OpenCL-Based Design Pattern for Line Rate Packet Processing

Jehandad Khan, Peter Athanas (Virginia Tech)
John Marshall, Skip Booth (Cisco Systems)
Programmable Packet Processor

Traditional switch

Control plane

Packets

Table mgmt

Control traffic

Data plane

P4-defined switch

Control plane

P4 Program

P4 table mgmt

Data plane

CHREC
NSF Center for High-Performance Reconfigurable Computing

VT
P4.org

P4 programs specify how a switch processes packets.
FPGAs for Packet Processing

• The *ideal* co-processor
  – Highly parallel
  – Arbitrary data paths
  – No cache delays
  – Low power
We ❤️ FPGAs
FPGAs for Packet Processing

• The *not-so-ideal* co-processor
  – Long compile times
  – Complicated design process
  – Less abundant expertise
  – Cost
We love FPGA Design
OpenCL for FPGA Design

• OpenCL simplifies the design problem
  – Programmable by a larger community
  – Simulation capability
  – Timing guarantees
  – Pipelining
  – Memory replication
  – Downside: limited expressiveness
Is OpenCL a good intermediate format?

• What is the achievable throughput?
• What are the tradeoffs?
• What are the design constructs we need?
OpenCL assumes a host / device model:

a. Host copies data to device
b. Host launches work on device
c. Device signals completion
d. Host copies data back

NOT SUITABLE FOR PACKET PROCESSING!
**Solution: “Persistent Kernels”**

**Launch-once-never-terminate kernels**

- Infinite loop in the kernel waits for data and processes it.
- OpenCL kernel
- Channel or OpenCL Pipe for input
- Output Channel realized as FIFOs on the FPGA
Overall Architecture

Based on *simple_router.p4*
**Match + Action Stage**

**Control Plane**
- Host Launches kernels to update state
- Update Kernel
- Updates

**Data Plane**
- Persistent Kernel listens on both channels
- State storage for persistent kernel
- Output Channel
- PHV Out

**Data Plane**
- local type_t entries[SIZE]
- Infinite Loop
- Match+Action Kernel

**Packet Header Vector (PHV) passed stage to stage**

**CHREC**
- NSF Center for High-Performance Reconfigurable Computing

**VT**
Match Engines in Prototype

1. One TCAM
   a. Longest Prefix Match

2. Two exact match engines
   a. Source MAC address
   b. Destination MAC address

All using on-chip RAM
Core first written in OpenCL, yet rewritten in Verilog (RTL)
Test Platform

Cisco UCS C240 server

Arria 10 DevKit

Altera Arria 10 AX115S2 FPGA
Capable of running at 70 Mpps
Follow up Work

• P4 -> HMC enabled FPGAs

Conclusion

• Using some clever tricks we can create a high-performance packet pipeline in OpenCL

• A high throughput design is possible
  – The design patterns can serve as guidelines for any data flow problem
  – Optimal use of on-chip resources is essential

• Performance portability ...