様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成　３０　年　　６　月　１４　日現在

機関番号：　１２６０１
研究種目：　研究活動スタート支援
研究期間：　2016～2017
課題番号：　１６Ｈ０６６７９

研究課題名（和文）アプリケーションのデータ構造に着目したメニーコア向け自動最適化フレームワーク

研究課題名（英文）Auto-tuning Framework Focusing on Application Data Structure for Many-core Processors

研究代表者

星野　哲也（HOSHINO, Tetsuya）

東京大学・情報基盤センター・助教

研究者番号：４０７７５９４６

交付決定額（研究期間全体）：（直接経費）　　2,300,000円

研究成果の概要（和文）：近年増加傾向にあるメニーコアプロセッサを用いた計算環境において、その性能を引き出すためにはVector Processing Unit (VPU)を効率良く利用することが重要である。しかし、VPUの効率的な利用にはハードウェアやコンパイラに関する知識が必要であり、またプログラムのデータ構造の変更などが往々にして必要となる。
本研究では、データ構造を抽象化するためのコンパイラ指示文の提案と、その指示文を解釈するトランスレータの開発、自動ベクトル化を促進するフレームワークデザインの提案と、そのデザインに則った境界要素法向けのフレームワークの開発を行った。

研究成果の概要（英文）：Nowadays, the number of computational environment using many-core processors is increasing. To bring out the efficient performance of many-core processors, it is important to efficiently use the Vector Processing Unit (VPU). However, the knowledge of hardware and compiler is required to efficiently use the VPU, and moreover, data structural changes are often required.
In this research, we propose a set of compiler directives for abstraction of data layout. We also implement a translator for the proposed directives. Furthermore, we propose a framework design to enhance the efficient vectorization. Also, we implement a BEM-BB framework using the proposed framework design.

研究分野：　高性能計算

キーワード：　メニーコアプロセッサ　GPU　SIMD

１．研究開始当初の背景

地震や気象予測、自動車や航空機の設計など、多数のパラメータからなる複雑な現象の解析を行うために、多数の模型を製作し実験を行うのは困難であり、今日では大規模な計算環境を用いたシミュレーションと併せて解析することが一般的となっている。これらシミュレーションを行うためのアプリケーションは多数存在するが、その多くは CPU(スカラプロセッサ)向けに作られたものであり、現在一般的となりつつあるメニーコアプロセッサを導入した計算環境への対応が課題である。メニーコアプロセッサはスカラプロセッサと異なる性能特性を持ち、十分な性能を得るためには往々にしてアルゴリズムの見直し、最適化、再実装が必要となるためである。

この問題を解決するために数多くの研究が行われてきたが、特に OpenMP や OpenACC と言った指示文ベースのプログラミングモデルが注目されている。指示文ベースのプログラミング規格モデルでは、既存のアプリケーションに指示文を挿入することで指定された領域を並列化し、メニーコアプロセッサを利用できる。また、指示文をサポートしていないコンパイラでは指示文は無視され、元のプログラムと同様に実行可能であるため、ソースコードの保守性、異なるデバイス間での可搬性に優れている。しかし、メニーコアプロセッサ向けのプログラミングモデルとして広く使われている CUDA などと比較して、指示文ベースのプログラミングモデルには記述の自由度に制限があり、その影響についての評価も十分ではなかった。

応募者らは以前の研究において、OpenACC の記述の自由度の低さがアプリケーションの性能に及ぼす影響を評価するために、宇宙航空研究開発機構により研究開発されている UPACS に OpenACC・CUDA を適用し、比較評価を行った。この結果、OpenACC の機能的な制限(スレッド間同期がない、スレッド ID を取得できない)が性能ギャップの要因となること、またプロセッサの性能特性の違いに起因され、最適なデータ構造がプロセッサ毎に異なることを確認した。

プロセッサ毎に得意とするデータ構造が異なることはよく知られているが、OpenACC は指示文ベースであるという特性上、元とするプログラムが同一であり、また計算環境によらず実行できる可搬性が長所であるため、再実装を必要とする CUDA などと比較して特に問題になる。そこで応募者らはこの問題を解決する、自動並列化・最適化フレームワークの研究開発を、日本学術振興会の支援の元、「科学技術アプリケーションのメニーコア環境を支援する自動最適化フレームワーク」の題目で特別研究員(DC1)として進めてきた。標準仕様である OpenACC をベースとして拡張し、特にプログラムのデータ構造の実行プロセッサ向けの最適化に特化したフレームワークの開発を目指したものである。

