

独自ベクトル処理機能を備えたプロセッサ向け 自動ベクトル化コンパイラの開発

永池 晃太郎[†] 大津 金光^{††} 横田 隆史^{††} 小島 駿^{††}
[†] 宇都宮大学工学部情報工学科 ^{††} 宇都宮大学大学院地域創生科学研究科

1 はじめに

我々は機械語コードの変更なしに同時並列演算数を変更することができる, データ並列処理のためのスケーラブルなベクトル拡張を実現したベクトル拡張付き RISC-V を提案している [1].

我々のベクトル拡張付き RISC-V は, ARM のベクトル拡張である ARM SVE[2] を参考に組込み向けに RISC-V[3] をベクトル拡張したものである. これにより, 機械語コードが同時演算数に依存しないスケーラブルなベクトル拡張を実現したが, ベクトル拡張に対応したコンパイラがない.

この問題に対して, 解決策として既存の RISC-V コンパイラに変更を加えることによって, 我々のシステム向けのベクトル化アセンブリコードを得ることのできるコンパイラを開発するという手段を考えた.

コンパイラの開発はコンパイラ基盤である LLVM[4] を用いて行う. コンパイラ基盤はコンパイラの機能がモジュール化されており, 既存機能の再利用が可能になっている. そのため, コンパイラの開発の際にコンパイラ基盤を用いることによって独自部分のみの開発で済む. 前述した通りベクトル拡張付き RISC-V は RISC-V を拡張したものであることから, LLVM の RISC-V コンパイラとしての機能を再利用してコンパイラを開発を行う.

2 ベクトル拡張付き RISC-V

ベクトル拡張付き RISC-V は RISC-V が有するカスタム命令のためのオペコード領域を利用している. カスタム用オペコード領域は 4 つが用意されているが, そのうちの 2 つを利用しており, 1 つをベクトルロード, ストア命令, もう一方をベクトル演算命令とベクトル制御命令に使用している.

3 LLVM コンパイラ基盤

LLVM の構成を図 1 に示す. LLVM では C 言語などのソースコードから中間表現である LLVM IR に変換を行うフロントエンド, LLVM IR に対して最適化, アセンブリコード生成を行うバックエンドからなる.

LLVM には自動ベクトル化機能が備わっている. フロントエンドにて入力ソースコードの繰り返しによる処理をベクトル化された LLVM IR に変換を行う.

LLVM のバックエンドでは RISC-V とその拡張命令セットが対応しているが, ベクトル拡張である V 拡張については現時点では実験的な実装にとどまっている.

4 LLVM バックエンドにおける独自命令の生成機能の実装

謝辞

本研究は, 一部日本学術振興会科学研究費補助金 (基盤研究 (C)24500055, 同 (C)15K00068) の援助による.

参考文献

- [1] 氏名, 氏名, 氏名: “タイトル”, 情報処理学会論文誌: コンピューティングシステム, Vol.X, No.SIG X, pp.X-Y, 20XX.

Development of automatic vectorization compiler for processors with original vector processing function.

[†]Nagaike Kotaro, ^{††}Ootsu Kanemitsu, ^{††}Yokota Takashi, ^{††}Kojima Shun,

Department of Information Science, Faculty of Engineering, Utsunomiya University ([†])