Optimizing Hardware for Neural Network Inference using Virtual Prototypes

Jan Zielasko[1,2], Rolf Drechsler[1,2]

[1] Institute of Computer Science, University of Bremen, Germany

[2] Cyber-Physical Systems, DFKI GmbH, Germany

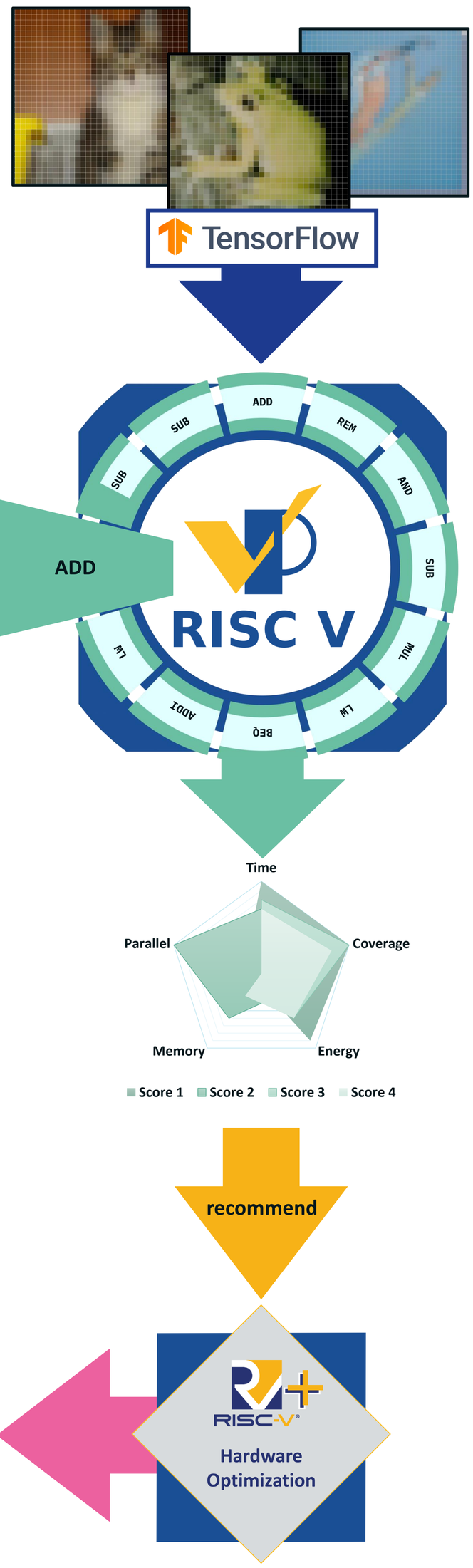
Jan.Zielasko@DFKI.de

Overview

- **Tailoring hardware** to applications significantly increases their performance.
- **Virtual Prototypes** (VPs) enable early software development and design space exploration
- **RISC-V Opt-VP** is a Virtual Prototype driven binary analysis platform
- By analyzing the execution, it identifies **instruction sequences** that are promising candidates for hardware optimization

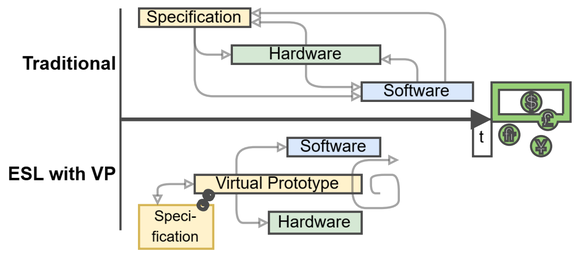
1. Application

- Running **MLPerf Inference: Tiny Deep Learning Benchmarks for Embedded Devices**
- E.g., a ResNet8 image classification model trained on the CIFAR10 dataset
- Using **TensorFlow Lite for Microcontrollers**



2a. Virtual Prototype Driven Tracing

- Tracing module interfacing ISS core
- Taint tracking at instruction level
- Construct **bounded execution trees**
- Lossless compression of trace information



2b. Execution Trees

- Compress trace data into execution trees

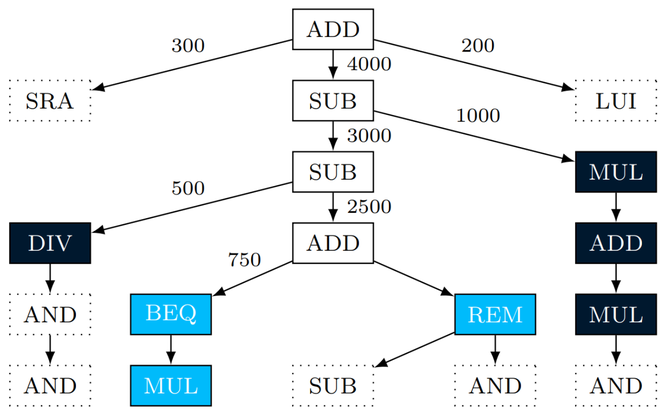


Figure 1: Excerpt of bounded execution tree for the ADD instruction

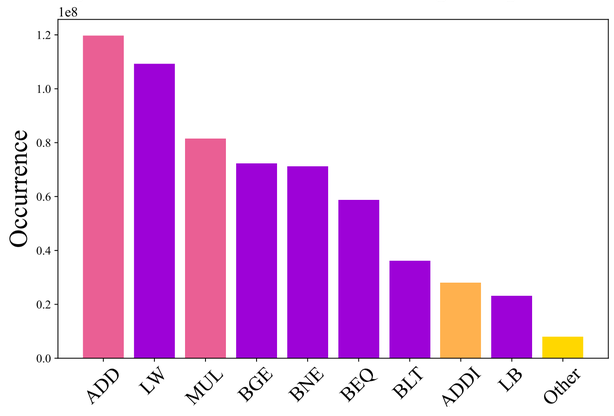
☐ Discovered Sequence | ■ Variant | ■ Considered for extension

