MAY 18-20
2021 P4 Workshop
Hosted by ONF

Can P4 run on “switchy” CPUs?

Cristian Dumitrescu, SW Architect, Intel
Han Wang, SW Architect, Intel

P4-DPDK

- Run P4 programs on multi-core CPUs.
- Develop better and faster vSwitches by combining the P4 language flexibility with the DPDK performance.

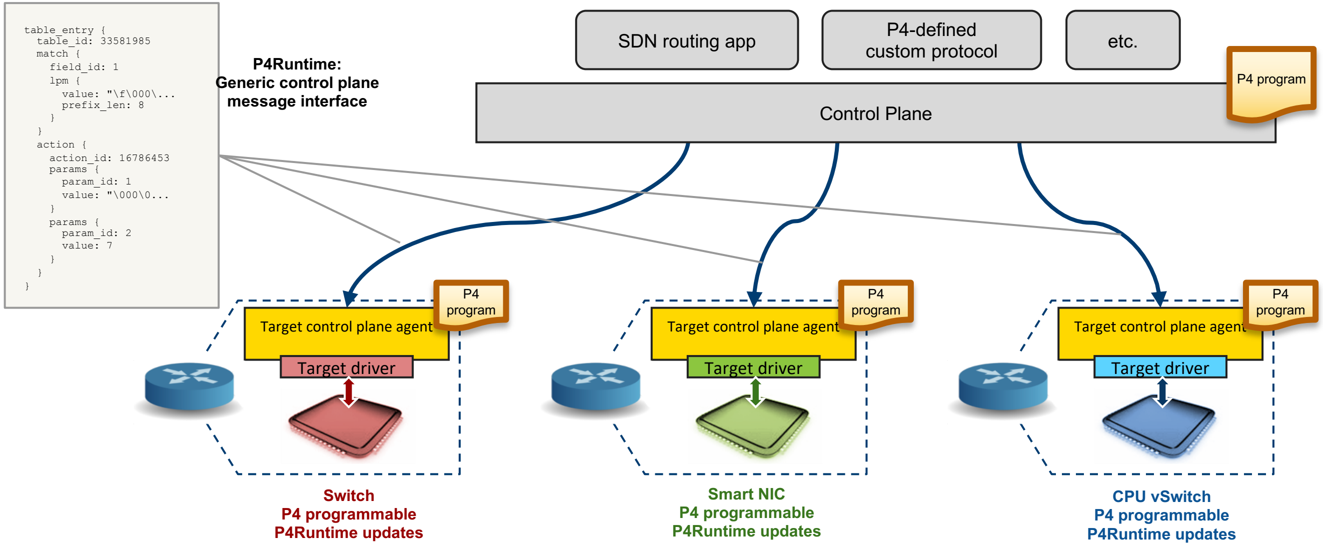
P4-DPDK

- Open-source project on p4.org and dpdk.org
  - P4C compiler: https://github.com/p4lang/p4c/tree/main/backends/dpdk
  - DPDK back-end: http://git.dpdk.org/dpdk/tree/lib/pipeline
  - DPDK Target agent: http://git.dpdk.org/dpdk/tree/examples/pipeline

Why CPU target for P4? Flexible infrastructure.

- The vSwitches on the server CPUs (< 1Tbps) are the smaller siblings of the DC switches (N x Tbps).
- Unify all switch pipelines (switch, NIC, CPU) under the same P4 environment for better integration and productivity.
- Develop custom vSwitches for VMs and container plumbing.

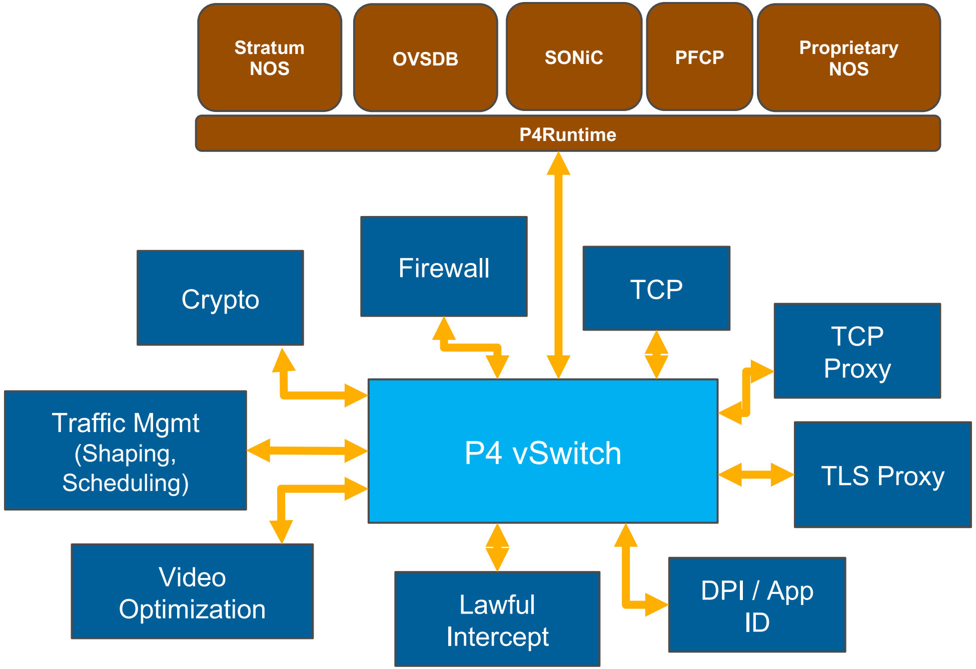
Why CPU target for P4? Flexible infrastructure.



