vSwitch Acceleration with Hardware Offloading

CHEN ZHIHUI
JUNE 2018
Current Network Solution for Virtualization

Software Solution

SRIOV Solution
VF Representor for Virtualization

- **VF Representor**
  - Net Device modeling of eswitch port and exposed through PF driver.
  - VF and its representor works like Linux veth pair
  - Flow configuration (add/remove)
  - Works under switchdev mode

- **Access from both kernel and DPDK**
  - Multi Queue (RSS/TSO/CSUM)
  - Attach/Detach in DPDK
  - Multiple DPDK instances over VF representor

- **With VF representor, vSwitch can work with SRIOV together and reduce CPU% consumed by virtio.**

- **Disadvantages:**
  - 3x PCIe access for traffic from VM to wire and vice versa, PCIe can become a bottleneck for throughput.
  - Need vendor specific driver in VM.

---

![Diagram](image1)

- Host
  - VM1
    - VF1
  - VM2
    - VF2

- SW Datapath
  - ESwitch
  - NIC
  - Port

- PF, R1, R2

---

VF Representor for Virtualization
Flow Table with Mellanox Adapter

- **Key match fields**
  - Ethernet
  - IP(v4 /v6)
  - TCP/UDP
  - Inner packet for Overlay
  - VNI

- **Flexible fields extraction by “Flexparse”**

<table>
<thead>
<tr>
<th>Classification</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRC MAC = Dest MAC = SRC IP = Dest IP = Protocol =</td>
<td>Counter</td>
</tr>
<tr>
<td>SRC MAC = Dest MAC = SRC IP = Dest IP = Protocol =</td>
<td>Another rule</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VLAN tag = Tunneling type Inner packet SRC IP = Dest IP = Protocol =</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>VLAN tag = Tunneling type Inner packet SRC IP = Dest IP = Protocol =</td>
<td>Header rewrite</td>
</tr>
<tr>
<td>VLAN tag = Tunneling type Inner packet SRC IP = Dest IP = Protocol =</td>
<td>Meta Data</td>
</tr>
<tr>
<td>VLAN tag = Tunneling type Inner packet SRC IP = Dest IP = Protocol =</td>
<td>Flow-ID Tag</td>
</tr>
</tbody>
</table>

- **Action**
  - Forwarding
  - Drop
  - Counter
  - Encap/Decap
  - Flow ID
  - Header rewrite
  - …..
vSwitch Design with Hardware Offloading

- HW datapath on eswitch through configuring flow table.
  - TC Flower
  - DPDK RTE_FLOW
- Software datapath handle ‘the first packet’ and unsupported flows through VF representor.
- Support both SRIOV and VirtIO
  - Direct path to VM for SRIOV
  - Optimized vhost backend for virtio acceleration
    - TX: Forward packet to HW with meta data.
    - RX: Receive packet from VF with Flow ID which can identify destination VM.
- Rules management
  - Add/delete/Query
  - Aging
  - Batch operations
vSwitch with Hardware Offloading for SmartNIC

- Control plane and SW Datapath on ARM
- HW Datapath on NIC
- Both SRIOV and VIRTIO interface to VM

Advantages
- Support bare-metal cloud
- Separation of computing domain and networking domain, all host resources (core and memory) can be used for VMs.
- Efficient

Disadvantages
- Higher cost and power
- Two management domain.
Hardware Offloading Performance with SRIOV

Lossless VXLAN Performance with HW Offloading

System Configuration:
1. E5-2667 V3 @ 3.20GHz
2. Mellanox 100G ConnectX-5 NIC
3. RHEL7.5 Host and Guest
4. VXLAN Encap/Decap on NIC
5. T-Rex and Testpmd run in VMs
Hardware Offloading Performance with VirtIO

<table>
<thead>
<tr>
<th>Test Case\Cores</th>
<th>1</th>
<th>2</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM-&gt;HV-&gt;wire</td>
<td>9.98</td>
<td>18.3</td>
<td>36.4</td>
</tr>
<tr>
<td>VM-&gt;HW-&gt;VxLAN Encap-wire</td>
<td>9.95</td>
<td>18.3</td>
<td>36.2</td>
</tr>
<tr>
<td>Wire-&gt;HV-&gt;VM</td>
<td>13.4</td>
<td>23.3</td>
<td>46.3</td>
</tr>
<tr>
<td>Wire-&gt;VxLAN Decap-&gt;HW&gt;VM</td>
<td>13.5</td>
<td>25.0</td>
<td>45.8</td>
</tr>
</tbody>
</table>

System Configuration:
(1) E5-2650 V4 @ 2.20GHz
(2) Mellanox 100G ConnectX-5 NIC
(3) Performance Metric: Mpps at 64byte

- Overhead caused by HW Encap/Decap on the path from/to VM can be minimized.
- If vhost-backend is strong enough, HW Offloading can bring high performance to virtio also.
RoCEv2 Support

- Only support SRIOV interface
- All RoCEv2 must go through HW datapath.
- ARP should be handled by SW datapath so that two endpoints of RoCEv2 can exchange address information.
- All RoCEv2 should be sent to wire or local VFs directly through configured rules like following
  - match \{ip_proto=udp, dport=RoCE, dmac=<mac of VF1>\} action \{fwd to VF1\}
  - match \{ip_proto=udp, dport=RoCE, dmac=<mac of VF2>\} action \{fwd to VF 2\}
  - match \{ip_proto=udp, dport=RoCE\} action \{fwd to wire\}
- New HW support VxLAN Encap/Decap for RoCEv2
  - Encap header can be based on inner (src ip, dst ip) + VNI.
  - ECN information need be copied from outer header to inner header after decap on RX.
Other key Consideration

- **VF LAG**: VM sees only one VF while it can use two physical ports for Load balancing and link redundancy.
- **VF Mirroring**: mirroring the traffic from/to one VF to a dedicate admin VF for monitoring and traffic analysis.
- **Connection Track**: sending TCP packets with specific flags to software for processing connection state.
- **Live Upgrade**: update to new instance, need migrate both SW & HW datapath and interfaces from current instance to new instance.
- **SRIOV Live Migration**: VM with SRIOV VF can be migrated from one machine to another machine.
Thank You!