Design-Space Exploration for FPGA-Accelerators in Cloud Environments

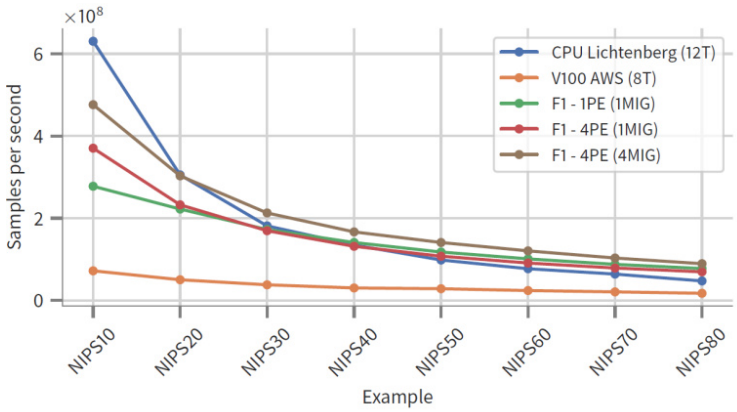

Figure 1: Performance of the SPN cores on different platforms.

Project Manager
Lukas Sommer

Principal Investigator
Prof. Dr.-Ing. Andreas Koch

Project Term
2019 - 2019

Clusters
Lichtenberg Cluster Darmstadt

Additional Software
Gurobi

Institute
Institute of Scientific Computing

University
Technische Universität Darmstadt

Introduction

FPGA cards are increasingly deployed to cloud data centers and made available to users as platform for the implementation of specialized accelerators, for example by Amazon in their AWS F1 instances. As part of our research, we develop an open-source framework (TaPaSCo) that allows to easily build heterogeneous systems, which offload compute-intensive tasks to FPGA-accelerators. Recently we have extended our framework to support cloud data-center setups. Based on this framework, we now seek to develop a heterogeneous system for high-throughput inference in so-called Sum-Product Networks (SPNs). SPNs are a special class of machine learning networks. Part of the development process is the design-space exploration (DSE), to determine the optimal number of parallel processing elements, the optimal operating frequency, etc. The use of the Lichtenberg cluster will allows us to efficiently and automatically traverse the large design space.

Methods

In order to configure an FPGA, typically a so-called bitstream is needed. In case of the Amazon F1 platform, an Amazon FPGA Image (AFI) must be generated. In both cases, the process starts with a block design. This block design is generated using the TaPaSCo tool flow, which automatically generates system-on-chip (SoC) designs. Then, the design is synthesized and implemented, which are very computing intensive tasks. For large devices, like the Xilinx UltraScale+ VU9P FPGA on the F1 platform, this results in run times of many hours even on high-end hardware. Whether the design actually works is only known after the implementation is finished. In case of a failure, the whole process is repeated with modified parameters. Usually,

the clock frequency is reduced. TaPaSCo can automate this task using the DSE. TaPaSCo creates a set of configurations and implements all of them. As soon as a working implementation is found, the task is finished. Should all configurations result in an implementation failure, TaPaSCo generates a new set of configurations. Our research benefits from compute nodes with a moderate core count and with a high amount of memory. As some steps of synthesis and implementation cannot run multi-threaded, we restrict the core count for each implementation run (task) to one. At the same time, a single task can consume up to 32 GB of memory[1]. Thus, the MEM section of the HHLR is a perfect fit for running a DSE. Furthermore, we use the MPI nodes for benchmarks of the SPNs on CPUs.

Results

Using the HHLR to run the DSE, we are able to find the optimal design clock frequency for a variable number of SPN cores and and memory channels. We obtain the optimal clock frequency for the following design variations:

- 1 SPN core, 1 memory channel: Least resource usage and thus highest operating frequency.
- Up to 4 SPN cores, 1 memory channel: Increased resource usage reduces operating frequency. Still, multiple cores computing in parallel might outweigh the reduced frequency. However, the single memory channel imposes a bottleneck.
- Up to 4 SPN cores, 4 memory channels: Additional memory channels reduce the operating frequency further. Nevertheless, the bottleneck of only having a single, shared memory channel is removed.

Discussion

Using the results from the DSE, we obtain the maximum design frequency for each of the configurations listed above. Using this information, we benchmark all configurations and identify the best-performing configuration. As the results (Fig. 1) show, using multiple (up to 4) SPN cores (PEs) together with four memory channels shows the best performance. The second best configuration is pairing one PE with one memory channel. This configuration shows good performance because it achieves high design frequencies, due to comparably low resource usage. Using multiple PEs with a single memory channel is not a good choice, because concurrent access on a single link leads to congestion of the bus. That is, the PEs cannot get input data or write out results fast enough, effectively stalling the computation. In addition, we compare the FPGA-accelerated results on the F1 platform with CPU and GPU implementations. A 12-core Xeon CPU is outperformed by a factor of up to 1.9x and a Nvidia Tesla V100 graphics processing unit (GPU) is outperformed by a factor of up to 6.6x. We aim to continuously improve TaPaSCo and the generated designs. Optimizations may enable higher design frequencies or more efficient resource usage. However, this would require running a new DSE. Furthermore, we may look into benchmarking different use-cases on the Amazon F1 platform. In all cases, using the HHLR

JUSTUS-LIEBIG-UNIVERSITÄT GIESSEN TECHNISCHE UNIVERSITÄT DARMSTADT UNIKASSEL VERSITÄT Philipps Universität Marburg

enables us to find the optimal design parameters.

Publications

Ober, M.; Hofmann, J.; Sommer, L.; Weber, L.; Koch, A.: High-Throughput Multi- Threaded Sum-Product Network Inference in the Reconfigurable Cloud. In Fifth International Workshop on Heterogeneous High-performance Reconfigurable Computing (H2RC), 2019 https://doi.org/10.1109/H2RC49586.2019.00009

Reference

[1]See table for device “XCVU9P” Xilinx Memory Recommendations, Accessed 2020-05-31. https://www.xilinx.com/products/design-tools/vivado/memory.html

Last Update: 2023-03-16 01:29