

## 発表概要

# LLVM IR レベルディレクティブに基づいた For ループの自動並列化の実現

神宮 健吾<sup>1,a)</sup> 大津 金光<sup>1,b)</sup> 横田 隆史<sup>1,c)</sup>

2020年1月15日発表

マルチコアプロセッサの普及によりマルチスレッディングによる処理性能の向上が可能となった。しかしながら、プログラムの中にはそのソースコードを参照できないものもあり、それらはソースコードレベルでの並列化が不可能である。この問題に対して、バイナリ変換技術を用いたバイナリコードの自動並列化が有望な手法として提案されている。この背景のもと、我々はバイナリコードを LLVM コンパイラ基盤の中間表現 (LLVM IR) に変換し、LLVM のコンパイラパスを活用して並列化するシステムを提案してきた。しかしながら、LLVM が備える並列化では、コード解析とコード生成パスは密接につながっており、それらのパスを再利用して新たな並列化手法を実現するには膨大な時間と労力が必要であった。本研究では、並列化におけるコード解析とコード生成を独立に開発・実行するために、LLVM IR における OpenMP のようなディレクティブを提案する。本研究では初期実装として for ループのためのディレクティブと、それに基づいて LLVM IR を並列化するコンパイラパスを開発した。ベンチマークスイートでの実験の結果、我々のディレクティブにより並列化したプログラムは、逐次プログラムと同様の結果を出し、OpenMP API を用いてソースコードレベルで並列化したものと同程度の高速化を達成した。

## Presentation Abstract

### An Implementation of Automatic Parallelization for For-Loops Based on A Directive at LLVM IR-Level

KENGO JINGU<sup>1,a)</sup> KANEMITSU OOTSU<sup>1,b)</sup> TAKASHI YOKOTA<sup>1,c)</sup>

Presented: January 15, 2020

With the spread of multicore processors, it has been possible to improve processing performance by multithreading. However, some programs can not be able to get its source code, and can not be parallelized at source code level. To solve this problem, automatic parallelization of binary code using binary translation technique has been proposed. Based on this background, we have proposed a system that translates a binary code into the intermediate representation of the LLVM compiler infrastructure (LLVM IR) and parallelizes a program utilizing LLVM compiler passes. However, in the parallelization of LLVM, code analysis and code generation passes are closely connected, and it takes a lot of time and effort to reuse them to implement new parallelization methods. In this research, we propose directives like OpenMP in LLVM IR to independently develop and execute code analysis and code generation passes for parallelization. As an initial implementation, we develop a directive for for-loop and compiler passes for parallelizing LLVM IR based on it. As a result of experiments, programs parallelized by our directive output the same result as the sequential one, and they achieved speedup almost same as the programs parallelized at source code level using OpenMP API.

This is the abstract of an unrefereed presentation, and it should not preclude subsequent publication.

<sup>1</sup> 宇都宮大学工学研究科情報システム科学専攻  
Department of Information Systems Science, Graduate  
School of Engineering, Utsunomiya University, Utsunomiya,  
Tochigi 321-8585, Japan

a) jingukng@virgo.is.utsunomiya-u.ac.jp  
b) kim@virgo.is.utsunomiya-u.ac.jp  
c) yokota@virgo.is.utsunomiya-u.ac.jp