２．研究の目的

本研究の目的は、メニーコアプロセッサの備える、比較的長いベクトル長を持つ Vector Processing Unit (VPU)を、簡単かつ効率的に使う手法をユーザに提供することにある。本研究のタイトルにあるデータ構造は、VPU の効率的利用に密接に関係している。

メモリアクセスはキャッシュライン単位で行われ、一度のメモリアクセスでキャッシュラインサイズの倍数アドレスから始まるキャッシュラインサイズ分のデータを取得する。図１は、キャッシュラインサイズが 32Byte のプロセッサにおける、典型的なメモリアクセスパターンを示したものである。上段のパターンは最も効率が良く、1 度のメモリアクセスで済む。中段のパターンでは、メモリアクセスが倍数アドレスから始まっておらず、キャッシュライン２つ分のメモリアクセスを必要とする。下段のパターンは最も効率が悪く、本来計算に必要であるデータ量の４倍の領域へのメモリアクセスを要する。アクセスパターンによる性能への影響はキャッシュラインサイズが大きい程大きくなるが、メニーコアプロセッサのキャッシュラインサイズは 64Byte (Intel Xeon Phi KNL など), 128Byte (NVIDIA GPU など)と比較的大きいため、よりメモリアクセスパターンの最適化が重要となる。この最適化のためには、往々にしてプログラムのデータ構造の見直しが必要となる。

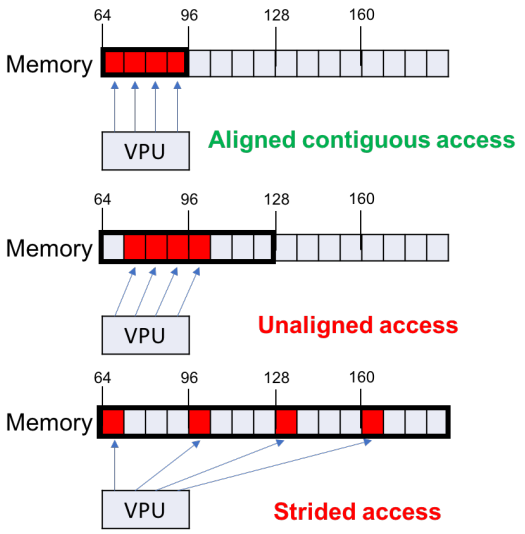

図 1　VPU の典型的なメモリアクセスパターン。赤く塗られた部分が VPU での計算に必要なデータを示し、太線で囲まれた部分が実際にアクセスされる範囲を示す。

さらに、VPU の利用のためには、プロセッサの命令セットに対応する intrinsics と呼ばれる組み込み関数を用いた命令レベルの並列化や、コンパイラ指示文を用いた半自動並列化が必要となるが、それらの利用にはハードウェアやコンパイラの知識が必要とされ、様々な困難を伴う。

本研究では、これらの困難さをユーザプログラムと切り離し、簡便に利用する手法をユーザに提供することを目的としている。

３．研究の方法

本研究では、

(1) 指示文ベースのプログラミングモデル OpenACC の拡張
(2) Domain specific なフレームワークを用いた手法

の２つのアプローチで、VPU を効率的に利用するための手法の開発を行った。

４．研究成果

(1) 指示文ベースのプログラミングモデル OpenACC の拡張

上記について概要を示す。詳細については、論文③を参照していただきたい。

本研究では、VPU の効率的な利用に大きく関わる、データレイアウトを抽象化するための指示文を新たに提案した。

参照実装として、提案した指示文を含む拡張 OpenACC を解釈する source-to-source のトランスレータを実装し、実際のアプリケーションを用いて評価を行った。

図２は、CCS-QCD と呼ばれるアプリケーションを用いての GPU 上での性能評価の結果である。アプリケーションプログラム中に現れる 17 の配列に対し、拡張指示文を用いてデータレイアウトを GPU 向けに変換する。図の一番左がベースラインであり、2 番目が手動によりデータ構造を書き換えた際の実行時間、3 番目がトランスレータによるデータ構造の変更による実行時間である。トランスレータによるデータ構造の変換では、実行時に変換するためのオーバーヘッドが発生するために、オーバーヘッドのない手動の変更と比較して低速であるが、数パーセントの性能向上が得られている。