3. Analysis

- Analyze trees using **scoring function**
- Choose a set of **metrics** that match the target hardware optimization
- Evaluate all discovered instruction sequences to identify best suited sequence

4. VP Evaluation

Figure 2: Instruction distribution for image recognition



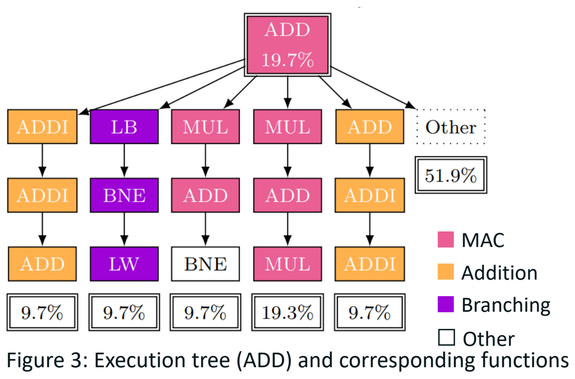
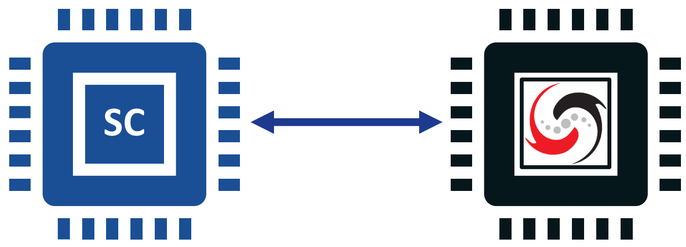
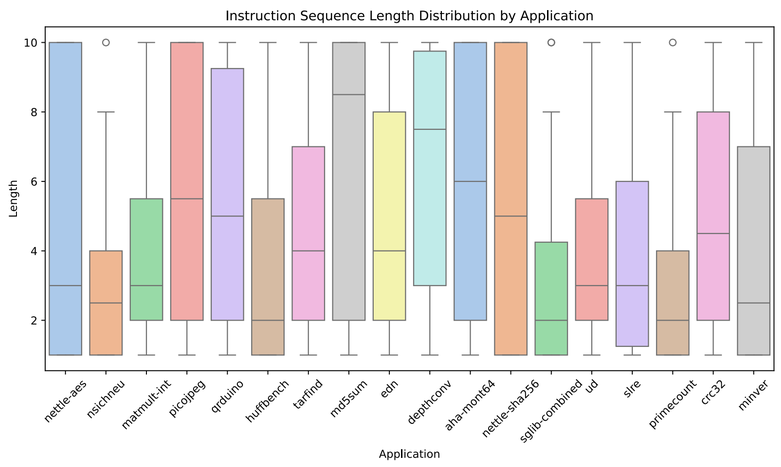


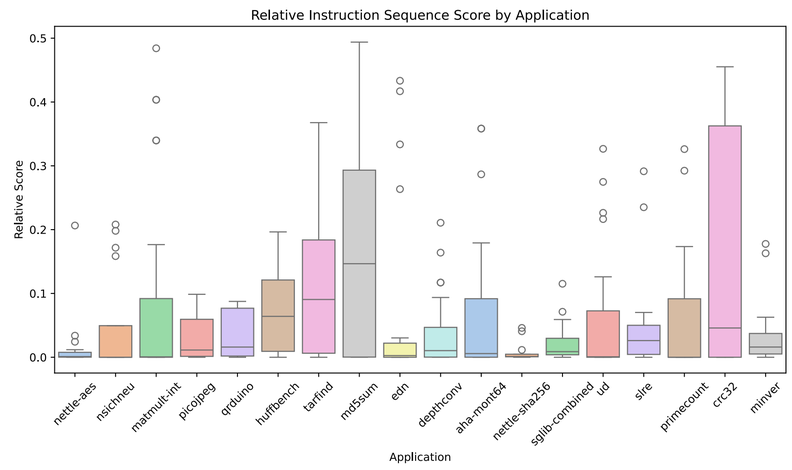
Figure 3: Execution tree (ADD) and corresponding functions

5. Work in Progress

- Implement behavior on VP/TLM level
- Estimate performance impact
- Generate RTL design using SpinalHDL
- Use Co-Simulation to compare results







Analysis of Embench 2.0

- Full tracing and analysis of new Embench set up to depth 10
- Coverage as optimization target
- Score = % of execution accelerated
- Showcases the effectiveness of custom instructions for different applications

Funded by:

Federal Ministry of Education and Research

ECXL

FAIRe

Grant number 01IW22002 and 01IS23074

More info on GitHub

Universität Bremen

dfki ai Deutsches Forschungszentrum für Künstliche Intelligenz *German Research Center for Artificial Intelligence*