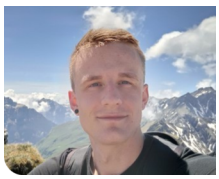
CuSharp

A GPU Compute Framework for .NET

Graduate



Adrian Locher

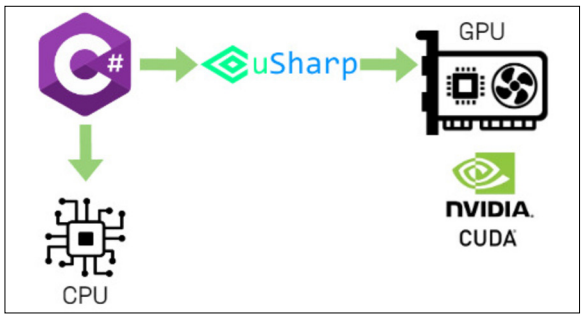
Jason Benz

Introduction: The number of computationally intensive applications is growing. For many easily parallelizable problems, GPUs offer better performance than CPUs. As a result, GPUs are now used not only for graphical applications, but also for machine learning and cryptography. GPU-accelerated programs have traditionally been written in C, C++ for high-performance applications such as physics simulations and graphical applications and more recently in Python to optimize machine learning algorithms. Most GPU-APIs, including Nvidia CUDA, restrict their developers to using C, C++ or Python to write programs targeting those APIs.

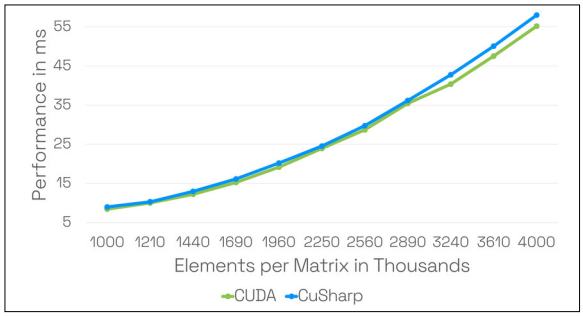
Result: In this thesis a framework called CuSharp has been developed that allows developers to build and run GPU-executable kernels directly in C#. This is achieved by using existing toolchains complemented by a specifically developed cross-compiler. The C# kernel is compiled to Microsoft Intermediate Language (MSIL) by the existing Roslyn compiler. Subsequently, the CuSharp compiler cross-compiles MSIL to NVVM IR, a platform-independent intermediate representation. Finally, NVVM IR is translated to PTX ISA, an assembly-like language for Nvidia GPUs, using the NVVM compiler library (libNVVM). The architecture also allows for future development efforts to add cross-platform support. CuSharp supports the compilation of static methods, written in a specific C# subset, either just-in-time or ahead-of-time. For a kernel that computes matrix multiplications, a performance slowdown between 1.4% and 4.8% was measured for CuSharp-compiled kernels (GPU execution time of the kernel only).

Conclusion: This thesis shows the challenges of interfacing with the LLVM compiler infrastructure and the Nvidia CUDA API. In addition, it provides an overview of the complex landscape of APIs that can be used to interface with GPU devices in general, by comparing their toolchains and languages. Furthermore, it demonstrates that GPU kernels can be written in a high-level language such as C# while suffering only minor performance degradation.

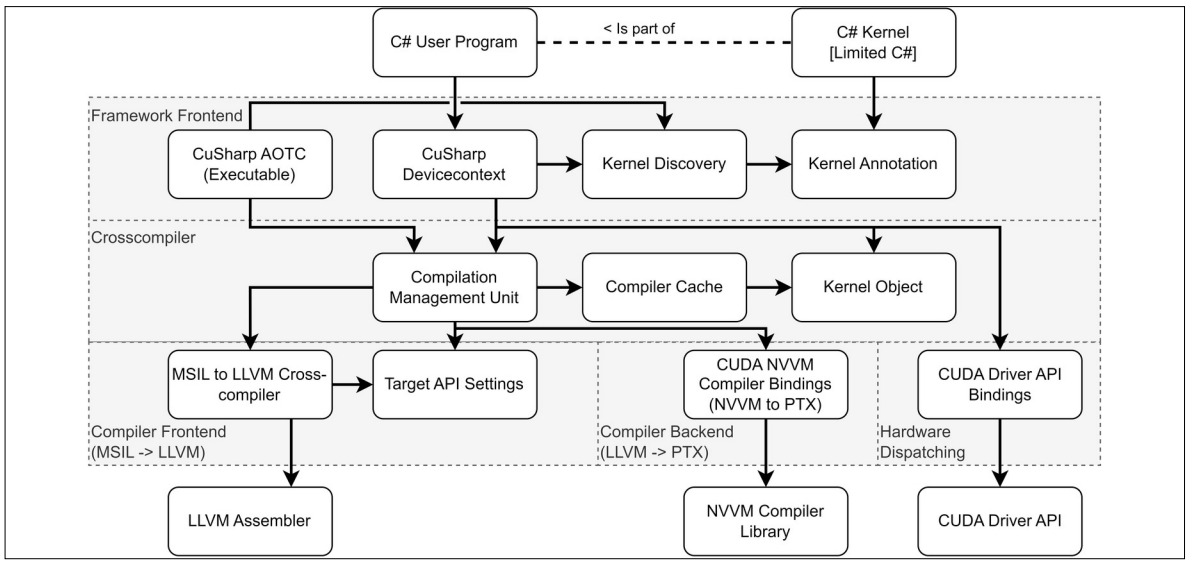
Overview of CuSharp Interacting with External Systems.
Own presentment



Performance Results of a Double-Type Matrix Multiplication.
Own presentment



Software Architecture of the CuSharp Framework.
Own presentment



Advisor
Philipp Kramer

Co-Examiner
Christian Marrocco, Arni AG, AG

Subject Area
System Software, Software, Software Engineering - Core Systems