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

令和　２ 年　 ６ 月 １０ 日現在

機関番号：８２４０１
研究種目：若手研究(B)
研究期間：2017～2019
課題番号：１７Ｋ１２６９４

研究課題名（和文）ベクトル長agnosticなアーキテクチャのためのSIMD IRコード生成技法

研究課題名（英文）SIMD IR Code Generation for Vector-Length Agnostic Architectures

研究代表者
Ｌｅｅ　Ｊｉｎｐｉｌ（LEE, JINPIL）

国立研究開発法人理化学研究所・計算科学研究センター・特別研究員

研究者番号：３０７６４８７３

交付決定額（研究期間全体）：（直接経費）　　2,800,000円

研究成果の概要（和文）：本研究では特定のベクトル幅を前提とせずに並列性を表現する中間言語の設計や、それを用いたSIMD命令生成の研究を行う。中間言語を特定のベクトル幅や命令セットと独立させることで様々なアーキテクチャで利用可能になる。その結果、Intelのような固定ベクトル長を持つ命令セットだけでなく、ARMのSVE命令セットやFPGAのようなデータを連続したストリームとして扱うアーキテクチャのSIMDコードを生成することができるようになった。プログラミングモデルとしてOpenMPを採用し、SIMD指示文に対する独自拡張とタスク機能を用いて、処理系の中で提案手法の中間言語に変換することでSIMD化を実現している。

研究成果の学術的意義や社会的意義
既存のベクトル命令は固定ベクトル長を前提としており、命令セット毎にその幅が異なる。最新のARMやRISC-Vアーキテクチャにはコードにベクトル長を明記せず、ランタイムや実行アーキテクチャによってベクトル幅が決まるアプローチが採用されている。またFPGAのような再構成可能なハードウェア上で計算カーネルを実装するときも性能最適化のためにベクトル化を行うが、ハードウェア資源の制約によってその幅は流動的に変化する。本研究で提案した中間形式を用いることによって、様々なベクトル長を持つアーキテクチャを統一されたアルゴリズムで開発することができるようになる。

研究成果の概要（英文）：In this study, we propose an intermediate representation form called SIMD IR to express parallelism without assuming a specific vector width and generate SIMD instructions in our compiler. We made SIMD IR independent from specific vector widths and instruction sets for a variety of architectures. As a result, the compiler can handle not only Intel's vector instruction set, but also ARM's SVE or FPGAs that treat data as a continuous stream. We extended the OpenMP SIMD and task directive so that the compiler can use SIMD IR to generate SIMD code for various architectures.

研究分野：高性能計算

キーワード：SIMD並列化　コンパイラ最適化　高性能計算

※科研費による研究は、研究者の自覚と責任において実施するものです。そのため、研究の実施や研究成果の公表等については、国の要請等に基づくものではなく、その研究成果に関する見解や責任は、研究者個人に帰属されます。

様　式　Ｃ－１９、Ｆ－１９－１、Ｚ－１９（共通）

１．研究開始当初の背景

(1) ベクトル化とは 1 命令で処理されるデータの幅を増やすことで計算のスループットを向上させる最適化技術である。既存の CPU アーキテクチャはベクトル幅が固定された命令セットを採用しており、アーキテクチャ毎に異なる幅を持つ。ベクトル化コンパイラは様々なベクトル幅を持つアーキテクチャに対して個別に対応しなければならなかった。

(2) 半導体資源と消費電力の制約の下で最大限の性能を達成するために、各ベンダーはベクトル幅を増やすことでスループット性能を上げており、ARM を含む一部のアーキテクチャは特定のベクトル幅を前提としない命令セットを提案している。

(3) ムーアの法則の追従が困難になり、従来の汎用 CPU アーキテクチャから Field-Programmable Gate Array (FPGA)などの再構成可能なハードウェアによるアプリケーション特化のアーキテクチャを利用することがトレンドになっている。このようなアーキテクチャでもベクトル化による性能最適化を行うが、そのデータ幅は対象アプリケーションやハードウェア資源の制約により流動的に変化する。

２．研究の目的

(1) 本研究の目的は様々なベクトル長を持つアーキテクチャに対して統一されたアルゴリズムで性能最適化を行うための基盤技術を研究開発することである。性能最適化とはユーザが記述された抽象度の高いアプリケーションコードから SIMD 命令を生成することである。

(2) もう 1 つの目的は提案した性能最適化技術をより簡単に利用できるプログラミングモデルの提案である。SIMD 命令セットを利用するためには intrinsic 関数やコンパイラによる自動並列化を用いるが、本研究では OpenMP の SIMD 指示文を拡張して明示的な並列化記述によるプログラミング環境の改善を目指す。