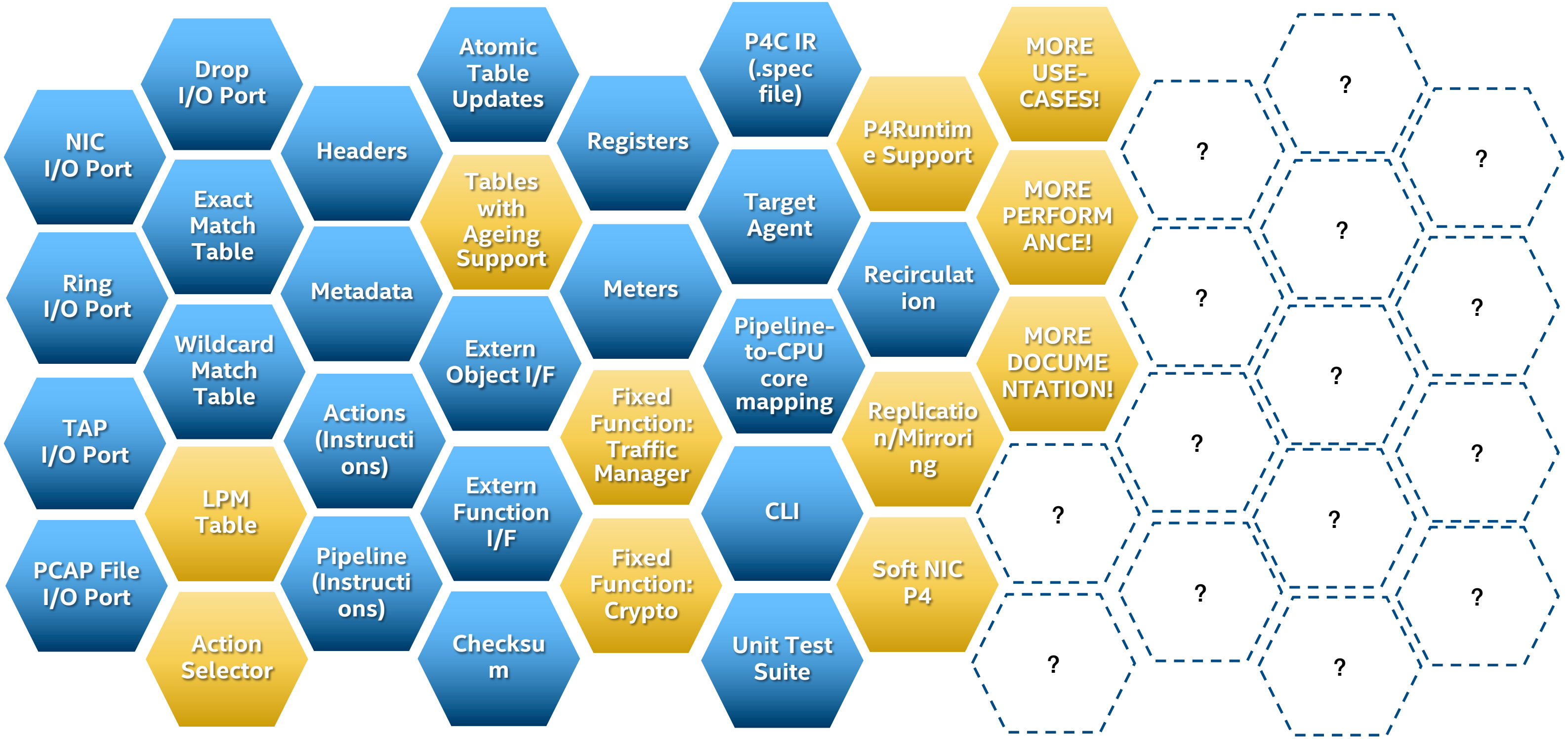
```
table_entry {
  table_id: 33581985
  match {
    field_id: 1
    lpm {
      value: "\f\000\...
      prefix_len: 8
    }
  }
  action {
    action_id: 16786453
    params {
      param_id: 1
      value: "\000\0...
    }
    params {
      param_id: 2
      value: 7
    }
  }
}
```

Why CPU target for P4? Better network stacks.

