

機関番号：11301

研究種目：若手研究(B)

研究期間：2009～2010

課題番号：21700049

研究課題名(和文)

異種複数のプロセッサを適材適所で活用する高性能計算フレームワークの構築

研究課題名(英文)

A High-Performance Computing Framework to Exploit Various Processors

研究代表者

滝沢 寛之 (TAKIZAWA HIROYUKI)

東北大学・大学院情報科学研究科・准教授

研究者番号：70323996

研究成果の概要(和文)：

本研究の目的は、ソースコードレベルの移植性を高く維持しつつ、必要に応じて複合型計算システム中の各プロセッサの性能を効果的に利用できる高性能計算フレームワークを構築することである。そのために、高水準言語で記述されたプログラムを各種プロセッサ向けに自動チューニングする仕組みや、高水準言語からシームレスに利用できる数値計算ライブラリ、および多様な計算資源を適材適所で利用するためのスケジューリング手法を検討する。

研究成果の概要(英文)：

The purpose of this work is to achieve a high-performance computing framework that can exploit the computing power of each processor in a heterogeneous computing system while keeping the portability of source codes. For making good use of various computing resources, this work explores an auto-tuning mechanism of a high-level language, numerical libraries seamlessly used from the high-level language, and a job scheduling method.

交付決定額

(金額単位：円)

	直接経費	間接経費	合計
2009年度	2,000,000	600,000	2,600,000
2010年度	1,300,000	390,000	1,690,000
年度			
年度			
年度			
総計	3,300,000	990,000	4,290,000

研究分野：総合領域

科研費の分科・細目：情報学・計算機システム・ネットワーク

キーワード：ハイパフォーマンスコンピューティング、GPU コンピューティング

1. 研究開始当初の背景

従来の CPU に加えて演算アクセラレータを備える複合型の高性能計算システムが急速に増加している。近年、演算アクセラレータとしてしばしば利用される Graphics Processing Unit(GPU)や Cell Broadband Engine(CBE)への各種アプリケーションの実装とその性能評価が盛んに報告されており、そのような報告の多くでプログラミング

の煩雑さとプログラムの負担による最適化の重要性が指摘されている。このため、GPU や CBE などの存在を意識することなくそれらのすぐれた演算能力を利用するための開発環境が研究されている。

しかし、GPU や CBE は従来の CPU とは異なる特性や性能を持ち、その結果として処理の得手不得手がある。適切なプロセッサの選択はプログラム実行時に定まるパラメー

タにも強く依存しているため、それらのパラメータが定まるプログラム実行以前に、プログラマやコンパイラが各処理に対して適切なプロセッサを指定するのは困難である。このように深刻な問題であるにも関わらず、複合型計算システムの異種複数のプロセッサから適切なプロセッサを実行時に自動選択するための研究は国内外を含めてこれまでほとんどなされていない。

この問題に対して、本研究代表者らは同一の記述から CPU および GPU 向けのコードを自動生成し、各処理を担当するプロセッサを実行時に自動選択する機能を有するフレームワークとして **Stream Programming with Runtime Auto-Tuning (SPRAT)** をこれまでに提案してきた。しかし、SPRAT がプロセッサを自動選択できるのはストリーム処理として記述されたプログラムの一部のみに対してであり、ストリーム処理として記述することが困難/不可能な処理に対するプロセッサ自動選択はできない。

2. 研究の目的

本研究計画の目的は、複合型高性能計算システムに搭載された異種複数のプロセッサを自動的にかつ効果的に使い分ける機能を有する高性能計算フレームワークを構築することである。異種複数のプロセッサで動作するプログラムを統一的に記述するために本研究代表者らが提案している高水準プログラミング環境を拡張し、さらに高性能計算で多用されるライブラリ等をその言語中からシームレスに利用可能とすることで、プログラマが意識することなく複合型計算システム中の異種複数のプロセッサを適材適所で自動的に活用できるフレームワークの構築を目指す。また、特に消費電力あたりの演算性能に着目してプロセッサを使い分けることを重要課題とし、アプリケーション実行に要する消費エネルギーの大幅な削減を達成する。

(1) ドメイン特化型言語の設計と言語処理系の実現

ストリーム処理以外の科学技術計算で頻出する処理や並列処理を記述するために、新たなプログラミングモデルとそれを簡便に記述するための文法を定義する。また、CPU と GPU を含む複数のプロセッサ向けのコードを自動生成する言語処理系を実装し、抽象化が適切であることを明らかにする。

(2) 異種複数のプロセッサに自動的に適切に割り当てる実行時環境の構築

実行時性能予測に基づくプロセッサ自動選択機能を有する実行時環境を構築する。その実現のために、各プロセッサ性能の適切なモデル化や、効率よく的確に性能パラメータを得るためのプロファイリング方法を明らか