３．研究の方法

(1) 特定のベクトルを明示的に含まない中間言語 SIMD IR を設計する。コンパイラはオープンソースソフトウェアである LLVM を利用し、LLVM IR と SIMD IR 間のコード変換を行うソフトウェアを実装する。ベクトル化などの性能最低化は SIMD IR 上で行われる。

(2) 実装されたコンパイラの性能を評価するために FPGA を用いる。FPGA 上で任意の計算ハードウェアを実現する際に性能を向上させるためにベクトル化を行うが、そのベクトル幅はユーザによって最適な値を指定するか、メモリバンド幅や FPGA の資源を考慮して自動的に決めることが可能である。

４．研究成果

(1) LLVM の中間言語である LLVM IR は様々なアーキテクチャのアセンブリコードを生成するために用いられる。特定のアーキテクチャに依存しない言語構文を定義しているが、ベクトル長は固定（研究開始時は固定ベクトル長のみであったが、それ以降可変ベクトル長をサポートするようになった）であるため、LLVM IR に変換された時点で扱うベクトル長が決まってしまう。本研究では LLVM の外部プロジェクトである Polly を用いて独自の中間言語 SIMD IR を定義した。Polly は整数計画法の理論に基づいてループ文の解析を行い、ベクトル化や GPU 並列化を行う。SIMD IR は Polly から得られる情報をもとに独立して実行可能なループイテレーションや最適な実行順序を計算する。実態は Polly の内部形式である SCoP と LLVM IR に追加情報を付与することで実装されている。特定のベクトル長を前提としないため、ループ文の本体を表す LLVM の BasicBlock は SIMD 化を避け、scalar 型のみで表現する。

(2) SIMD IR を用いたコード生成をテストするために FPGA 向けコンパイラを実装した。C2SPD は C 言語で作成されたソースコードを FPGA 向け Domain Specific Language (DSL)に変換するコンパイラである。図 1 に C2SPD によるコード変換の流れを示す。C 言語コードは LLVM の C 言語フロントエンドの Clang によって LLVM IR に変換され、提案手法によって SIMD IR に変換される。コンパイラは SIMD IR に記述された並列可能なループイテレーションを FPGA 上で実行できるように DSL コードを生成する。本研究では理化学研究所計算科学研究センターのプロセッサ研究チームが開発を行っている SPGen を FPGA のコード生成の DSL と

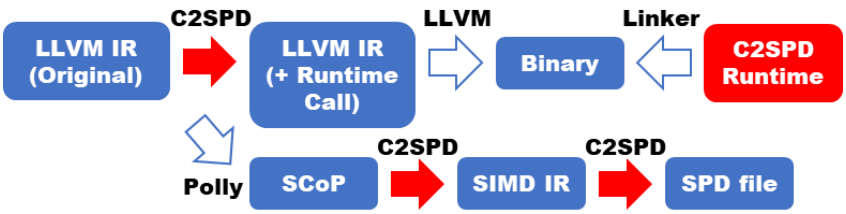

図 1. C2SPD によるコード変換

して用いている。

(3) C2SPD は C 言語で記述されるが、性能最適化のために一部の情報をユーザが提供することができる。図 2 に C2SPD で用いられる C 言語の拡張構文を示す。ソースコードの中の最外ループは region 指示文によって指定されているため、その中の実行文が FPGA 上に展開される。offload 指示文で指定されたループ文が 1 つのモジュールになり、モジュール同士はバスで接続される。FPGA における最適化はモジュール同士のデータバスの幅を増やすか、region 指示文のイテレーションを複製して FPGA に展開することによってカスケード接続を実現することである。どちらの手法もスループット性能を向上させるものであるが、安全に実行するためには各イテレーションが独立に実行されることが保証されなければならない。本研究では SIMD IR による並列性の解析によってこの問題を解決している。

```
float old[M][N];
float new[M][N];

#pragma spd region    // Top Module
for (int t = 0; t < TIME_STEP; t++) {
#pragma spd offload   // Module 1
  for (int i = 1; i < M-1; i++)
    for (int j = 1; j < N-1; j++)
      new[i][j] = (old[i][j-1] + old[i][j+1]
                 +  old[i+1] + old[i-1][j]) * 0.25;

#pragma spd offload   // Module 2
  for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
      old[i][j] = new[i][j];
}
```

図 2. C2SPD の言語構文

(4) 図 3 に 2 次元ステンシルコードによる性能評価の結果を示す。VL はデータバスの幅を示すパラメータであり、UC はイテレーションの複製によるカスケード接続の段数を示す。両方のパラメータを 16 にしたとき、計算モジュールは 256 倍に複製されると考えることができる。実行結果は逐次実行に比べて 220 倍改善されており、FPGA 資源の割り当てに見合った性能改善が得られている。

