DPDK support for new hw offloads

Alejandro Lucero, Netronome

DPDK Summit - San Jose – 2017
Netronome Agilio SmartNIC: a highly programmable card designed for network packet/flow processing

- 120 Flow Processing cores
- Hardware accelerators: crypto, hash, queue, LB, TM
- Hardware offloads: checksum, VLAN, TSO, IPSec, …, OVS, eBPF, P4, Contrail vROUTER, virtio
- 10G, 25G, 40G, 100G
- Up to quad PCIe Gen3x8
DPDK support for new hw offloads

OVS

KERNEL

Orchestrator

OVS

vhost-net

OVS-kernel

NIC driver

NIC

VM

virtio-net

user

kernel

HW
DPDK support for new hw offloads

OVS KERNEL OFFLOAD

Orchestrator

OVS

OVS-kernel

NFP PF netdev

Flow offload

VF

PF

NFP

OVS

WIRE

user

kernel

HW

Repr netdev
Repr netdev
Repr netdev
Repr netdev
DPDK support for new hw offloads

OVS
KERNEL OFFLOAD
+ (Netronome) XVIO

user
kernel

HW

Flow offload
OVS-DPDK:

- Better performance than (kernel) OVS
- Consumes CPU in the Host. Scalable?
DPDK support for new hw offloads

OVS-DPDK

VM virtio-net
VM virtio-net
VM virtio-net
VM virtio-net
Orchestrator
OVS-DPDK PMD
NIC

user
kernel
HW
DPDK support for new hw offloads

OVS-DPDK &
SR-IOV
OVS-DPDK: offload?

- Partial offload proposed in the OvS mailing list (just classification giving hints for action to OvS)
- Full (classification + action) Offload? Does it make sense?
  - VMs using SR-IOV (native NIC performance)
  - OVS-DPDK needs CPUs. With offload CPU just for slow path
  - Different tenants, different service: virtio AND SR-IOV
  - Security
  - Just experimental work done (Netronome)
DPDK support for new hw offloads

OVS-DPDK
FULL OFFLOAD
DPDK support for new hw offloads

OVS- DPDK FULL OFFLOAD (optional)
DPDK support for new hw offloads

Virtual ports (Representors) packet delivery (slow path)

Representors

Multiplexed PF PMD based on metadata
OVS-DPDK Offload: what is needed?

- Representors PMDs could be created inside PF PMDs, but ...
  - hotplug/unplug: representors are not PCI devices
  - Transparency: representors naming
  - Who is taking over the PF? Bifurcated driver?

- OVS Flow rules offload?
  - Changes to OVS-DPDK? Using TC through PF?
  - Is rte_flow enough for OVS flows syntax?
eBPF offload

BPF: Berkeley Packet Filter (tcpdump, libpcap, netfilter)

Kernel executes BPF programs via in-kernel virtual machine

eBPF: extended BPF. Sockets filtering and tracing (since 3.18)

Attaching eBPF programs to kernel TC classifier (since 4.1)

XDP: eXpress Data Path
DPDK support for new hw offloads

XDP (eXpress Data Path) in the Linux kernel

Bare metal packet processing at the lowest point in the software stack

It does not require any specialized hardware

It does not required kernel bypass

It does not replace the TCP/IP stack

It works in concert with TCP/IP stack along with all the benefits of BPF (eBPF)
DPDK support for new hw offloads
XDP/eBPF & DPDK

Do we need XDP/eBPF in userspace networking? How to do it?

Good for being “kernel compatible”: executing eBPF/XDP programs, but …

Can eBPF-DPDK be eBPF-kernel compatible?

Likely good for any DPDK-based network stack

Support at PMD level with offload option

It is already possible (with limitations) to use eBPF in userspace
eBPF Offload

XDP consume host resources (CPU, PCIe bandwidth)

Netronome’s NFP: Packet processing through eBPF programs with hardware offload

IOvisor: eBPF to the extreme
DPDK support for new hw offloads

Userspace

Network Stack

NIC DRIVER

XDP

PCle

NIC HW

eBPF program

Userspace

Network Stack

NIC DRIVER

XDP

PCle

NIC HW

eBPF

NFP

DROP, FORWARD

HOST CPU

Kernel eBPF Offload

DROP, FORWARD
DPDK support for new hw offloads

**virtio Offload: virtio capable NIC**

- VMs with SR-IOV (device passthrough) but using virtio interface
  - Pros: VM provisioning, performance
  - Cons: VM migration, East-West traffic

- VM migration: requires a migration friendly NIC
- East-West traffic: memory vs NIC
- DPDK: virtio changes (vhost), iommu changes????
- Other option: vDPA (vHost Data Path Acceleration)
Dataplane Acceleration Developer Day (DXDD)

- Date: December 11-12 (Monday & Tuesday)
- Time: 8:30 a.m. – 8:00 p.m.
- Location: Computer Science Museum (Mountain View, CA)
- Why should you attend?
  - Discussions about recent dataplane acceleration development
    - P4-16 introduction
    - TC offload introduction
    - eBPF introduction
  - Extensive hands-on training
    - P4-14 labs
    - TC labs
- Register: [https://open-nfp.org/dxdd-2017](https://open-nfp.org/dxdd-2017)
Questions?