P4 for NIC - A cloud provider perspective

Hariharan Thantry, Google

Google Infrastructure, a million ft view!

- Consist of datacenters that interconnect many racks of machines with a non-blocking CLOS fabric design, with a centralized SDN controller for programming the forwarding paths.
- Core fabric optimized for latency reduction in forwarding packets.
  - No complex packet transformations (encap/decap) or policies (ACLs) inside the core network.

*Google cloud **shares** this infrastructure with the rest of Google!*

Google Cloud Networking - the host elements

- SNAP is an OS Bypass user space packet processing framework that is used to build packet processing pipelines.
- Andromeda on SNAP is the end host cloudstack that enables VM/container connectivity to other VMs, Google services, internet.
- Congestion control stacks run on SNAP to enforce rate limiting, pathing alternatives for egress.
- SNAP co-exists with Linux networking that drives a lot of native Google traffic.

Multiple use cases, multiple collaborative stacks!

Google Cloud Networking - the business case for DPUs

- Any CPU cycle spent on I/O packet processing (networking, storage) is a cycle less for VMs.
- All host memory used for storing flow state is less memory that can be made available to VMs.
- Supporting certain use cases (Baremetal) means that we don’t have any software running on the host complex.

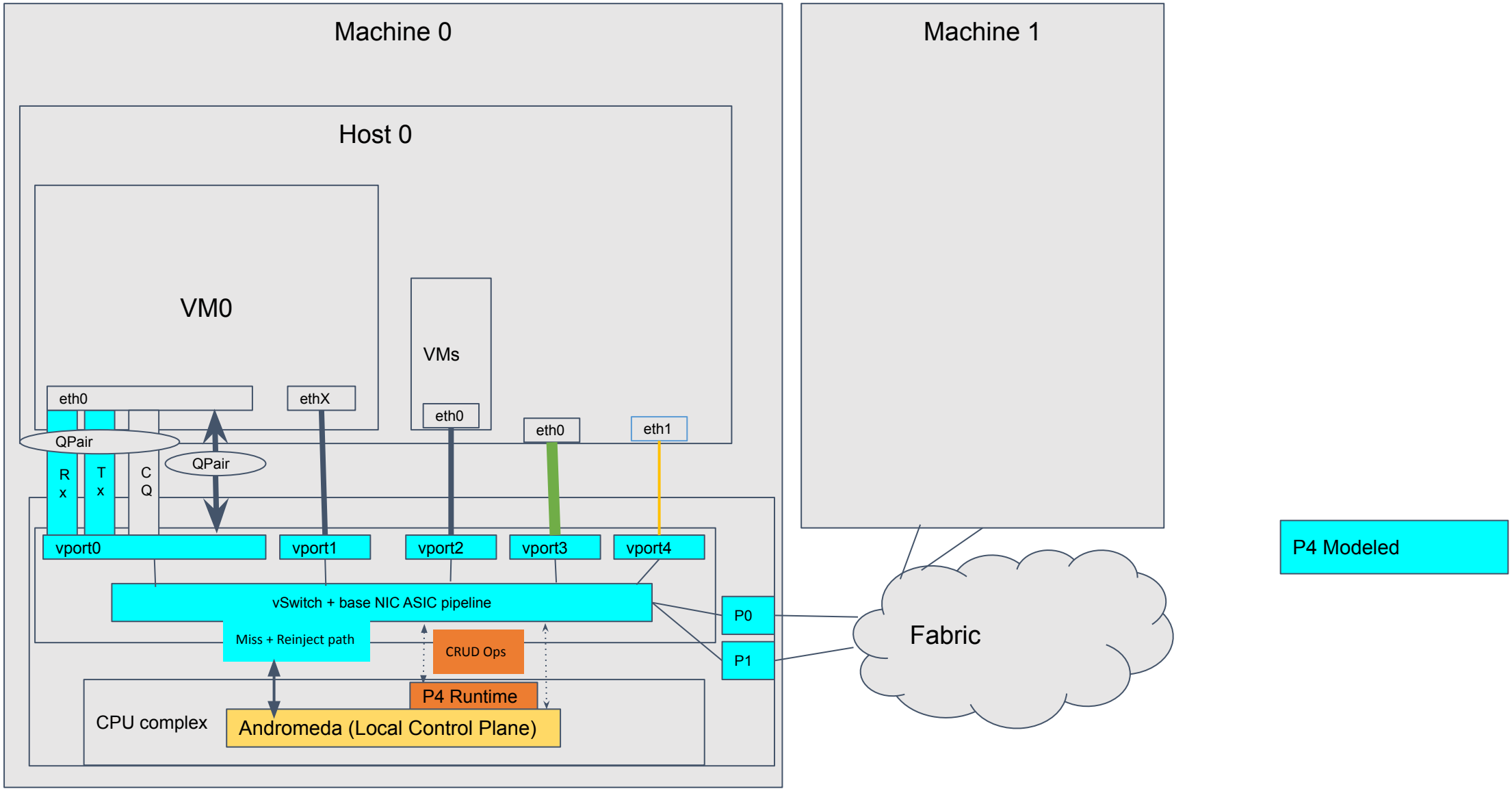
*Offloading packet processing to SmartNICs/DPUs for cloud vendors makes **business** sense!*

DPUs and P4

- Need standardized mechanisms to express the universe of packets, and their transformations.
- P4 is an attractive framework, proven in the switch world, but…

*“NICs are an **endpoint** element, routers are **transit** elements.”*

DPUs and P4*



*Highly simplified view. Endpoint offload blocks (TSO, LRO, Encrypt/Decrypt, Memories not shown)

P4 and NICs – Language extensions*

- Descriptors form a way to describe work to the pipeline.
  - Having a standardized descriptor as part of arch definition useful (enables metadata standardization for commonly available offloads like TSO, LRO, RSS, Encrypt/Decrypt).
  - Machine model standardization (e.g. metadata sizing, visibility, and host-device handshake mechanisms) for NIC especially useful.
  - Per vport parse graphs, security contexts (VM should not be able to spoof a host).
- Dataplane Events that can be queued as a first class language construct.
  - Endpoint needs to be notified of packet transmission through completion events.
  - Mechanism to model pipeline notifications (e.g. ageing, add-on-miss, counter overflow, network anomalies)
- Encryption as a core feature for PNA, and hardware friendly feature expressibility.
- Expressing connection tracking that is hardware friendly, and a first class language construct!
- Expressing (static) priorities for P4 tables (for guiding cache replacements).

"Greater industry/academia participation in extending the language for NICs to drive faster adoption"

*Some of these are being discussed in Portable NIC Architecture forum

P4 and NICs – ecosystem/standardization

- Investments in ecosystem improvement for running P4 on NIC programs.
  - DPDK, BMv2, an open-source compiler targetable ASIC software model?
  - Automated test packet generation/table entries from P4 programs?
  - P4Runtime or TDI as the P4 programming interface of choice for NIC?
    - Performance, toolchain (e.g. DPDK) support, and ecosystem adoption key discriminants.
  - More fixed function blocks (compression, encrypt, shaping etc).
- More long-term academia involvement in improving language expressibility, test generation frameworks, collaborations on developing lower level IRs for targeting ASIC family architectures.

"Ecosystem investments leads to better standardization, improves cloud adoption and lowers TTM!"

Thank You

“Evolving P4 language and ecosystem to express SmartNIC features is hugely relevant to Google.”

https://cloud.google.com/products/networking