行時情報予測 |
|---|---|---|---|
| 解析速度 | 低速 | 高速 | 高速 |
| 情報の種類 | 豊富 | 性能のみ | 豊富 |
| 情報の粒度 | プロファイル/トレース | プロファイル | プロファイル/トレース |
| 情報の精度 | 正確 | 近似 | 近似 |

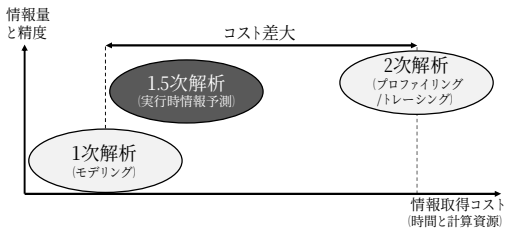


図 1．基課題で解決しようとする問題

上述のギャップを埋めるため，基課題では，少数のノード上で対象アプリケーションを静的・動的に解析することにより，同アプリケーションを多数のノード上で実行した場合の実行時情報を予測する手法（実行時情報予測）の開発を行っていた．具体的には，実行時情報として関数コール回数，キャッシュミス回数，MPI 関数の通信量等に着目し，少ないコア数かつ小さな問題サイズで取得したそれらの情報を用いて当該実行時情報のスケーラビリティを予測するモデルを求め，得られたモデルを用いて多いコア数かつ大きな問題サイズで当該プログラムを実行した際の実行時情報を予測する手法を開発していた．実行時情報予測は現在の 1 次解析（モデリング）と 2 次解析（PR/TR）との間に位置する 1.5 次解析に相当する（図 1）が，並列アプリケーションの PR/TR が資源的・時間的制約により難しい場合において近似的な実行時情報を提供することでエンドユーザの生産性の向上に資する．

一方，基課題にて開発していた手法は，適用可能なアプリケーションの範囲が限定的という問題を抱えていた．具体的には，基課題にて開発していた手法はコア数に比例する量のファイル I/O が必要なことから，アプリケーションの実行規模が増大するにつれて処理速度が低下する．そのため，500 コア程度の実行規模の並列アプリケーションであればこの手法で十分短時間に実行時情報を取得できるが，超大規模（例えば富岳の全系である 763 万コア）なアプリケーションの実行時情報を実用的な時間内に取得することは難しい．また，基課題にて開発していた手法は解析対象のアプリケーションが MPI によって並列化されていることを前提としており，特に予測対象のトレースが MPI 関数の通信トレースに特化されていた．MPI は高性能計算分野において現在主流の並列プログラミングモデルではあるが，MPI 以外を用いて開発されたアプリケーションも実際には存在しており，そうしたアプリケーションのトレース予測にこの手法は利用できない．

## ２．研究の目的

本国際共同研究では，上記の問題点を解決することによって実行時情報予測の適用対象を広範なアプリケーション群に展開することを目的とした．すなわち，実行時情報予測に必要なファイル I/O の量を削減することによって予測対象のアプリケーションの実行規模を拡大するとともに，MPI 以外の方法によって並列化されたアプリケーションのトレース予測にも対応することによって予測対象のアプリケーションの種類を拡大する．本国際共同研究の実施により実行時情報予測は広範囲な並列アプリケーションに適用可能となり，ポストペタスケールに真に資するアプリケーション解析基盤となることが期待された．

## ３．研究の方法

上記目的の達成に向けて，具体的には，実行時情報予測の Charm++への拡張，および，トレース圧縮技術の開発に取り組む計画であった．以下，それぞれの開発方法を述べる．