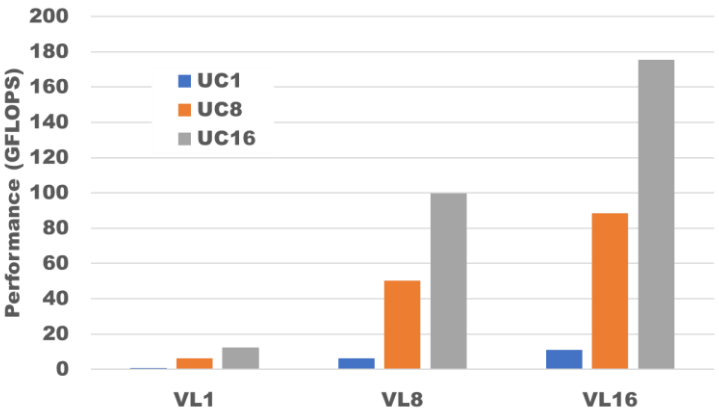

図 3. C2SPD の性能評価結果

(5) FPGA ではプログラミングモデルとして独自の指示文を用いているが、CPU 向けの並列化は OpenMP が主流であるため、OpenMP の SIMD 並列化を支援する独自拡張の提案を行った。ARM 社の高性能計算向けベクトル命令セットである Scalable Vector Extension (SVE)はベクトル長を前提としないアプローチをとっている。しかし、OpenMP の SIMD 指示文で SVE でベクトル化されたコードを呼ぶインターフェースが存在しないため、本研究で言語拡張 alias simd 指示文の提案を行った。alias simd 指示文はベクトル化された関数に与えることができ、Intel の AVX 命令セットのような固定ベクトル長だけでなく、ARM SVE のようなベクトル長が可変なものも対象にすることができる。指定された関数は simd 指示文によって並列化されたループ文で利用することができるため、明示的なベクトル化を実現する。図 4 に性能評価の結果を示す。シミュレータによる性能結果から明示的なベクトル化が自動ベクトル化より高い性能を達成し、ベクトル幅が増えるほど性能差が顕著である。

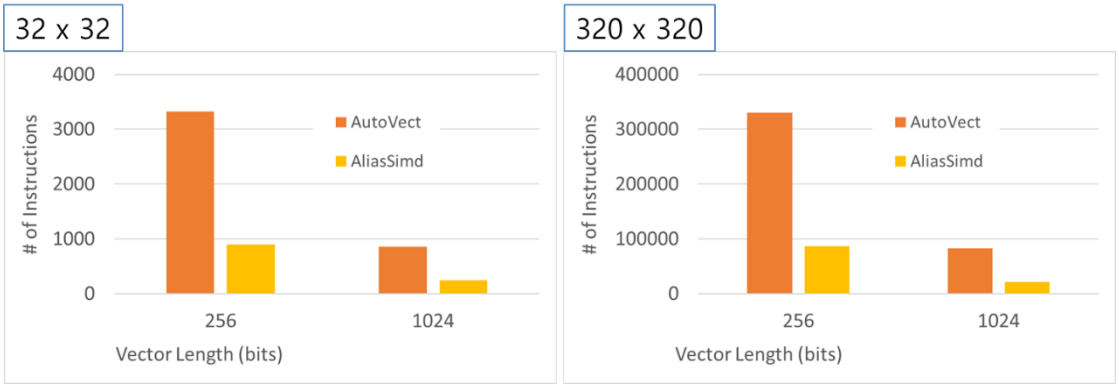

図 4. alias simd の性能評価結果

５．主な発表論文等

〔雑誌論文〕　計7件（うち査読付論文　3件／うち国際共著　2件／うちオープンアクセス　0件）

１．著者名 李 珍泌、上野 知洋、佐藤 三久、佐野 健太郎	４．巻 2018-HPC-165
２．論文標題 SPGenのC言語フロントエンドによるループ最適化と性能評価	５．発行年 2018年
３．雑誌名 研究報告ハイパフォーマンスコンピューティング	６．最初と最後の頁 1〜7
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 ストリーム計算ハードウェアコンパイラSPGenのためのPolyhedral Model を用いたループスケジュール最適化	４．巻 2018-HPC-167
２．論文標題 李 珍泌、上野 知洋、佐藤 三久、佐野 健太郎	５．発行年 2018年
３．雑誌名 研究報告ハイパフォーマンスコンピューティング	６．最初と最後の頁 1〜6
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Lee Jinpil、Ueno Tomohiro、Sato Mitsuhisa、Sano Kentaro	４．巻 5
２．論文標題 High-productivity Programming and Optimization Framework for Stream Processing on FPGA	５．発行年 2018年
３．雑誌名 Proceedings of the 9th International Symposium on Highly-Efficient Accelerators and Reconfigurable Technologies	６．最初と最後の頁 1〜6
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.1145/3241793.3241798	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Watanabe Yutaka、Lee Jinpil、Boku Taisuke、Sato Mitsuhisa	４．巻 11128
２．論文標題 Trade-Off of Offloading to FPGA in OpenMP Task-Based Programming	５．発行年 2018年
３．雑誌名 Lecture Notes in Computer Science	６．最初と最後の頁 96〜110
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.1007/978-3-319-98521-3_7	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