にする。また、演算アクセラレータを搭載するノードとしないノードが混在する不均質なクラスタ内のタスクスケジューラと連動し、クラスタシステム全体としての実行時間や消費エネルギーの最小化を実現する。

3. 研究の方法

平成 21 年度は、異種複数のプロセッサの存在を意識することなく、科学技術計算で求められる各種ライブラリ(メッセージパッシング通信、線形代数、高速フーリエ変換など)をシームレスに利用可能なドメイン特化型言語の設計とその言語処理系、および実行時環境の開発を行う。このためにはストリーム処理以外の、より広範な処理に対する各プロセッサの性能予測が必要であり、性能予測方法やその予測のための性能パラメータ取得方法の確立が求められる。これらの検討により、各種ライブラリを使って所望する計算の大部分を記述するという科学技術計算の分野で一般的な開発スタイルで、CPU と CUDA 対応の GPU を含む異種複数のプロセッサを搭載する複合型計算システムにおいてプロセッサの実行時自動選択機能を利用可能とする。

平成 22 年度は、平成 21 年度に得られた研究成果をさらに発展させるとともに、タスクスケジューラとの連携によって大規模高性能計算システムのエネルギー効率の改善を目指す。また、構築された高性能計算フレームワークの有効性および実用性を実アプリケーションに基づいて検証する。このために、動作中のプロセスの状態を保存し、他の計算資源上でリスタートする機能(チェックポイント/リスタート)により、動的ジョブスケジューリング機能を実現する。

4. 研究成果

CPU 向けの線形代数ライブラリ (Basic Linear Algebra Subroutines, BLAS) と GPU 向けの BLAS ライブラリを自動的に使い分けるための性能予測方法、および性能パラメータ取得方法を検討し、CPU と GPU がお互いに得意な処理の実行のみを担当する実行時プロセッサ選択機付き BLAS ライブラリを実現した。同ライブラリは本研究代表者らが提案している高水準プログラミング言語 SPRAT からシームレスに利用可能となっており、異種複数のプロセッサを適材適所で活用できるように設計されている。評価の結果、問題サイズと性能の関係が各プロセッサで大きく異なることが示され、適材適所の利用の必要性が明らかになった。特徴的な事例として、BLAS 関数の一つである `sger` に対する CPU, GPU (2 種類)、および CBE のデータサイズに対する性能変化の様子を図 1 に示す。

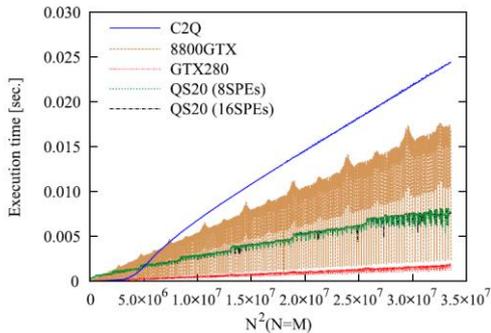


図1 各種プロセッサの性能特性の違い

また、現在 GPU の計算能力を利用するために事実上の標準の開発環境として利用されている CUDA (Compute Unified Device Architecture) では、ハードウェアを意識していくつかのパラメータを試行錯誤で設定しなければならない。したがって、GPU の存在を意識することなくその計算能力を利用するためには、それらのパラメータ設定を自動化する必要があるため、パラメータ自動チューニング機能を検討し、高水準プログラミング環境 SPRAT への実装と性能評価を行った。

さらに、異種複数のプロセッサ向けのプログラミング環境として OpenCL が利用可能となったことから、それらを活用するための検討も行った。現在最も標準的に使われている CUDA と OpenCL との性能の違いの原因を調査し、OpenCL においても CUDA と同等の最適化手法を適用する(図2)ことにより、CUDA と同等の性能を達成できることが示された(図3)。CUDA と OpenCL の性能差の主要因を明らかにしたこの研究成果は、情報処理学会山下記念賞を受賞するなど、学術的に高く評価された。

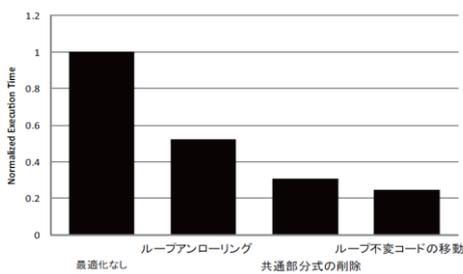


図2 各種手動最適化による性能の変化

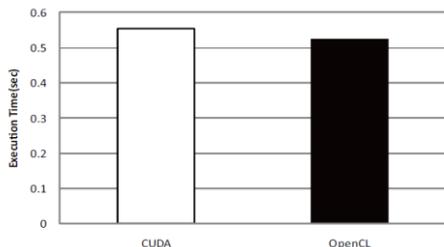


図3 CUDA と OpenCL の性能差

また、OpenCL で記述されたアプリケーション各種パラメータが性能に与える影響を定量的に評価するとともに、その主要計算部分(カーネル)の性能を予測する手法も検討した。その結果、演算アクセラレータの種類に依存して最適なパラメータ設定が変化することを明らかにし、性能と移植性を両立するためにはパラメータ設定の自動化が重要であることを示した。また、カーネルの実行時間はカーネルに与えられる引数等に依存して変化するが、引数が実行時間に与える影響にはいくつかの種類が存在するために、引数を影響の種類毎に分類して考慮することで、単純な重回帰分析と比較して高精度の性能予測ができることを明らかにした。

