OVS Performance on Steroids - Hardware Acceleration Methodologies

OVS Conference Nov 2017  Rony Efraim
Agenda

- OVS Offload - Accelerated Switch And Packet Processing (ASAP²)
- Full OVS Offload (ASAP² Direct) - SRIOV
  - Software based VS Hardware based
  - OVS support for HW offload
  - OVS HW Offload – ConnectX-5 performance
  - Future work for HW offload
- Partial OVS Offload (ASAP² Flex) - DPDK
  - RFC OVS-DPDK using HW classification offload
  - Vxlan in OVS DPDK
  - Multi-table
  - vxlan HW offload concept
- Mellanox OVS Offload Community Work
ASAP\(^2\) takes advantage of ConnectX-4/5 capability to accelerate "in host" network stack
- Two main use cases:

**ASAP\(^2\) Direct**
- Full vSwitch offload (SR-IOV)

**ASAP\(^2\) Flex**
- vSwitch acceleration
ASAP$^2$-Direct

Full Virtual Switch Offload (SRIOV)
Software based Vs. Hardware based

**Traditional Model: All Software**
High Latency, Low Bandwidth, CPU Intensive

- OVS-vswitchd
- User Space
- OVS Kernel Module

**ConnectX-4: Hardware Offload**
Low Latency, High Bandwidth, Efficient CPU

- OVS-vswitchd
- User Space
- OVS Kernel Module
- ConnectX-4 eSwitch

- First flow packet
- Fallback FRWD path
- HW forwarded Packets
OVS support for HW offload

- Changes are made only in the OVS user space code.
- HW offload of flow using TC flower.
- Packets forwarded by the kernel datapath are transmitted on the representors and forwarded by the e-switch to the respective VF or to the wire.
### OVS HW Offload – ConnectX-5 performance

<table>
<thead>
<tr>
<th>Test</th>
<th>ASAP2 Direct</th>
<th>OVS DPDK</th>
<th>Benefit</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Flow VXLAN</td>
<td>66M PPS</td>
<td>7.6M PPS (VLAN)</td>
<td>8.6X</td>
</tr>
<tr>
<td>60K flows VXLAN</td>
<td>19.8M PPS</td>
<td>1.9M PPS</td>
<td>10.4X</td>
</tr>
</tbody>
</table>

- **ConnectX-5 provide significant performance boost**
  - Without adding CPU resources
Future work for HW offload

- Table offload to support recirculate
- Connection tracking
- LAG (bonding) for SRIOV
- VF live migration
ASAP² Flex
HW accelerate OVS-DPDK
For every datapath rule we add a rte_flow with flow id
The flow id cache contains mega flow rules
When packet is received with flow id, no need to classify the packet to get the rule
### RFC Performance

- Code submitted by Yuanhan Liu.
- Single core for each pmd, single queue.

<table>
<thead>
<tr>
<th>Case</th>
<th>#flows</th>
<th>Base MPPs</th>
<th>Offload MPPs</th>
<th>improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wire to virtio</td>
<td>1</td>
<td>5.8</td>
<td>8.7</td>
<td>50%</td>
</tr>
<tr>
<td>Wire to wire</td>
<td>1</td>
<td>6.9</td>
<td>11.7</td>
<td>70%</td>
</tr>
<tr>
<td>Wire to wire</td>
<td>512</td>
<td>4.2</td>
<td>11.2</td>
<td>267%</td>
</tr>
</tbody>
</table>
- There are 2 levels of switches that are cascaded.
- The HW classification accelerates only the lower switch (br-phys1).
- br-phys1 is a kernel interface for vxlan.
- The OVS datapath is required to classify the inner packet.
The action of a rule can be to go to other table.

It can be used to daisy chain classification rules

<table>
<thead>
<tr>
<th>Table 0</th>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Match A → flow ID 1</td>
<td>Match E → flow ID 2</td>
</tr>
<tr>
<td>Match B → drop</td>
<td>Match F → flow ID 3</td>
</tr>
<tr>
<td>Match C → Table 1</td>
<td></td>
</tr>
<tr>
<td>Match D → Table 1</td>
<td></td>
</tr>
<tr>
<td>Default no flow ID</td>
<td>Default flow ID 4</td>
</tr>
</tbody>
</table>
VxLAN HW offload concept

- If the action is to forward to internal interface add HW rule to point to a table named the internal interface.
- If the in port of the rule is internal port (like vxlan), add rule to the table named of the interface with a flow id.
- When a packet is received with a flow id, use the rule even if the in port is internal port.
- A packet that tagged with flow id is a packet that came on a physical port and is classified according to the outer and the inner headers.

```
Table 0
Match A → flow ID 1
Match B → drop + count
Match C → Table 1 + count
Match D → Table 1 + count
Default no flow ID

Table 1
all the rules that the src port is the vxlan interface
Match E → flow ID 2
Match F → flow ID 3
Default flow ID 4
```

Diagram:
- VM
- Tap
- Uplink/PF
- Br-phy1
- Br-int
- VM
- Tap
- vxlan
- Br-phy1
- IP address
- VM
- Tap
- Uplink/PF
- Br-phy1
- IP address
VxLAN HW offload

- If in port is HW port, add rule to the HW action can be flow id or to table according to the port to forward to
- If the in port is internal port (like vxlan), add a rule to all the HW port with action flow id (because traffic can came form any external/HW port)
- The flow id need to be unique

Table 0

- Match A → flow ID 1
- Match B → drop + count
- Match C → Table 1 + count
- Match D → Table 1 + count
- Default no flow ID

Table 1

- all the rules that the src port is the vxlan interface
- Match E → flow ID 2
- Match F → flow ID 3
- Default flow ID 4

Table 1

- all the rules that the src port is the vxlan interface
- Match E → flow ID 2
- Match F → flow ID 3
- Default flow ID 4
Mellanox OVS Offload Community Work

- **Linux Kernel**:
  Using SR-IOV offloads with Open-vSwitch – netdev conf 1.2
  [link](https://netdevconf.org/1.2/papers/efraim-gerlitz-sriov-ovs-final.pdf)

- **Open vSwitch**:
  [PATCH V11 00/33] Introducing HW offload support for openvswitch
  [link](https://mail.openvswitch.org/pipermail/ovs-dev/2017-June/333957.htm)

- **Open Stack**
  Os-vif: [link](https://review.openstack.org/#/c/398277/)
  Nova: [link](https://review.openstack.org/#/c/398265)
  Neutron: [link](https://review.openstack.org/#/c/275616)

- **DPDK**
  RTE_Flow API
  OVS-DPDK RFC:
  [link](https://www.mail-archive.com/ovs-dev@openvswitch.org/msg12562.html)
ASAP² Mellanox team

- Roi Dayan
- Paul Blakey
- Hadar Hen Zion
- Mark Bloch
- Or Gerlitz
- Natali Shechtman
- Natan Oppenheimer
- Oded Shanoon
- Olga Shern
- Yuanhan Liu
- Moshe Levi

- Rabie Loulou
- Rony Efraim
- Shahar Klein
- Shani Michaeli
- Tal Anker
- Haggai Eran
- Ilya Lesokhin
- Lior Narkis
- Vlad Buslov
- Chris Mi
Thank You