MACSAD: Multi-Architecture Compiler System for Abstract Dataplanes (aka Partnering P4 with ODP)

P Gyanesh Patra, Christian Rothenberg
University of Campinas (UNICAMP)
Campinas, Sao Paulo, Brazil
gyanesh,chesteve@dca.fee.unicamp.br

Gergely Pongrácz
TrafficLab, Ericsson Research
Budapest, Hungary
gergely.pongracz@ericsson.com

ABSTRACT

Software Defined Networking (SDN) strives for deep programmable hardware and software dataplanes without giving up on performance. Domain Specific Languages (DSL) such as P4 seek to provide top-down high-level capabilities to define the datapath pipeline agnostic to the network platform and independent from any network protocols. At the crossroads, bottom-up industry efforts at the OpenDataPlane (ODP) initiative are pursuing open-source multi-architecture APIs for dataplane programmability across various networking platforms. Towards P4 code reuse for various targets (portability), we propose MACSAD as a compiler system that brings together the higher-level P4 language and the abstract, target-independent ODP APIs. The demo showcases two P4 applications compiled into heterogeneous datapath platforms supporting ODP.

CCS Concepts

•Software and its engineering → Domain specific languages; •Hardware → Emerging languages and compilers;

Keywords

Software Defined Networking; P4; OpenDataPlane

1. INTRODUCTION

The evolution of SDN is driving research and product developments on protocol independent abstract data planes. Approaching the challenge top-down, enabling advances on Domain Specific Languages (DSL) include P4 and POF. Bottom-up from the datapath perspective, so-called Protocol Independent Switch Architectures (PISA) chips are entering the programmable networking hardware scene. While P4 [2] is well-known for providing high-level abstractions that allow dataplane programmability agnostic to the hardware target, the backend compilation tasks are still very much open and less understood, pushing the complexity down the stack to be solved on a per-target basis.

OpenDataPlane (ODP) [1] is a less-known, recent initiative working on an abstract API specification covering functional needs of data plane applications. The vendor and platform neutral APIs span common features across multiple targets turning application written with ODP APIs portable. Vendors can implement ODP APIs in an optimized manner according to their targets while abstracting the hardware acceleration features (e.g., Crypto). ODP is supported by multiple platforms comprising of x86, ARM and various other SoC architectures.[1] ODP can be viewed[2] as a set of common, unifying APIs of higher abstraction than DPDK, which becomes a specific API implementation.

Towards code portability and in line with the P4 evolution roadmap on architecture-language separation to reuse the same compiler for new targets, We developed MACSAD[3] with the main goal of bringing P4 and ODP together to enable developers to write P4 programs seamlessly portable across network platforms while transparently leveraging hardware acceleration capabilities. By translating P4 into ODP APIs, which are high-enough level to allow platform abstraction without imposing strict models and overheads, we overcome the hazards of developing and maintaining target-specific heterogeneous backends without compromising performance and hardware-acceleration options.

Demo contributions. We show the ability of MACSAD to compile two P4 programs (L2-FWD and VxLAN-GW) into portable dataplane applications using ODP

SIGCOMM '16, August 22–26, 2016, Florianopolis, Brazil

DOI: http://dx.doi.org/10.1145/2934872.2959077

[1] http://www.opendataplane.org/downloads/
[2] A useful analogy is comparing ODP to OpenGL, but for networking instead of video graphics.
[3] It is pronounced as 'Maksad' which means *purpose* or *motive* in Hindi. It is also simply referred as "MAC".

