

One for All, All for One: A Heterogeneous Data Plane for Flexible P4 Processing

Jeferson Santiago da Silva¹, Thibaut Stimpfling¹, Thomas Luinaud¹, Bachir Fradj¹ and Bochra Boughzala²

¹Polytechnique Montréal, Canada, {firstname.lastname}@polymtl.ca, ²Kaloom Inc., Montréal, Canada, bohra@kaloom.com

Abstract—The P4 community has recently put significant effort to increase the diversity of targets on which P4 programs can be implemented. These include fixed function and programmable ASICs, FPGAs, NICs, and CPUs. However, P4 programs are written according to the set of functionalities supported by the target for which they are compiled. For instance, a P4 program targeting a programmable ASIC cannot be extended with user-defined processing modules, which limits the flexibility and the abstraction of P4 programs.

To address these shortcomings, we propose a heterogeneous P4 programmable data plane comprised of different targets that together appear as a single logical unit. The proposed data plane broadens the range of functionalities available to P4 programmers by combining the strength of each target. We demonstrate the feasibility of the proposed P4 data plane by coupling an FPGA with a soft switch which emulates a programmable ASIC. The proposed data plane is demonstrated with the implementation of a simplified L2 switch. The emulated ASIC match-table capacity is extended by the FPGA by an order of magnitude. The FPGA also integrates a proprietary module using a P4 extern.

Index Terms—P4, Heterogeneous systems, FPGA

I. INTRODUCTION

Data center applications have pushed the envelope towards high-throughput yet programmable network devices. Such a trade-off can be alleviated by recent advancements in programmable ASICs and network programming languages as P4 [1]. However, to achieve higher throughput, these programmable ASICs limit the functionalities that can be expressed in P4 due to tighter hardware constraints. For instance, integrating new proprietary modules with P4 externs is not supported on a programmable ASIC.

In addition, standalone hardware platforms supporting P4, shown in Table I, exhibit an inherent trade-off between the degree of programmability and performance. To overcome these limitations, complementary P4 targets are required.

TABLE I
AVAILABLE P4 TARGETS

Target	Degree of Programmability	Throughput	Latency	Power Efficiency
ASIC	Limited	Very high	Very low	Very High
FPGA	High	High	Low/very low	High
NIC	High	Limited	Limited	High
CPU	Very high	Very low	Very high	Low

The first four authors have equally contributed to this work.

This work is supported by CNPq/Brazil, Mitacs/Canada, and Kaloom Inc.

While some research has been conducted to diversify the number of P4 targets [2], little work has been devoted to exploiting the strength of complementary targets. We believe that combining multiple targets into a heterogeneous hardware platform allows filling this gap.

In this paper, we introduce the concept of a heterogeneous data plane (HDP) platform. As a proof of concept, we present a HDP comprising an ASIC and an FPGA. By coupling a programmable ASIC and an FPGA, we extend the set of functionalities that can be described using P4. The FPGA programmability enables the implementation of custom P4 pipelines, which covers proprietary modules expressed as P4 externs, while increasing hardware resources.

In contrast to previous works that have used FPGAs only as hardware accelerators [3], the FPGA is here integrated as a main component of a logical P4 pipeline.

The main contribution of this work is to start a discussion towards heterogeneous data-plane platform. We present a proof of concept that:

- Combines multiple targets into a logical P4 pipeline.
- Exploits the programmability characteristics of FPGAs to extend the limitations of an ASIC.
- Implements a proprietary module as a P4 extern.

We believe that HDP will open new doors to more innovative P4 programs and network applications.

II. HETEROGENEOUS DATA-PLANE

A heterogeneous data plane (HDP) comprises different P4 targets. Loosely coupling diverse targets extends the range of functionalities and applications that can be expressed in P4.

A generic view of a HDP is depicted in Fig. 1. In this figure, multiple P4 targets are connected together to form a single logical P4 pipeline. Similarly, a control-plane entity manages the HDP. A compiler tool chain splits the original P4 description across the different targets. The proposed architecture is described in the following subsections.

A. Hardware and Communication Infrastructure

The proposed HDP is composed of multiple P4 targets abstracted into a single logical P4 target. Hence, a HDP can combine any set of P4 targets, from x86 servers, GPUs, NICs, to FPGAs and ASICs. One HDP example is presented in the rounded rectangle of Figure 1.

The communication links between the P4 targets of a HDP does not require any specific protocol nor medium.