### ３．１．Charm++への拡張

実行時情報予測の単位をプロセスからイベントに拡張する．MPI における並列処理の単位はプロセスであり，プロセスに対する計算資源の割り当てはアプリケーションの実行中に変化しないため，現在は実行時情報予測をプロセス単位で行っている．一方，Charm++の場合は，Chare と呼ばれるオブジェクトのメンバ関数が並列処理の単位であり，Chare インスタンスに対する計算資源の割り当てはランタイムシステムによって動的に行われる．すなわち，MPI のようにアプリケーションレベルでプロセスの挙動を制御することはできず，Charm++においてはアプリケーション開発者にとってプロセス単位の実行時情報はあまり意味を持たない．Charm++ではメンバ関数の起動，終了，（メンバ関数起動のための）メッセージ送信などをイベントとしてアプリケーション開発者向けにイベントトレースを出力する機能を有していることから，実行時情報予測の単位もイベントとする．実行時情報予測をイベントに拡張するためには，これまでに開発した予測モデルを修正する必要がある．具体的には，プロセス数の関数として定義していたモデルを Chare インスタンス数と計算資源数の関数に変更する．変更後のモデルを用いて Charm++アプリケーションに対する実行時情報予測を行い，予測精度や情報収集コストなどの評価を行う．

### ３．２．トレース圧縮技術の開発

アプリケーション解析に必要な情報の精度を可能な限り保ちつつ，トレースを大幅に圧縮する方法を開発する．圧縮による情報損失の最小化を目的としている先行研究とは異なり，基課題が想定しているトレース予測の応用（1.5 次解析）においてはある程度の誤差が許容されることから，ある程度の情報損失と引き換えに大幅なデータ圧縮が可能な技法（不可逆圧縮）を積極的に利用できる．ただし，復元後のデータに誤差が含まれてもよいかは圧縮対象の実行時情報によって異なるため，不可逆圧縮に加えて可逆圧縮も併用する．例えば，データに数ビットの誤りが発生しても実用上問題ないと考えられる通信タイミングや通信量に対しては不可逆圧縮を行い，1 ビットの誤りが致命的な結果となる通信先に関しては可逆圧縮を適用する．現在開発中のトレース予測プログラム自動生成ツール内に上記のデータ圧縮機能を実装し，トレース予測プログラムの実行性能，および，同プログラムが出力するトレースの精度やサイズ等について評価する．

## ４．研究成果

2023 年 4 月初旬から 2024 年 3 月末まで米国メリーランド州立大学を訪問し，高性能計算に関する複数の共同研究を行った．具体的には，並列アプリケーションのメモリアクセストレース予測に関する研究，GPU の性能ばらつきを考慮した負荷分散に関する研究，CPU と GPU の電力ばらつきを考慮したジョブスケジューリング手法に関する研究の 3 つを海外共同研究者と共同で行った．本章ではそれぞれの研究成果を述べる．

なお，これらの研究の一部は本研究課題の計画段階では想定していなかった研究であるが，上記の訪問研究期間中に海外共同研究者と広範囲なテーマについて議論する中で生まれた研究課題であり，本訪問研究の成果と位置付けることができる．

### ４．１．メモリアクセストレース予測

訪問研究期間中は，主として，並列アプリケーションのメモリアクセストレースを予測する研究を行った．具体的には，小規模実行時のメモリアクセストレースから対象アプリケーションのメモリアクセストレースを予測するモデルを生成し，生成したモデルを用いて大規模実行時のメモリアクセストレースを予測する手法の開発を行った．モデルはロード/ストア命令単位でフィッティングを行うことで生成する．ただし，間接参照等により不規則なメモリアクセスパターンを示す命令に関してはモデルによる予測が難しいと考えられることから，当該アドレスの計算に必要な命令のみを実行することでアドレスを生成する．アドレス計算のみを行うプログラムはアプリケーションコードから LLVM を用いて自動生成する．具体的には，アドレス計算に必要な命令をプログラムスライシングによって抽出し，抽出された命令のみからなるコードを生成する LLVM パスを実装する．本研究で対象とするアプリケーションは LLVM によってコンパイル可能な任意の並列アプリケーションであり，したがって MPI アプリケーションだけでなく Charm++アプリケーションも含まれる．

