発表概要

OpenCLコードの生成による Elixirアプリケーションの高速化

安部 竜矢[1,a)]　山崎 進[2]　高瀬 英希[1,3]

2020年7月31日発表

近年，プロセッサ 1 つのコアのクロック周波数が伸び悩んでおり，コア数を増やすことでプロセッサの性能の向上を図っている．マルチコアプロセッサを活用するためには，ソフトウェアで並列処理を明示したプログラムを記述する必要があるが，既存のフレームワークにおいて並列処理を記述するのは容易ではない．関数型言語の Elixir はパイプライン演算子と MapReduce モデルを組み合わせたプログラミングスタイルや，イミュータブル性などの特徴から並列処理の記述に非常に適している．現在，Elixir の記述をもとに SIMD 命令の生成を行う言語処理系 Pelemay の開発が進められている．Pelemay は並列化を記述した Elixir コードからマルチコアの性能を活用できる SIMD コードを生成することで，この高速化に成功している．しかし，Pelemay はプロセッサ以外の計算資源を使えない，パイプ演算子単位でコード生成を行うために通信のオーバヘッドが大きいなどの問題点がある．本論文の目的は Elixir アプリケーションの高速化である．そのために，Elixir の記述から OpenCL のコードを生成する手法を提案する．OpenCL コードは GPU 上で並列実行可能であるため，Elixir から GPU を駆動させることで高速化の達成を図る．

Presentation Abstract

Accelerate Elixir Application by Generating OpenCL Code

Tatsuya Abe[1,a)]　Susumu Yamazaki[2]　Hideki Takase[1,3]

Presented: July 31, 2020

In recent years, the clock frequency of one core of the processor has been stagnant, and the performance of the processor has been improved by increasing the number of cores. To use a multi-core processor, it is necessary to write a program that specifies parallel processing in software, but it is not easy to describe parallel processing in existing framework. The functional program language Elixir is very suitable for describing parallel processing because of its programming style combining pipeline operators and MapReduce models and immutability. At present, Pelemay, a language processing system that generates SIMD code based on Elixir, is under development. Pelemay has succeeded in speeding up Elixir code by generating SIMD code. But Pelemay has problems such as not being able to use computational resources other than the processor, and having a large communication overhead because code is generated for each pipe operator. The purpose of this research is to accelerate the application of Elixir. We propose a method to generate OpenCL code from Elixir description. Since OpenCL code can be executed in parallel on GPU, we aim to achieve high speed by driving GPU from Elixir.

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

[1] 京都大学大学院情報学研究科
Graduate School of Informatics, Kyoto University, Kyoto 606–8501, Japan
[2] 北九州市立大学国際環境工学部
Faculty of Environmental Engineering, The University of Kitakyushu, Fukuoka 808–0135, Japan
[3] JST さきがけ
PRESTO Program, Japan Science of Technology Agency, Kawaguchi, Saitama 332–0012, Japan
[a)] abe@lab3.kuis.kyoto-u.ac.jp