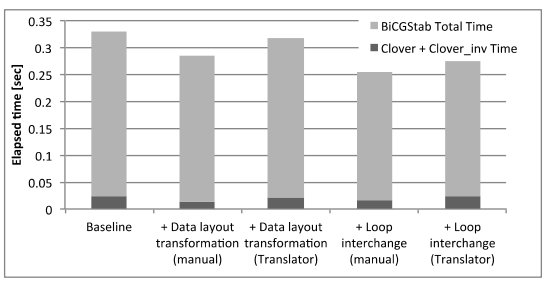

図 3　CCS-QCD の実行時間

さらに、変換したデータ構造に合わせて、プログラムのループ順を手動・トランスレータそれぞれで変更したものが 4・5 番目の実行時間である。これにより、手動による書き換えと比較すると 92.3%の実行性能ではあるが、ベースラインと比較すると 120.7%の性能が得られている。手動による変更ではプログラム全体を書き換える必要があるが、拡張指示文による変更は指示文数行の挿入により実現できるため、本提案の有効性が示されている。

参照実装であるトランスレータは最適化が不十分な点もあり、今後の課題である。

(2) Domain specific なフレームワークを用いた手法

上記について概要を示す。詳細については、論文②を参照していただきたい。

本研究では、電磁場解析などでよく使われる計算手法である、境界要素法（Boundary Element Method, BEM）向けのフレームワークである BEM-BB を元に、オリジナルではサポートされていない VPU の利用を促進する拡張を行った。

VPU は Single Instruction Multiple Data (SIMD)と呼ばれる、命令レベルの並列実行により利用されること一般的だが、本フレームワークでは演算内容はユーザが記述するため、並列化すべき命令は事前にはわからない。このような実装は Domain specific なフレームワークの設計として一般的であるが、命令レベルの並列化のサポートは容易ではない。

本研究における提案は、コンパイラによる自動ベクトル化を促進するためのフレームワークの設計である。本来自動ベクトル化にはコンパイラ指示文に関する知識が必要だが、ユーザに知識がなくとも、フレームワークに従ってプログラムを書くことにより、自動並列化可能なプログラムとなる。

図３は、提案の設計を 2 種類の静電場解析プログラムにより、Intel Broadwell (BDW) と Intel Xeon Phi KNL(KNL)上で評価したものである。SIMD design が提案の実装を用い

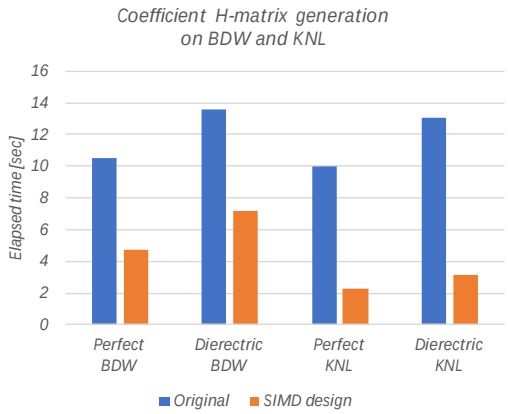

図 2　フレームワークによる静電場解析の高速化

た際の実行時間である。特に KNL はベクトル化の恩恵が大きく、オリジナルの実装と比較して 4 倍程度の性能向上が得られている。
今後は、本設計の有効性を示すために、他のフレームワークへの適用を試みる。

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 3 件）
①【査読有】Y. Nomura, I. Sato, T. Hanawa, S. Hanaoka, T. Nakao, T. Takenaga, D. Sato, T. Hoshino, Y. Sekiya, S. Ohshima, N. Hayashi, O. Abe, Preliminary development of training environment for deep learning on supercomputer system, 32nd International Congress and Exhibition on Computer Assisted Radiology (CARS 2018), June 2018 (in press).

②【査読有】Tetsuya Hoshino, Akihiro Ida, Toshihiro Hanawa, and Kengo Nakajima: Design of Parallel BEM Analyses Framework for SIMD Processors, International Conference on Computational Science (ICCS 2018), June 2018 (in press).

③【査読有】T. Hoshino, N. Maruyama and S. Matsuoka, ”A Directive-Based Data Layout Abstraction for Performance Portability of OpenACC Applications,”2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 1147-1154.

〔学会発表〕（計 16 件）
① Tetsuya Hoshino, Akihiro Ida, Toshihiro Hanawa, and Kengo Nakajima,“ Performance Evaluations and Optimizations of H-Matrices for Many-Core Processors”,SIAM Conference on Parallel Processing for Scientific Computing 2018(SIAM PP 18), Tokyo, Japan (March 2018).

② 中島 研吾, 星野 哲也, 成瀬 彰, 塙 敏博, 三木 洋平：有限要素法における係数行列生成部 のマルチコア・メニィコア向け最適化, 情報処理学会研究報告, 2018-HPC-163(28),pp.1-8, 2018 年 2 月.