１．著者名 Lee Jinpil、Petrogalli Francesco、Hunter Graham、Sato Mitsuhisa	４．巻 10468
２．論文標題 Extending OpenMP SIMD Support for Target Specific Code and Application to ARM SVE	５．発行年 2017年
３．雑誌名 Lecture Notes in Computer Science book series	６．最初と最後の頁 62〜74
掲載論文のDOI（デジタルオブジェクト識別子） https://doi.org/10.1007/978-3-319-65578-9_5	査読の有無 有
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 李 珍泌、Francesco Petrogalli、Graham Hunter、佐藤 三久	４．巻 2017-HPC-160
２．論文標題 アプリに特化したSIMD最適化のためのOpenMP仕様拡張の提案とARM SVEを用いた評価	５．発行年 2017年
３．雑誌名 研究報告ハイパフォーマンスコンピューティング	６．最初と最後の頁 1〜8
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 該当する

１．著者名 李 珍泌、上野 知洋、佐藤 三久、佐野 健太郎	４．巻 2018-HPC-163
２．論文標題 HPC向けストリームプロセッサ生成のためのC言語フロントエンドの開発	５．発行年 2018年
３．雑誌名 研究報告ハイパフォーマンスコンピューティング	６．最初と最後の頁 1〜7
掲載論文のDOI（デジタルオブジェクト識別子） なし	査読の有無 無
オープンアクセス オープンアクセスではない、又はオープンアクセスが困難	国際共著 －

〔学会発表〕　計8件（うち招待講演　1件／うち国際学会　4件）

１．発表者名 李 珍泌、上野 知洋、佐藤 三久、佐野 健太郎
２．発表標題 SPGenのC言語フロントエンドによるループ最適化と性能評価
３．学会等名 第165回HPC研究発表会
４．発表年 2018年

１．発表者名
李 珍泌、上野 知洋、佐藤 三久、佐野 健太郎

２．発表標題
ストリーム計算ハードウェアコンパイラSPGenのためのPolyhedral Model を用いたループスケジュール最適化

３．学会等名
第167回HPC研究発表会

４．発表年
2018年

１．発表者名
Lee Jinpil、Ueno Tomohiro、Sato Mitsuhisa、Sano Kentaro

２．発表標題
High-productivity Programming and Optimization Framework for Stream Processing on FPGA

３．学会等名
The Ninth International Symposium on Highly Efficient Accelerators and Reconfigurable Technologies（国際学会）

４．発表年
2018年

１．発表者名
Watanabe Yutaka、Lee Jinpil、Boku Taisuke、Sato Mitsuhisa

２．発表標題
Trade-Off of Offloading to FPGA in OpenMP Task-Based Programming

３．学会等名
The 14th International Workshop on OpenMP 2018（国際学会）

４．発表年
2018年

１．発表者名
Lee Jinpil

２．発表標題
C/C++ Front-end for Streaming Processing on FPGAs

３．学会等名
The 2018 International Conference on Field-Programmable Technology Workshop RECONF-HPC（招待講演）（国際学会）

４．発表年
2018年

１．発表者名
Jinpil Lee、Francesco Petrogalli、Graham Hunter、Mitsuhisa Sato

２．発表標題
Extending OpenMP SIMD support for target specific code and application to ARM SVE

３．学会等名
13th International Workshop on OpenMP (IWOMP 2017)（国際学会）

４．発表年
2017年

１．発表者名
李 珍泌、Francesco Petrogalli、Graham Hunter、佐藤 三久

２．発表標題
アプリに特化したSIMD最適化のためのOpenMP仕様拡張の提案とARM SVEを用いた評価

３．学会等名
2017年並列／分散／協調処理に関する『秋田』サマー・ワークショップ（SWoPP2017）

４．発表年
2017年

１．発表者名
李 珍泌、上野 知洋、佐藤 三久、佐野 健太郎

２．発表標題
HPC向けストリームプロセッサ生成のためのC言語フロントエンドの開発

３．学会等名
第163回ハイパフォーマンスコンピューティング研究発表会

４．発表年
2018年

〔図書〕　計0件

〔産業財産権〕

〔その他〕

―

６．研究組織

	氏名 （ローマ字氏名） （研究者番号）	所属研究機関・部局・職 （機関番号）	備考