

発表概要

バリア同期と共有メモリを備えた GPGPUプログラム合成器 Kani-CUDA

蟹 暁^{1,a)} 朝倉 泉² 増原 英彦² 青谷 知幸²

2017年3月3日発表

GPGPU とは GPU を汎用計算のために用いる技術であり、高い並列性によって大規模な計算を高速に行うことができる。しかし、GPGPU プログラミングは一般的なプログラミングよりも記述が複雑であり正しさを確信することが難しい。たとえば、共有メモリを用いたブロッキング最適化におけるメモリアドレスの計算式が複雑になることや、非同期的に実行されるスレッド間でメモリ競合が起きないことの確認などである。このような難しさに対して GPGPU プログラミングを支援するためのプログラム合成器が考案されている。しかし、これらの合成器はバリア同期や共有メモリがない単純な実行モデルを採用しているため、高度な最適化がなされたプログラムを扱うことができなかった。そこで我々はバリア同期と共有メモリを備えたより CUDA に近い意味論を持ったプログラム合成器 Kani-CUDA を提案する。Kani-CUDA は既存の GPGPU プログラム合成器 SynthCL をもとに設計されているが、その実行方式は SynthCL がインターリーブ実行であるのに対し、Kani-CUDA ではバリア同期を扱うためにロックステップ実行を採用している。応用例として熱拡散ステンシル計算に共有メモリ最適化を施したプログラムにおいて、配列の添え字計算式やバリア同期命令の配置の合成を達成した。

Kani-CUDA: A GPGPU Program Synthesizer with Barrier Synchronization and a Shared Memory

AKIRA KANI^{1,a)} IZUMI ASAKURA² HIDEHIKO MASUHARA² TOMOYUKI AOTANI²

Presented: March 3, 2017

GPGPU uses a GPU for general purpose computation and performs large amount of computation at high speed by exploiting high parallelism. However, compared to sequential programming, GPGPU programming is more complicated, and more difficult to confirm correctness. For example, blocking optimizations using shared memory require complicated memory address calculations, and asynchronously executed threads require confirmation of memory race freedom. To cope with those difficulties, there are program synthesizers that support GPGPU programming. However, those synthesizers cannot deal with highly optimized programs, since they have a simple execution model that does not have barrier synchronization and a shared memory. We propose a program synthesizer Kani-CUDA that has semantics close to CUDA, in particular supporting barrier synchronization and a shared memory. The design of Kani-CUDA is based on the one of SynthCL, an existing GPGPU program synthesizer. Unlike SynthCL that uses an interleave execution model, Kani-CUDA uses a lock-step execution model to deal with barrier synchronization. As an application, we synthesized array index expressions and placement of barrier synchronization statements in a stencil computation for a thermal diffusion with a shared memory optimization.

¹ 東京工業大学理学部情報科学科
Department of Information Science, Tokyo Institute of Technology, Meguro, Tokyo 152-8550, Japan

² 東京工業大学情報理工学院数理・計算科学系
Department of Mathematical and Computing Science, Tokyo Institute of Technology, Meguro, Tokyo 152-8550, Japan

a) kani.a.aa@m.titech.ac.jp