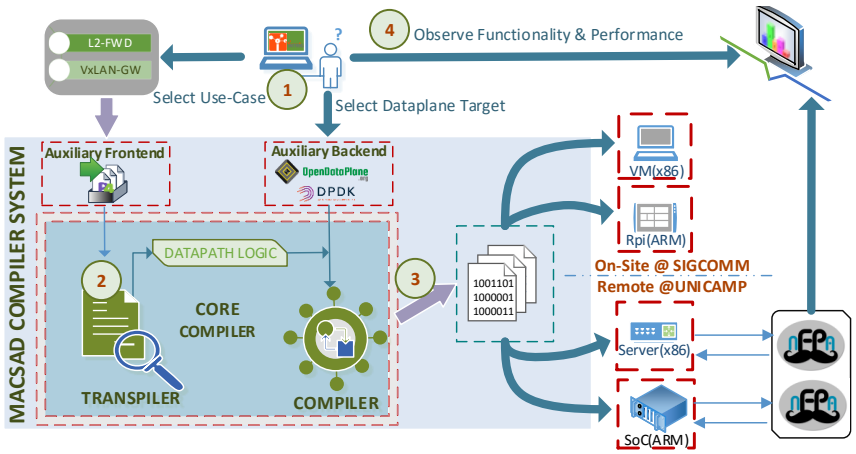

Figure 1: MACSAD Architecture & Use Case Demonstration Workflow.

APIs over different targets (x86, x86+DPDK, ARM-SoC). Demo attendants can select the use case and datapath target, learn about the compilation process, and observe the functional and performance behavior.

2. MACSAD

The Multi-Architecture Compiler System for Abstract Dataplanes is architected to achieve seamless portablity of dataplane applications written in a DSL (P4 being our starting focus) and implementable on a wide range of platforms efficiently. The following design objectives are guiding our efforts on MACSAD: (1) Fast and easy development environment of dataplane applications by using P4; (2) Portability of dataplane application across different network platforms by leveraging ODP APIs; (3) Dynamic, flexible pipeline by supporting protocol independent DSL (P4) in programmable targets (ODP).

Targeted to programming network datapath pipelines, MACSAD (Figure 1) is based on 3 main modules:

Auxiliary Frontend: A plugin framework to include/import different frontend DSLs –with P4 language being the initial choice and focus of this demo.

Auxiliary Backend: Binds target-specific SDKs in order to support different platforms –with ODP being the premier choice because of its cross-platform nature.

Core Compiler: A Transpiler component acts as a source-to-source compiler from the Auxilary Frontend plugin (standard P4 compiler and HLIR) into the Datapath Logic (our IR choice implemented in plain C) defining how the GCC-based compiler generates the the dataplane target binary code using ODP APIs.

3. DEMONSTRATION

Workflow. The demo is divided into 4 steps (see Fig. 1):
(1) User selects the P4 dataplane application (L2-FWD, VxLAN) and the target platform (on/off-site, x86, ARM).
(2) The *Transpiler* automatically generates the Datapath Logic IR from the P4 program.
(3) The *Compiler* module compiles the Datapath Logic IR using the ODP APIs for the selected target.
(4) Finally, the user interacts with the dataplane application to verify its functionality and performance.

Dataplane Target Platforms. Portability will be showcased using the following platforms: (*i*) Virtual Machine (x86) on commodity laptop and (*ii*) Raspberry Pi 3 (ARM) at SIGCOMM location, (*iii*) high-end server (x86 w/DPDK).

Experimental evaluation. The two use cases are evaluated using a two host topology connected to the MACSAD-compiled datapath DUT. The Network Function Performance Analyzer (NFPA)[4] is used for consistent benchmarking purposes and results are displayed in a Web GUI. User specified arguments include the list of interfaces and the number of CPUs to be used.

4. CONCLUDING REMARKS

Our ongoing work on MACSAD represents a promising approach towards dataplane program portability by transparently compiling the high-level P4 language into just enough low-level platform-independent code using the ODP APIs. Future work includes supporting additional DSL (e.g. POF, ONF PIF), exploiting multi-core and hardware acceleration capabilities of chips. We expect to contribute to missing pieces in ODP as well as the P4 evolution as needed while implementing more complex use cases (e.g., BNG, vEPC).

Acknowledgments

This work was supported by the Innovation Center, Ericsson Telecomunicações S.A., Brazil.

5. REFERENCES

[1] *Opendataplane [Online]. Available: http://opendataplane.org.*

[2] P. Bosshart et a. P4: Programming Protocol-independent Packet Processors. *SIGCOMM CCR.*, 44(3):87–95, July 2014.

[4] http://ios.tmit.bme.hu/nfpa/