P4-API Working Group
Annual Wrap-Up

Chris Sommers, Distinguished SWE, Keysight Technologies
Steffen Smolka, Software Engineer, Google

P4 API Working Group - Overview

The P4-API WG maintains the P4-Runtime Specification:

- Written Specification
- Protobuf schema (.proto & generated collaterals: Python, golang)
- CI/CD builds to ensure .protos are good, render the docs, etc.

Current specification version is 1.4.0-dev

GitHub: https://github.com/p4lang/p4runtime

Latest versions:

- https://p4.org/p4-spec/docs/p4runtime-spec-working-draft-html-version.html
- https://p4.org/p4-spec/docs/p4runtime-spec-working-draft-pdf-version.html

Meets every four weeks 9:30am Pacific.

Community details (e-mail, Calendar, Slack): https://github.com/p4lang/p4runtime#community

P4 API Working Group – 2022-23 Highlights

It was a relatively calm year –11 Pull Requests were merged. **The spec is pretty stable!**

A few small .proto changes:

- Add selector_size_semantics to ActionProfiles
- Deprecate egress_port (uint32) in favor of egress (bytes) in Replica

Seven small changes/clarifications to the written specification

Various open issues under discussion – clarifications, enhancements, etc.

Changing of the Guard

- Founding co-chair Antonin Bas has “retired” as co-chair (but is still in the WG).
  Thanks for all the hard work and dedication!
- Chris Sommers joined as co-chair Apr 2023

P4Runtime – Use-Cases & Future Direction

- SDN Control of Switches – Stratum, SONiC-PINS
- SmartNIC, DPU, IPU – IPDK Project
- SONiC-DASH Behavioral Model (bmv2, P4-DPDK) – SAI wrapper around P4Runtime allows SONiC integration

Future Possibilities:

- xPUs present challenges of scale – huge table sizes (millions of entries). P4Runtime may need to evolve to optimize the control-plane performance.
- Accommodating P4-TDI (Table-driven Interface), a generic abstraction to support architecture-specific constructs.
- Relationship to OPI (Open Programmable Infrastructure) project networking use-cases – to be explored.

P4Runtime at Google

- P4Runtime-based switches used at Google
  - P4 Integrated Network Stack (PINS) - SONIC + P4Runtime
- Initial focus on data center
  - Top-of-Rack Switches
  - CLOS Switches
  - Fabric Border Routers
- P4Runtime supersedes OpenFlow at Google. Advantages:
  - **Generic** - works for ***any pipeline*** and ***any headers***
  - **Precise** - behavioral, machine-readable semantics via P4 program
  - ➔ **Amenable to automation** - immediate, automated validation when P4 program changes
- To learn more:
  - **Code:** github.com/sonic-net/sonic-pins
  - **SIGCOMM paper:** *SwitchV: Automated SDN Switch Validation with P4 Models*
  - **P4 workshop talks:**
    - *Keynote* by Parveen Patel
    - *Escaping Babel - The Flow Must Go On* by Victor Rios

Thank You!

Chris Sommers,
Keysight Technologies
Steffen Smolka, Google