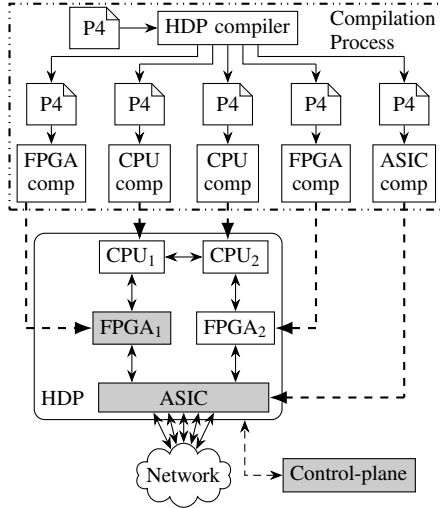


Fig. 1. Overview of a Heterogeneous Data-Plane (HDP)

Since the proposed HDP is programmed with P4, there is no communication protocol requirements between targets. The only constraint lies in the feasibility of describing the chosen protocol using P4 idioms. Thus, to connect multiple targets in a single HDP, different media may be used, ranging from PCIe, Ethernet-capable links, QPI, to proprietary communication interfaces. For instance, GPUs, or FPGAs connected to a server are likely to use a PCIe link, while an ASIC connected to an FPGA could use an Ethernet link.

B. Compilation Process

As shown in the dashed rectangle of Fig. 1, the HDP compiler splits a P4 program into multiples target-specific P4 programs. Each generated P4 program implements only a portion of the functionality expressed in the original P4 program. The challenge of HDP compiler is to efficiently partition the original P4 program and to map each functionality to a specific target, based on performance constraints and user needs given in a configuration file.

III. PROOF OF CONCEPT

As a proof of concept, we propose a HDP highlighted in grey boxes of Fig. 1. The proposed HDP contains an ASIC that only includes basic P4 functionalities and an FPGA. On this HDP, we have implemented a L2 switch with a packet counter. The HDP compiler is not covered in this work as we have manually mapped the P4 program to each target.

The L2 switch is divided into the ASIC and the FPGA. Hence, the L2 MAT entries are divided between the ASIC and the FPGA. In addition, the packet counter, implemented on the FPGA as a P4 extern, computes the number of packets matched on the FPGA. Thus, the FPGA extends an unsupported ASIC feature. In addition, an external DDR3 DIMM connected to the FPGA increases by an order of magnitude the size of the L2 MAT compared to an ASIC.

Because we had no access to a programmable ASIC, the bmv2 soft switch was used as an ASIC emulator. The bmv2

TABLE II
QUALITATIVE PLATFORM EVALUATION

Platform	Generic externs	Extensible M-A Tables	Line-Rate Processing
FPGA	✓	✓	×
ASIC	×	×	✓
Proposed work	✓	✓	✓

runs on a x86 server equipped with a 10 Gb/s Intel NIC. The FPGA P4 pipeline is compiled by Xilinx SDNet 2017.4, synthesized, and implemented using Xilinx Vivado 2018.1 on a Xilinx ZC706 evaluation board. The bmv2 and FPGA are connected using an Ethernet link. A simplified controller runs in an x86 computer and communicates with the proposed architecture using a generic API. Qualitative results achieved by the proposed HDP are presented in Table II.

IV. DISCUSSION AND RESEARCH DIRECTIONS

This section presents the challenges and research directions related to heterogeneous data planes.

One challenge faced by a HDP relates to mismatched target performances. For instance, a lookup engine using an external memory connected to an FPGA may not match the lookup rate of an ASIC. Thus, the throughput of a HDP could be limited by a single target. However, in the proposed proof of concept, this limitation can be alleviated by using caching strategies. That is, the external memory connected to the FPGA would be seen as the main memory, while the ASIC memory would be used as a cache. Another solution can employ multiple parallel FPGA pipelines.

In addition, a HDP introduces resources overhead. One resource overhead relates to the replication of parsers and deparsers on the HDP targets. Finally, the HDP can introduce a variable latency as a packet may traverse multiple targets. However, these impacts may be reduced by using custom protocols to communicate between the HDP targets.

Still, we believe the new ideas allowed by a HDP outweigh its overhead. As future work, we plan to develop the compiler tool-chain, which is the key element to leverage the HDP. Lastly, we hope that HDP ideas will be assessed and challenged by P4 enthusiasts to refine the proposed platform.

ACKNOWLEDGMENT

We would like to thank the anonymous reviewers for their comments and our shepherd Prof. Pierre Langlois.

REFERENCES

- [1] P. Bosshart *et al.*, “P4: Programming Protocol-independent Packet Processors,” *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 3, pp. 87–95, Jul. 2014. [Online]. Available: <http://doi.acm.org/10.1145/2656877.2656890>
- [2] H. T. Dang *et al.*, “Whippersnapper: A P4 Language Benchmark Suite,” in *Proceedings of the Symposium on SDN Research*, ser. SOSR ’17. New York, NY, USA: ACM, 2017, pp. 95–101. [Online]. Available: <http://doi.acm.org/10.1145/3050220.3050231>
- [3] A. Caulfield, P. Costa, and M. Ghobadi, “Beyond SmartNICs: Towards a Fully Programmable Cloud,” in *IEEE International Conference on High Performance Switching and Routing*, ser. HPSR ’18. IEEE, 2018, p. to appear. [Online]. Available: https://www.microsoft.com/en-us/research/uploads/prod/2018/05/beyond_smart_nics.pdf