

Java meets CUDA

Studenten



Marc Emch



Floris Staub

Einleitung: Prozessoren sind darauf ausgerichtet, viele komplexe Operationen sequenziell auszuführen. Auch moderne Multicore-Prozessoren ermöglichen nur eine geringe Parallelisierung. Dem gegenüber steht die Grafikkarte. Diese kann lediglich simple arithmetische Operationen ausführen, hat jedoch tausende von Threads, die parallel ausgeführt werden können. Dadurch ist es möglich, sehr rechenintensive Programme, wie zum Beispiel Finanzmarkt-Simulationen, stark zu beschleunigen.

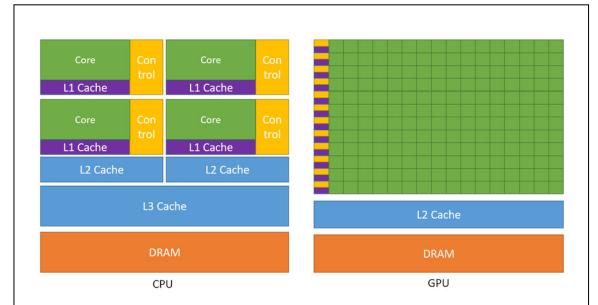
Ziel der Arbeit: In vielen Programmiersprachen ist es möglich, mit geringem Aufwand des Entwicklers Teile des Rechenaufwandes auf die Grafikkarte auszulagern. Java bietet jedoch von sich aus keine solche Möglichkeit. Da Java immer noch eine der meistverbreiteten Programmiersprachen ist, hat diese Arbeit das Ziel, eine Lösung zu entwickeln, die es einem Programmierer ermöglicht, Teile eines Java-Programms auf der Grafikkarte auszuführen. Dabei soll möglichst wenig zusätzlicher Aufwand für den Programmierer entstehen.

Ergebnis: In dieser Arbeit wurde ein Tool entwickelt, welches markierte Teile eines Java-Programms nimmt und in CUDA-Code übersetzt, sodass diese auf Nvidia-Grafikkarten ausgeführt werden können. Dabei wird Java Bytecode als Input genommen, die markierten Teile werden ausgeschnitten und durch einen "native" Funktionsaufruf ersetzt. Diese native Funktion wird dann als C Source Code erzeugt, und zusammen mit dem CUDA-Code in eine Shared Library kompiliert, welche letztendlich von der Java Virtual Machine geladen und an die Klasse angebunden wird, aus welcher der Code ursprünglich ausgeschnitten wurde.

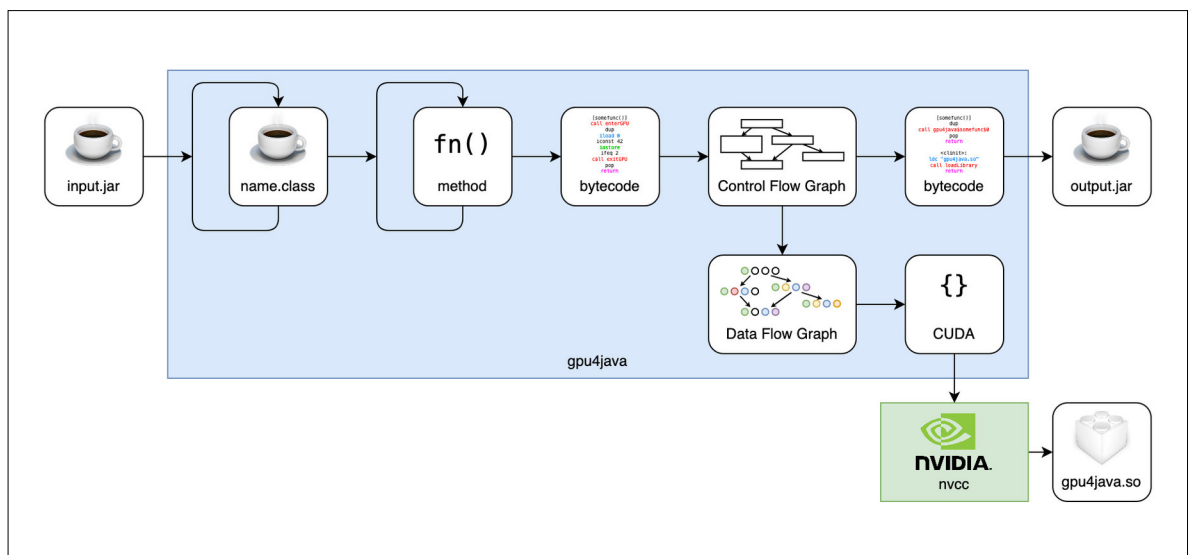
Eine Nvidia GTX 1050 Ti Grafikkarte.
nvidia.com



Architekturvergleich zwischen CPU und GPU.
nvidia.com



Pipeline unseres Tools.
Eigene Darstellung



Examinatorin
Prof. Dr. Mitra
Purandare

Themengebiet
Software