OvS Hardware Offload with TC Flower

Simon Horman
Open vSwitch 2017 Fall Conference
San Jose
Overview

- OvS Kernel Datapath Offload Models
- Overview of TC Flower
- TC Flower Based Offload
Motivation for Hardware Offload

- Provide greater throughput
- Increase CPU core efficiency and scalability
Kernel Datapath

User-Space

ovs-vswitchd

Kernel

misses, flows, stats

OVS Datapath

packet

packet
OVS Datapath Hooks

- User-Space
  - ovs-vswitchd
  - flows, stats, misses

- Kernel
  - OVS Datapath
  - flows, stats, misses

- SmartNIC
  - SmartNIC Datapath
  - packet
  - packet
OvS-TC

User-Space

flows, stats

flows, stats, misses

Kernel

TC Datapath

misses

flows, stats, misses

OVS Datapath

SmartNIC

packet

SmartNIC Datapath

packet
Overview of TC Flower
Overview of TC Flower

- Packet classifier for Linux kernel traffic classification (TC) subsystem
- TC Flower classifier allows matching packets against pre-defined flow key fields:
  - Packet headers: f.e. IPv6 source address
  - Tunnel metadata: f.e. Tunnel Key ID
  - Metadata: Input port
- TC actions allow packet to be modified, forwarded, dropped, etc…
  - pedit: modify packet data
  - mirred: output packet
  - vlan: push, pop or modify VLAN
  - ...
Example of TC Flower

- Filter packets received on eth0
- Drop TCP packets with destination port 80

```bash
# tc qdisc add dev eth0 ingress
# tc filter add dev eth0 protocol ip parent ffff:
   flower ip_proto tcp dst_port 80
   action drop
```
Hardware Offload Policy

- per-netdev configuration
  - Allow disabling/enabling adding flows to hardware
    
    # ethtool -K eth0 hw-tc-offload on
    # ethtool -K eth0 hw-tc-offload off

- skip_hw and skip_sw flags
  - Allow users to influence placement of flows by kernel
  - Default is to add to hardware and try to add to software

- in_hw and not_in_hw flags
  - Allow kernel to report presence of flow in hardware
Example of Setting Hardware Policy

- Add flow only to hardware

  # tc qdisc add dev eth0 ingress
  # tc filter add dev eth0 protocol ip parent ffff: \
  flower skip_sw ip_proto sctp dst_port 80 \
  action drop
Example of Viewing Rule in Hardware

- Policy was to only add rule to hardware (skip_sw)
- Rule is present in hardware (in_hw)

```bash
# tc filter show dev eth0 ingress
filter parent ffff: protocol ip
    pref 49152 flower chain 0
    handle 0x1
    eth_type ipv4
    ip_proto sctp
dst_port 80
    skip_sw
    in_hw
...
TC Flower Based Offload
Tables and Flows

● **OvS Datapath**
  o Single table
  o Match on in_port
  o Flows have a wide key and are disjoint
  o And therefore can be partitioned into slices
  o Megaflows are priority independent

● **TC Flower**
  o Multi-table (chain) support
  o Attached to in_port
  o Flows have a wide key
  o Only one mask per priority
Offload Integration in OvS

- New netdev ops called by DPIF layer
- Try to offload each flow
  - f.e. By adding to TC Flower
- If unsuccessful then add to software datapath
  - f.e. kernel datapath
Configuration

- Disabled by default
- Enabled/disabled globally

```
# ovs-vsctl set Open_vSwitch . other_config:hw-offload=true
```

- TC Policy controls placement of flows
  - none (default): Try to add to TC software datapath and hardware if present
  - skip_sw: Try to add to TC software datapath
  - skip_hw: Try to add to hardware

- Also set globally

```
# ovs-vsctl set Open_vSwitch . other_config:tc-policy=none
```
Viewing Flows

- Dump all datapath flows (default)
  ```
  # ovs-dpctl dump-flows
  ```
- Dump only flows that in kernel datapath
  ```
  # ovs-dpctl dump-flows type=ovs
  ```
- Dump only flows that are offloaded
  ```
  # ovs-dpctl dump-flows type=offloaded
  ```
Current Features

- **Matches**
  - L2 ~ L4 and Tunnel metadata matches
  - L2: type, addresses, VLANs
  - MPLS: LSE fields
  - L3: Addresses, protocol, TTL, ...
  - L4: UDP/TCP/SCTP ports
  - Tunnel Metadata: Tunnel ID

- **Actions:**
  - Drop, output, VLAN push/pop
Status

- **Offload Integration in OvS**
  - Included in OvS v2.8
- **TC Flower**
  - Initially added in Linux kernel v4.2
- **NFP Driver**
  - Basic offload support present since Linux kernel v4.13
Future Work
Stateless Match/Action Enhancements

- **Set Action**
  - Patches available
- IPv6 label and neighbour discovery
- Maskable match of MPLS LSE fields
- GENEVE options
Conntrack

- Aim to allow enhanced rules to be written
  - By taking into account Conntrack state
- Proposal is to follow implemented by Open vSwitch kernel datapath:
  - Conntrack action passes packet to conntrack subsystem
  - Packet is then classified for a second time; conntrack state may form part of flow key