High-Performance IT Systems

Automatic Algorithm Recognition and Replacement for FPGA Overlays

Bachelor/Master Thesis

In a Nutshell
- Get familiar with domain-specific FPGA overlays
- Identify suitable methods for automatic algorithm recognition and replacement
- Implement the method for a certain application domain (e.g. image processing, machine learning)
- Demonstrate usage with different algorithm variations that are recognized and replaced

Many applications can benefit from hardware acceleration. Essentially, the whole application or the compute intensive parts are moved to the accelerator to increase performance or reduce the energy footprint. However, migrating an existing application or developing the hardware variant from scratch is challenging. One approach to tackle this problem is FPGA overlays. FPGAs can accelerate a wide range of applications by customizing the hardware towards the required compute and data access patterns. However, FPGAs themselves are difficult to program and require time-consuming synthesis processes. FPGA overlay architectures provide an abstraction layer on top of the original FPGA fabric. The granularity of the overlays can vary from fine-grained operations (add, shift, etc.) to coarse-grained structures (adders, vector processors, soft cores, etc.). Instead of synthesizing the FPGA fabric (time-consuming), the overlay can be reconfigured to the required needs (fast, similar to software compilation). But how can we automatically use FPGA overlays in a transparent way to the user?

The goal of this thesis is to automatically detect the compute intensive parts of an application (for a certain domain, e.g. image processing and/or machine learning) and replace it with a call to the FPGA overlay. The main focus is not on the design of the overlay architectures (fundamental work in this regard has been done and can be reutilized), but on the recognition of equivalent algorithm variants that can be matched to the same overlay. The figure shows a schematic example, where three different matrix multiplication variants (one can think of variants with/without pointers, different data types, etc.) can be matched to the same overlay, which is configured to serve the different purposes.

Recommended Skills
- Interest in algorithms, data structures and compiler technologies (LLVM)
- Interest in hardware acceleration and computer architecture (mainly FPGAs)
- Experience in software development in C/C++

Supervisor
Heinrich Riebler, Phone: 05251/60-5382 E-Mail: heinrich.riebler@upb.de

For more information please visit:
http://cs.uni-paderborn.de/hit/teaching/student-projects/open-theses-topics/