上記の LLVM パスは，以前の研究で研究代表者が開発した LLVM パスを元に開発する．ただし，以前の研究で開発した LLVM パスは LLVM-3.6.0 を対象としており，最新の LLVM-19.0.0 とは複数のビルトイン関数のインターフェースが異なるため，最新の環境では動作しない．そこで，以前の研究で開発した LLVM パスを LLVM-19.0.0 に移植する作業を訪問研究期間中に行った．移植作業は概ね完了しており，現在動作確認中である．

### ４．２．GPU の性能ばらつきを考慮した負荷分散

研究代表者らが行った実験結果によると，1 つのスーパーコンピュータを構成する GPU は，同一製品仕様であっても製造ばらつきに起因する性能ばらつきが存在する．この性能ばらつきが原因で，マルチ GPU アプリケーションを並列化する際，タスクを均等に分割して各 GPU に割り当てると，GPU 間の処理性能の違いから実行時間にばらつきが生じてしまう．

上記の問題について海外共同研究者と議論した結果，Charm++を用いた解決法を思いつくに

至った．Charm++では Chare と呼ばれる小さな粒度のタスクの集まりとしてアプリケーションを記述し，Chare の GPU への割り当ては Charm++ランタイムシステム内のロードバランサが自動的に行う．したがって，GPU 間の処理性能差を考慮して負荷分散を行うアルゴリズムを Charm++のロードバランサに採用すれば，GPU 間の実行時間のばらつきが減少し，マルチ GPU アプリケーションの性能が向上することが期待される．

上記手法の有効性を実験的に確かめるため，訪問研究期間中に実験環境の整備を行った．具体的には DGEMM アプリケーションをマルチ GPU 上で実行するように Charm++を用いて実装し，既存の Charm++のロードバランサによる負荷分散の効果を検証した．その結果，既存のロードバランサは CPU の負荷のみを考慮して負荷分散を行っており，GPU の負荷は考慮していないことを確認した．今後は GPU の負荷を考慮したロードバランサを開発する予定である．

４．３．CPU と GPU の電力ばらつきを考慮したジョブスケジューリング

1 つのスーパーコンピュータを構成する CPU や GPU は，同一製品仕様であっても製造ばらつきに起因する電力ばらつきが存在することが知られている．ジョブスケジューラが上記の電力ばらつきを考慮して各ジョブの実行に使用するノードを決定することで，システム全体のエネルギー効率の改善やシステムスループットの向上が期待できる．

そこで，訪問研究期間中に，電力ばらつきを考慮したジョブスケジューリング手法を新たに開発した．開発した手法では，ターンアラウンド時間を考慮しつつジョブキュー内の消費エネルギーが大きいジョブに対して優先的にエネルギー効率のよいノードを割り当てることで，ターンアラウンド時間の悪化を防ぎつつシステムのエネルギー効率を改善する．また，各ジョブを各ノードで実行した場合の消費エネルギーを見積もるための電力モデリング手法も新たに開発した．

研究代表者らが行った実験結果によると，開発した手法は FCFS (First Come First Serve)，および，ばらつきを考慮した最新のスケジューリング手法に対して，それぞれ，-17.6%と 5.0%のターンアラウンド時間増加と引き換えにシステムのエネルギー効率を最大 5.7%と 5.2%（平均では 4.1%と 3.6%）改善できた．本研究成果に関する論文は高性能計算分野の著名な国際会議に現在投稿中である．

５．主な発表論文等

〔雑誌論文〕　計0件

〔学会発表〕　計0件

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

| | 氏名<br>（ローマ字氏名）<br>（研究者番号） | 所属研究機関・部局・職<br>（機関番号） | 備考 |
|---|---|---|---|
| | アビナブ　バテル<br><br>(Abhinav Bhatele) | メリーランド州立大学・Department of Computer Science・Associate Professor | |

７．科研費を使用して開催した国際研究集会

〔国際研究集会〕　計0件

８．本研究に関連して実施した国際共同研究の実施状況

| 共同研究相手国 | 相手方研究機関 | | | |
|---|---|---|---|---|
| 米国 | The University of Maryland | | | |