PRUNE
Dynamic and Decidable Dataflow for Signal Processing on Heterogeneous Platforms

Version 1.0.0

Jani Boutellier¹
Jiahao Wu²
Heikki Huttunen¹
Shuvra S. Bhattacharyya²

¹ Tampere University of Technology
² University of Maryland
Dataflow computing (1/2)

• In dataflow computing a program is expressed as a graph consisting of actors (nodes)
• Actors are independent procedures that exchange information over first-in-first-out (FIFO) channels

Fig. 1. A dataflow graph with 4 actors.
Dataflow computing (2/2)

- The actors of a program can easily be distributed to different execution units of a multicore platform.
- FIFO-style communication channels enable *pipelined processing*.

![Diagram of pipelined processing](image)

Fig.2. Pipelined processing of 3 program iterations (blue, red, yellow) on three parallel cores.
PRUNE - Introduction

• PRUNE is Linux-based Open Source dataflow computing framework
• PRUNE inherently supports OpenCL devices such as GPUs and multicore CPUs
• The programmer describes the functionality of each actor in C or OpenCL, and the actor graph in XML
• PRUNE provides a C-based runtime library with a small API that enables actors to exchange information according to the dataflow computing rules
• PRUNE has been designed maximum processing performance in mind
PRUNE – Framework (1/3)

• PRUNE consists of three main components
  – PRUNE compiler
  – PRUNE runtime
  – PRUNE analyzer

• The PRUNE compiler is a source-to-source compiler that translates the XML actor graph into a top-level C file that
  – Initializes the actors
  – Initializes OpenCL
  – Starts program execution
  – Cleans up the program after finishing
The PRUNE runtime provides C API calls

- fifoReadStart() / fifoReadEnd()
- fifoWriteStart() / fifoWriteEnd()

for actors that are *not* to be executed on OpenCL devices

Non-OpenCL actors are required to have the following functions: `actorInit()`, `actorFire()`, `actorFinish()`

Actors that are to be mapped to OpenCL devices are written in regular OpenCL. For these actors the framework provides *port* names that need to be used to ensure correct operation (port is a connection to a FIFO)
PRUNE – Framework (3/3)

• The PRUNE analyzer is a prototype software for checking the correctness of PRUNE graphs (i.e. not the C / OpenCL code within actors)

• The analyzer is based on the University of Maryland DIF framework, but the required support binaries are provided with PRUNE

• The analyzer checks the PRUNE graphs against the 5 design rules that are described in the main PRUNE publication "PRUNE: dynamic and decidable dataflow for signal processing on heterogeneous platforms" (IEEE Transactions on Signal Processing)
PRUNE - Applications

- A couple of application examples are provided with PRUNE
  - A run-time reconfigurable predistortion filter
  - A video motion detection application
  - A convolutional neural network for vehicle classification
- Details of these applications can be found in the IEEE Transactions on Signal Processing article
- The applications have been tested e.g. on the following OpenCL hardware
  - Intel i7, AMD Carrizo, AMD RX, AMD Capeverde, ARM Mali
PRUNE - Comparison

• PRUNE is inspired by, and in many ways similar to, the ETH Zürich Distributed Application Layer (DAL) and in fact uses a mostly compatible XML format for graph specifications

• Compared to DAL, PRUNE however
  – Enables adaptive processing on OpenCL devices
  – Improved processing performance
  – Analyzable program graphs

• Multi-application support and distributed processing features of DAL are currently not implemented in Prune
PRUNE - References

• The main PRUNE repository

  https://gitlab.com/jboutell/Prune

• The TUT EffProc team

  http://www.tut.fi/effproc