TC Classification with Open vSwitch

Simon Horman
Motivation

Would like to:

- Partially or fully offload Open vSwitch
  - Software: e.g. TC
  - Hardware: e.g. iNIC
- Do so using mechanisms present in upstream
Importance of Offloading

OVS L2/L3 Forwarding to 8 VMs with 64K Flows

- 100 Wildcard Rules
- 1000 Wildcard Rules
- 10000 Wildcard Rules
- 64000 Wildcard Rules

5X Throughput Improvement + 50% CPU Savings

- 12 CPU Cores
- 8 CPU Cores
- 1 CPU Core

OVS Offload to iNIC with PMD to VMs
eBPF TC Classifier and Actions

- eBPF programs supplied to kernel by user-space
- Data may be shared with user-space using eBPF maps

```
User-Space
  Helper
  query, update,... load

Kernel
  Map  Prog
  eBPF Classifier
  from prior processing to further processing
```
eBPF Offload

- eBPF seems well suited to offloading to programmable hardware
  - May be interpreted or;
  - JITed and run natively
  - Mechanism for very fast and flexible packet handling
TC Offload Possibility

- One possibility is to add “ingress -1” support to TC
  - TC qdisc may be added before Ingress
  - Managed by Kernel; Executed in Hardware
Hinting

- Mechanism for offloading Open vSwitch classifier
- Offload classifies packet and tags it with metadata
- Open vSwitch performs flow lookup using metadata
Modest Proposal for Hinting

- TC Classification → metadata added as skb mark
- Open vSwitch looks up flows using skb mark
- eBPF TC Classifier allows eBPF maps to synchronise flow/mark mapping between TC classifier and Open vSwitch
- And the eBPF may be offloaded to hardware
Hinting with eBPF Classifier

User-Space

- ovs-vswitchd
  - query, update...
  - load
  - flows, stats, misses...

Kernel

- Map
- Prog
- eBPF Classifier
  - packet with skb-mark
- OVS Datapath
  - packet

Hinting Packet Processing Path
Offloaded Hinting

User-Space

Kernel

iNIC

ovs-vswitchd

flows, stats, misses…

packet

query, load

update,…

packet with skb-mark

packet

OVS Datapath

Map

Prog

eBPF Classifier
Research at Netronome

Investigating:

- Integration of TC classifier with eBPF
- Hardware offload of Kernel OVS datapath using switchdev
- User-space driven hardware offload of OVS using match-interface
- ...
Questions

Thanks to the community for many of the ideas presented here and thanks for your time
Bonus Slides
Possible Flow Handling

- Makes use of skb-mark for hinting
- Flow handling should work with and without hw offload
Possible Flow-Miss Handling

User-Space

- ovs-vswitchd

Kernel

- OVS Datapath

iNIC

1. Packet

- Map
- Prog
- eBPF
- Classifier
Possible Flow-Miss Handling

User-Space

- ovs-vswitchd

Kernel

- OVS Datapath

iNIC

1. packet

- Map
- Prog
- eBPF Classifier

2. packet (without skb-mark)
Possible Flow-Miss Handling

1. packet

2. packet (without skb-mark)

3. miss: packet, key

User-Space

ovs-vswitchd

Kernel

OVS Datapath

iNIC

Map Prog

eBPF Classifier

© 2015 Netronome
Possible Flow-Miss Handling

1. Packet

2. Packet (without skb-mark)

3. Miss: Packet, Key

4. Store: Key/Mask → skb-mark

User-Space

- ovs-vswitchd

Kernel

- OVS Datapath

iNIC

- eBPF Classifier

- Map

- Prog
Possible Flow-Miss Handling

1. packet
2. packet (without skb-mark)
3. miss: packet, key
4. store: key/mask→skb-mark
5. add flow: key'/mask'→actions
exec: packet→actions

User-Space

Kernel

iNIC

Map Prog

eBPF Classifier

ovs-vswitchd

OVS Datapath
Possible Flow-Miss Handling

1. packet

2. packet (without skb-mark)

3. miss: packet, key

4. store: key/mask→skb-mark

5. add flow: key'/mask'→actions

exec: packet→actions

6. packet

User-Space

Kernel

iNIC

ovs-vswitchd

OVS Datapath

Map

Prog

eBPF

Classifier

Packet (without skb-mark)
Possible Flow-Hit Handling

1. Packet
   - eBPF Classifier
   - Map
   - Prog

2. Packet, skb-mark
   - OVS Datapath

3. Packet
   - ovs-vswitchd

User-Space

Kernel

iNIC