Red Hat's perspective on OVS HW Offload Status

Current state and what is WIP

Rashid Khan
Senior Manager, Networking Services
Nov. 17, 2017
Acknowledgements and Disclaimers

I am presenting the work of many many people... Special thanks to: Andrew T, Franck B, Eelco C, Marcelo L, Paolo A, Flavio L, Kevin T

Performance numbers shown in this presentation are based on test results from running a specific series of tests in our labs.

Test results vary from one setup to another and based on different use cases. Any test results mentioned are for example-only scenarios and are not conclusive nor a recommendation of one vendor’s solution over another.
AGENDA

Why offload?

Does it look promising?

What is left to do?

Backup / more info

Please view Franck’s presentation from Thursday 11:30am:

OVS-DPDK for NFV: go live feedback!

Please view Aaron Conole’s presentation from Thursday 3:30pm:

Conntrack + OvS
Why not just SW?
Simply way too many cores needed

- 4 Mpps/core with expert level tuning
  - Yes even with DPDK!

- Does not scale to 25G, 40G, 100G ....
OVS-DPDK: virtio, vhost-user, virtio PMD

```
while (1) {
    RX-packet()
    forward-packet()
}
```
Zero Packet loss

VM L2 forwarding, VLAN networks, \textit{intra}-NUMA node, single queue

- 2 × virtio-net interfaces (NUMA node0)
- 2 × 10Gb interfaces (NUMA node0)
- Testpmd DPDK application in VM with 2 × virtio-net
- OVS using 4 PMD threads (2 cores) to process data-plane traffic
- Directly connected packet generator and compute node, no HW switch.
- Bidirectional traffic, 128 flows, \textit{no broadcast or multicast packets}
- Measurement time, zero-loss: 2 hours, non-zero-loss: 5 mins
- Maximum rate while within specified loss:

<table>
<thead>
<tr>
<th>Frame size</th>
<th>Loss: 20 packets-per-million</th>
<th>Loss: 5 packets-per-million</th>
<th>Loss: 1 packet-per-million</th>
<th>Loss: 0 packets-per-million</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Loss: 5 packets-per-million</td>
<td>Loss: 1 packet-per-million</td>
<td>Loss: 0 packets-per-million</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Loss: 1 packet-per-million</td>
<td>Loss: 0 packets-per-million</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mpps</td>
<td>Gbps</td>
<td>Mpps/core</td>
<td>Gbps/core</td>
</tr>
<tr>
<td>64</td>
<td>9.38</td>
<td>6.30</td>
<td>4.69</td>
<td>3.15</td>
</tr>
<tr>
<td>256</td>
<td>7.66</td>
<td>16.19</td>
<td>3.83</td>
<td>8.45</td>
</tr>
<tr>
<td>1500</td>
<td>1.64</td>
<td>19.99</td>
<td>0.82</td>
<td>9.99</td>
</tr>
</tbody>
</table>
So what's the big deal?

Just add more CPUs, add more cores

If forwarding 10G of traffic takes ~4 cores
If Storage takes ~2 cores
That is already ¼ of a 24 core chip

This is “wasted” revenue for the cloud providers
They charge per cycle per second
Swing of the pendulum

All HW
100 - 10 years ago

All SW
9 - 0 years ago

Very near future (some HW, some SW)
Many HW vendors have OVS Offload solutions

NETRONOME  MELLANOX  CAVIUM

CHELSIO  BROADCOM

Others
## Offloading method

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Offloading Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Netronome</td>
<td>TC (kernel)</td>
</tr>
<tr>
<td>Mellanox</td>
<td>TC (kernel)</td>
</tr>
<tr>
<td>Broadcom</td>
<td>TC (kernel)</td>
</tr>
<tr>
<td>Chelsio</td>
<td>TC (kernel)</td>
</tr>
<tr>
<td>Cavium</td>
<td>OVS runs in the NIC firmware, offloading is transparent from CPU PoV</td>
</tr>
</tbody>
</table>

Accepted in upstream netdev
Example

PVP test

https://github.com/chaudron/ovs_perf
NETRONOME

Physical to Virtual to Physical test, L3 flows (10000 flow)

- Netronome[40G]
- OVS-DPDK, XL710[40G]
- Max PPS 10G
- Max PPS 40G

Open vSwitch CPU utilization

- Netronome[40G]
- OVS-DPDK, XL710[40G]

Total System CPU utilization

- idle
- nice
- guest
- steal
- soft
-irq
- iowait
- sys
- nice
- user
Mellanox

Physical to Virtual to Physical test, L3 flows (10000 flow)

Open vSwitch CPU utilization

Total System CPU utilization
NETRONOME

Purple is sw only
Mellanox
What is WIP / To-do List

Rome wasn't built in a day...

- Connection tracking offload
- Openstack integration
- Flow insertion / deletion rate improvement
- Expand to do additional actions
- Metrics / statistics / billing
- System level logging (supportability)
- Support for sending to multiple ports
- QOS
- Kubernetes integration
- Migration from one card to another
THANK YOU

For further questions / comments:
rkhan@redhat.com

We Are Hiring !!!!!
More information
SW used for testing

Netronome:
Linux upstream kernel, v4.13 for PVP test results. Linux V4.14rc4 for TC insertion rates. OVS master branch from October 26th (7b997d4). DPDK/testpmd on VM v16.07

Mellanox:
Linux upstream kernel, net-next commit e1ea2f9856b7. OVS master branch commit b05af21631ce, DPDK 17.11-rc2 (all git tips from Oct 30th).

OVS-DPDK:
RHEL7.4 latest kernel, OVS master branch from September 26th (97ee6d4), DPDK v17.05.2, DPDK/testpmd on VM v16.07