- Keep the data plane short and crisp in P4!
- All header processing should be part of the P4 vSwitch, everything else is plugged in as a port in the P4 switch.
- The P4 switch also connects with NICs and accelerators.
- Nothing is predefined, only what is actually used is defined: protocols, meta-data, tables, actions, data path, etc.
- Classification, ACLs, FIB, tunnels, NAT, meters, markers, stats, etc.
- Decoupled control plane manages the P4 vSwitch through the standard P4Runtime RPC interface.



P4-DPDK Feature Set



NIC I/O Port

Ring I/O Port

TAP I/O Port

PCAP File I/O Port

Drop I/O Port

Exact Match Table

Wildcard Match Table

LPM Table

Action Selector

Headers

Metadata

Actions (Instructions)

Pipeline (Instructions)

Atomic Table Updates

Tables with Ageing Support

Extern Object I/F

Extern Function I/F

Checksum

Registers

Meters

Fixed Function: Traffic Manager

Fixed Function: Crypto

P4C IR (.spec file)

Target Agent

Pipeline-to-CPU core mapping

CLI

Unit Test Suite

P4Runtime Support

Recirculation

Replication/Mirroring

Soft NIC P4

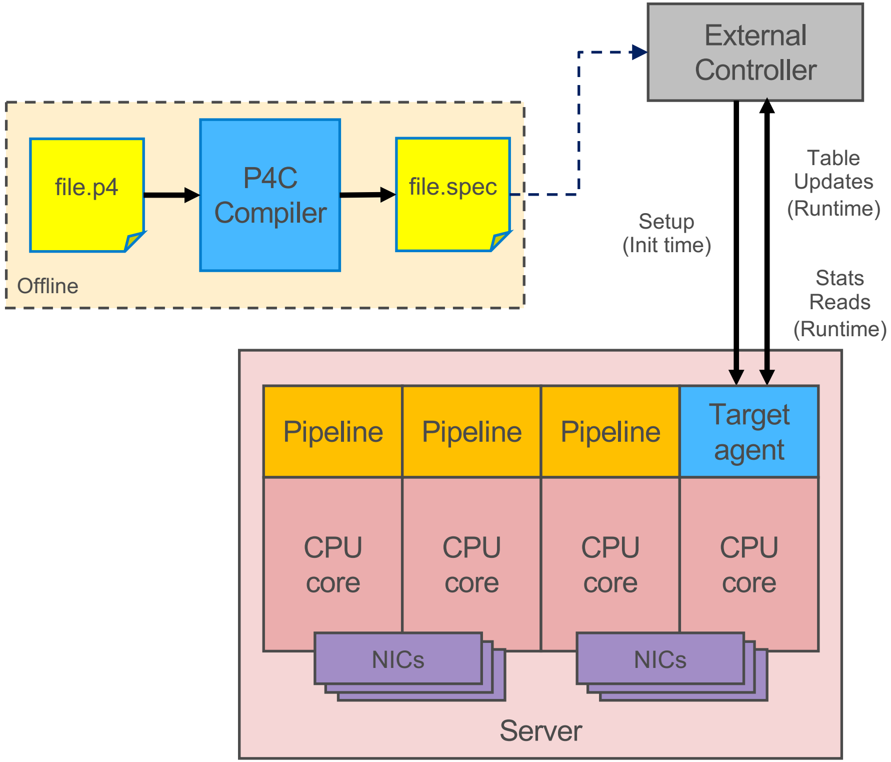
MORE USE-CASES!

MORE PERFORMANCE!

MORE DOCUMENTATION!

How does P4-DPDK work?

- Offline process: The .p4 file is translated to an intermediate representation .spec file.
- The .spec file defines the P4 objects to be created for the pipeline (headers, meta-data, actions, tables, data path, etc).
- The P4 actions and apply blocks are translated to instructions from a pre-defined set. The instruction operands are the P4 objects defined above, not low-level x86 instructions. Essentially, P4-DPDK is a P4 virtual machine.
- The external controller connects to the target agent to load the P4 blob (the .spec file) and for run-time table updates and queries (e.g. stats).
- The target agent creates the P4 objects for each pipeline at initialization based on the .spec file and maps each pipeline to a CPU core.
- Each CPU core executes one or multiple pipelines by running the associated instructions for each input packet. Each pipeline is single threaded.



Call to action

- The P4-DPDK project needs you!
- Please implement your use-cases in P4, run them with P4-DPDK, report issues and workarounds. Code patches with bug fixes and new features are also highly appreciated ☺.

MAY 18-20
2021 P4 Workshop
Hosted by ONF

Thank You

cristian.dumitrescu@intel.com
han2.wang@intel.com