発表概要

Hastega：Elixirプログラミングにおける超並列化を実現するためのGPGPU活用手法

山崎 進[1,a)]　森 正和[2,b)]　上野 嘉大[2,c)]　高瀬 英希[3,d)]

2018年8月1日発表

Elixir では Flow という MapReduce の並列ライブラリが普及している．Flow を用いると簡潔な表現でマルチコア CPU の並列性を活用できる．我々は Flow によるプログラム記述が GPGPU にも容易に適用できるという着想を得て，OpenCL によるプロトタイプを実装した．現行の GPU で採用される SIMD では，単純な構造で均質で大量にあるデータを同じような命令列で処理する場合に効果を発揮する．一方，Flow では，そのようなリストに対し一連の命令列で処理する．そこで，このような命令列と，リストを配列化したデータを GPU に転送・実行することで高速化を図る手法，Hastega を提案する．そこで，ロジスティック写像を用いたベンチマークプログラムを開発し，期待される性能向上を評価した．Mac Pro と GCE で評価した．言語は Elixir，Hastega，Rust，Python で比較した．その結果，次の 3 つの結果が得られた：(1) Hastega は Elixir 単体のコードと比べて 4.43〜8.23 倍高速になった (2) Hastega は Rust と比べて，1.48〜1.54 倍程度遅くなっただけである (3) Hastega は Python のコードと比べて，3.67 倍高速である．今後，本研究で得た知見を元に LLVM を用いてコード生成器を含む処理系を開発する予定である．

Presentation Abstract

Hastega: A Method Using GPGPU for Super-Parallelization in Elixir Programming

Susumu Yamazaki[1,a)]　Masakazu Mori[2,b)]　Yoshihiro Ueno[2,c)]　Hideki Takase[3,d)]

Presented: August 1, 2018

Elixir has Flow that is a popular parallel library similar to MapReduce. Flow can realize parallel programming on multi-core CPUs by simple description. We ideate that code description of Flow can be applied to GPGPU easily, and implements prototypes. The SIMD architecture that current GPUs adopt is effective when a single instruction sequence processes a simple, homogeneous and mass data structure, and code using Flow is such a single instruction sequence that processes such a linked-list structure. Thus, we propose the Hastega method, which is a code optimization method by sending such an instruction sequence and a mass array composed of data from the linked-list, and executing them. We develop a benchmark suit of the logistic maps, and evaluate performance of the proposal system using it. We evaluate it using Mac Pro and GCE. We compare Elixir, Hastega, Rust and Python. We get the following results: (1) Hastega is 4.43–8.23 times and 7.43–9.64 times faster than pure Elixir and Python code executed by only CPU, respectively. (2) The performance gap of Hastega is only 1.48–1.54 times compared with Rust. (3) Additionally, Hastega achieves 3.67 times faster than Python. We plan to develop a processing system including a code generator using LLVM based this research.

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

[1] 北九州市立大学
University of Kitakyushu, Kitakyushu, Fukuoka 808–0135, Japan
[2] 有限会社デライトシステムズ
Delight Systems Co., Ltd., Fukuoka 810–0067, Japan
[3] 京都大学
Kyoto University, Kyoto 606–8501, Japan
[a)] zacky@kitakyu-u.ac.jp
[b)] mori@delightsystems.com
[c)] delightadmin@delightsystems.com
[d)] takase@i.kyoto-u.ac.jp