今後、高性能計算システムはさらに大規模化・複雑化し、膨大な数の計算ノードから構成されるため、そのディペンダビリティの向上が極めて重要である。このため本研究課題では GPU コンピューティングにおけるディペンダビリティを高めるための検討も行い、CUDA アプリケーションや OpenCL アプリケーション向けチェックポイントリスタート機能を実装・評価した。その結果、それらのアプリケーションの障害からの復帰やプロセスマイグレーションの機能を実現することが可能であることが示された。

複合型計算システムの性能を活用するアプリケーションとして流体シミュレーションに着目し、その高速化を実現する実装方法についても検討した。

さらに、複合型計算システム向けのジョブスケジューリング手法を提案した。その結果、CPU と GPU のいずれかの使用頻度が低く、遊休資源化している場合に、その遊休資源を有効活用することによって、システムにジョブが投入されてからその実行完了までの時間(ターンアラウンドタイム)の期待値を短縮できることが明らかになった。

また、本研究課題の一部として評価していたコンパイラ指示行に基づくプログラミング環境の実用性評価結果を国内研究会にて発表した。

5. 主な発表論文等

(研究代表者、研究分担者及び連携研究者には下線)

[雑誌論文] (計 6 件)

1. K. Sato, K. Komatsu, H. Takizawa, and H. Kobayashi, "A History-based Performance Prediction Model with Profile Data Classification for Automatic Task Allocation in Heterogeneous Computing Systems," The

9th International Symposium on Parallel and Distributed Processing with Applications (ISPA2011), CD-ROM, May 2011.

2. H. Takizawa, K. Koyama, K. Sato, K. Komatsu, and H. Kobayashi, "CheCL: Transparent Checkpointing and Process Migration of OpenCL Applications," The 25th IEEE International Parallel and Distributed Processing Symposium (IPDPS2011), CD-ROM, May 2011.
3. K. Komatsu, T. Soga, R. Egawa, H. Takizawa, H. Kobayashi, S. Takahashi, D. Sasaki, and K. Nakahashi, "Parallel Processing of the Building-Cube Method on the GPU Platform, Computers&Fluids, Vol. 45, No. 1, pp. 122-128, 2011.
4. K. Sato, H. Takizawa, K. Komatsu and H. Kobayashi, "Automatic Tuning of CUDA Execution Parameters for Stencil Processing," Software Automatic Tuning: From Concepts to State-of-the-Art Results, pp. 209-227, 2010.
5. K. Komatsu, K. Sato, Y. Arai, K. Koyama, H. Takizawa, and H. Kobayashi, "Evaluating Performance and Portability of OpenCL Programs," The 5th International Workshop on Automatic Performance Tuning, USB, June 2010.
6. H. Takizawa, K. Sato, K. Komatsu and H. Kobayashi, "CheCUDA: A Checkpoint/Restart Tool for CUDA Applications," in Proceedings of International Conference on Parallel and Distributed Computing Applications and Technologies (PDCAT2009), USB, Dec 2009.

[学会発表] (計 5 件)

1. 菅原誠, 佐藤功人, 小松一彦, 滝沢寛之, 小林広明, "プログラム自動生成技術に基づく GPU コンピューティングの性能評価," 並列/分散/協調処理に関するサマー・ワークショップ(SWoPP2011), 2011年7月27日, 鹿児島.
2. H. Takizawa, "Towards OpenCL for SX," 13th Teraflop Workshop, 2010年10月

22日, Sendai.

3. 荒井勇亮, 佐藤功人, 滝沢寛之, 小林広明, "OpenCLによるGPUコンピューティングの性能評価," 第124回HPC研究会, 2010年2月23日, 熱海.
4. 滝沢寛之, "新アーキテクチャへのアプローチ," 自動チューニング技術の現状と応用に関するシンポジウム, 2009年10月22日, 東京.
5. 小松一彦, 小山賢太郎, 佐藤功人, 滝沢寛之, 小林広明, "プロセッサ自動選択機能を有するBLASの実現に向けた性能評価," FIT2009 第8回情報科学技術フォーラム, 2009年9月4日, 仙台.

6. 研究組織

(1) 研究代表者

滝沢 寛之 (TAKIZAWA HIROYUKI)

東北大学・大学院情報科学研究科・准教授

研究者番号: 70323996