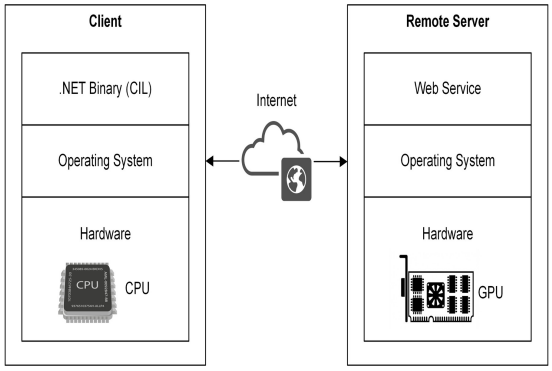
André Gasser

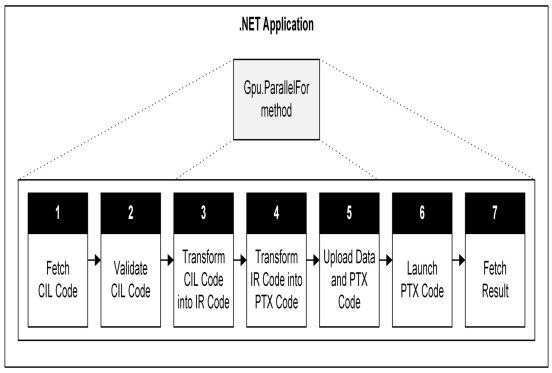
Students	André Gasser
Lecturers	Prof. Dr. Luc Bläser
Advisors	Dr. Felix Friedrich , ETH , Zürich , ZH
Topic	Software and Systems

GPU Parallelization as a Service

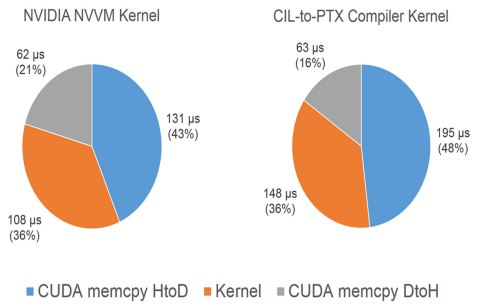
Building a system which allows execution of C# code on a remote GPU



System Overview



Code Transformation and Execution Overview



Kernel Performance Overview

General purpose GPU (GPGPU) computing gains momentum these days. However, many issues remain unsolved. During this master thesis, two major issues software engineers have to cope with have been addressed. First, development machines do often not possess enough GPU cores to handle large computational workloads. Operating a fully-equipped GPU computing cluster is often not a realistic scenario, as it means investing a lot of money in hardware. Likewise, the recurring costs must not be forgotten, such as operating personnel and expenses generated by power consumption of GPU clusters. Second, general purpose GPU computing is still a very low-level, hardware-oriented task. It requires specific knowledge within the field. This can be counter-productive, especially in today's software engineering world, in which projects are subject to demanding release schedules. Modern managed development environments, like .NET, do not provide built-in support for cross-compiling their code into code targeting the GPU platform. Although specific third-party libraries, such as CUDAfy.NET, exist, they often operate on a low abstraction level and require specific knowledge about how GPUs function internally.

It becomes clear that the issues mentioned above lead to an increase in software development expenses and effort. To address them, two prototypes, a compiler and a cloud web service, have been developed. The compiler translates CIL code into PTX assembly during runtime. An important design goal was to create a lightweight runtime system which can handle the required compilation tasks without requiring excessive libraries such as LLVM. Everything was constructed using pure CUDA technology from NVIDIA Corporation. The cloud web service acts as a GPU as a Service component. It hosts at least one GPU and executes PTX assembly code sent to it. This way, computational workloads can be offloaded to a remote system, which is equipped with proper GPU hardware. Both components together, the compiler and the cloud web service, provide a seamless development experience for the software engineer. They allow a software engineer to write program code in his favorite .NET programming language. This approach does not require the software engineer to possess any knowledge about the inner workings of GPUs. From a project manager's point of view, significant savings in time and money can be achieved.

This thesis has shown that offloading GPU workloads to remote systems is a feasible task. End users and service providers are decoupled by a well-defined service interface, which allows both of them to change or scale without affecting the opposite party. Also, a working compiler was built, which is able to translate a minimal subset of the CIL instruction set into PTX assembly. This is a key element in building a seamless user experience for the .NET software engineer. Regarding the future, the GPU as a Service concept could soon become a major business use case, in which developers, hardware manufactures and companies building development tools have an equally high stake.