③ 塙 敏博, 伊田 明弘, 星野 哲也：OpenCL を用いた FPGA による階層型行列計算, 情報処理 学会研究報告, 2018-HPC-163(26), pp. 1-8, 2018 年 2 月.

④ 野村 行弘, 佐藤 一誠, 塙 敏博, 花岡 昇平, 中尾 貴祐, 竹永 智美, 佐藤 大介, 星野 哲也, 関谷 勇司, 大島 聡史, 林 直人, 阿部 修, スーパーコンピュータ上での Deep Learning 学習 環境の初期構築, 電子情報通信学会技術研究報告, Vol.117, no. 281, MI2017-47, pp.1-2, 2017 年 11 月.

⑤ 塙 敏博, 伊田 明弘, 星野 哲也: 階層型行列計算の FPGA への適用, 情報処理学会研究報告, 2017-HPC-161(10), pp. 1-10, 2017 年 9 月.

⑥ 星野哲也,伊田明弘,塙敏博,中島研吾: 階層型行列法ライブラリ HACApK を用いたアプリ ケーションのメニーコア向け最適化, 情報処理学会研究報告, 2017-HPC-160(15), pp. 1-10, 2017 年 7 月.

⑦ Tetsuya Hoshino, Satoshi Ohshima, Toshihiro Hanawa, Kengo Nakajima, Akihiro Ida: Pascal vs KNL: Performance Evaluation with ICCG Solver, Research Poster Session, ISC High Performance 2017 (Frankfurt, Germany, June 18-22, 2017) (ポスター発表)

⑧ 塙敏博,星野哲也,中島研吾,大島聡史, 伊田明弘: GPU 搭載スーパーコンピュータ ReedbushH の性能評価, 情報処理学会研究報告, 2017-HPC-159(9), pp. 1-6, 2017年4月.

⑨ 星野哲也、大島聡史、塙敏博、中島研吾、伊田明弘、OpenACC を用いた ICCG 法ソルバーの PascalGPU における性能評価、情報処理学会研究報告（2017-HPC-158-18)、情報処理学会第 158 回 HPC 研究会（熱海、2017 年 3 月 8 日~10 日）

⑩ 塙敏博、中島研吾、大島聡史、星野哲也、伊田明弘、Xeon Phi+OmniPath 環境における OpenMP, MPI 性能最適化、情報処理学会研究報告（2017-HPC-158-21)、情報処理学会第 158 回 HPC 研究会（熱海、2017 年 3 月 8 日~10 日）

⑪ Tetsuya Hoshino, Naoya Maruyama, Satoshi Matsuoka: A Directive-based Data Layout Autotuning for OpenACC Applications, 2017 SIAM Conference on Computational Science and Engineering (SIAM CSE17) (Atlanta, GA, USA, February 27-March 3, 2017)

⑫ 塙敏博、中島研吾、大島聡史、星野哲也、伊田明弘、パイプライン型共役勾配法の性能評価、情報処理学会研究報告（2016-HPC-157-6)、情報処理学会第 157 回 HPC 研究会（那覇、2016 年 12 月 21 日~22 日）

⑬　中島研吾、大島聡史、塙敏博、星野哲也、伊田明弘、ICCG 法ソルバーの Intel Xeon Phi 向け最適化、情報処理学会研究報告（2016-HPC-157-16)、情報処理学会第 157 回 HPC 研究会（那覇、2016 年 12 月 21 日~22 日）

⑭　塙敏博、中島研吾、大島聡史、伊田明宏、星野哲也、田浦健次朗、データ解析・シミュレー ション融合スーパーコンピュータシステム Reedbush-U の性能評価、情報処理学会研究報告 （2016-HPC-156-10)、情報処理学会第 156 回 HPC 研究会（小樽、2016 年 9 月 15 日~16 日）

⑮　星野哲也、丸山直也、松岡聡：データレイアウト最適化指示文による OpenACC アプリケー ションの高速化、情報処理学会研究報告（2016-HPC-155-32)、2016 年並列／分散／協調処 理に関する 『松本』サマー・ワークショップ（SWoPP 松本 2016）（松本、2016 年 8 月 8 日 ~10 日）

⑯　Tetsuya Hoshino, Satoshi Matsuoka: Acceleration of a Compressed Flow Analysis Application with OpenACC, HPC in Asia Session, ISC High Performance 2016 (Frankfurt, Germany, June 19-23, 2016) (ポスター発表)

〔その他〕
ホームページ等

６．研究組織
(1)研究代表者
星野　哲也　（HOSHINO, Tetsuya）
東京大学・情報基盤センター・助教
研究者番号：40775946