Enabling OVS Hardware Offload using LiquidIO

Open vSwitch
November 16-17, 2017 | San Jose, CA

Manoj Panicker – Cavium, Inc
Open vswitch software models

- Two different models –
  - Data plane in kernel; Control in user space; Packets switched within kernel to VM.
  - Data and control plane in user space; Packets bypass kernel completely; Packets switched entirely in user space.

- Both models supported by open source community
- Hardware independent
Why move vSwitch out of the host?

Limitations with a pure software based vSwitch
- Requires significant host CPU cycles to get packets to the VM.
- Reduces host CPU cores available to run VMs.
- Challenge in keeping up with increasing bandwidth requirements.

Customers need a resolution to some or all of the above issues in their current deployment.
- Cavium’s customers also had other reasons to offload vSwitch to a NIC adapter
  - Manage vSwitch independent of host OS. Host OS could be under tenant control.
  - Upgrade or manage OVS or customization to vSwitch without modifying host OS.
Models to offload Open vSwitch to NIC adapters

• Data plane offload model
  ▪ Uses PCI pass-through of VF to VM to bypass host CPU.
  ▪ OVS Control plane continues to be on host OS.
  ▪ The vswitch bridge exists in the host
  ▪ Uses representors of the VFs attached to VM as ports in the bridge.
  ▪ Has enablers in the Linux kernel including support for
    o switchdev
    o TC/Flower based flow offload to network devices

• Full Open vSwitch offload (control and data plane)
  ▪ Leverages PCIe SRIOV and PCI pass-through of VF to VM to bypass host CPU.
  ▪ The OVS control and data plane operate from within the NIC adapter
  ▪ Does not require VF representors or switchdev for normal OVS operation.
  ▪ This is the LiquidIO model
LiquidIO OVS offload

LiquidIO accelerates OVS Data path and restores CPU cores back to the server

- Controller
- OVS-DB Server
- OVSDB
- vSwitchd
- OVS Data path
- X86 Userspace
- x86 Host
- X86 Kernel
- PCI Express Slot
- eth0, eth1
- PF, PF
- Accelerated Data Path
- LiquidIO

Controller

OVS-DB Server

OVSDB

vSwitchd

OVS Data path

X86 Userspace

x86 Host

LiquidIO

Controller

OVS-DB Server

OVSDB

vSwitchd

OVS Data path

X86 Userspace

x86 Host

LiquidIO

Controller

OVS-DB Server

OVSDB

vSwitchd

OVS Data path

X86 Userspace

x86 Host

LiquidIO

Controller

OVS-DB Server

OVSDB

vSwitchd

OVS Data path

X86 Userspace

x86 Host

LiquidIO

Controller

OVS-DB Server

OVSDB

vSwitchd

OVS Data path

X86 Userspace

x86 Host

LiquidIO
The LiquidIO model

Inter-core messaging protocol allows control to data plane communication.
LiquidIO Open vSwitch offload

- Data plane leverages acceleration for multi-core packet processing, packet parsing and scheduling provided by hardware.
- Uses hardware support for implementing message passing layer.
- Zero host CPU utilization for OVS processing.
- Uses standard LiquidIO VF drivers in VM. No VF representor required.
- Supports conntrack + NAT.
- Also supports encryption of network traffic.
LiquidIO OVS offload – life of a packet
Leverages conntrack module from the Linux kernel running on LiquidIO cores.

Message passing layer to keep copy of CT table in the data plane.

All packets continue to go to conntrack module in Linux kernel and decision made in kernel on flow update.

When connection gets to established state, the flow and CT table are updated in the data plane.

Once offloaded, CT state lookup happens in data plane.

Data plane supports timeout of CT entries.
## LiquidIO OVS customization – customer use cases

LiquidIO support all standard OVS vswitch operations but customers need to tweak firmware to fit in their deployment.

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Secure Access model</strong></td>
<td>▪ Deny host access to OVS running on LiquidIO.</td>
</tr>
<tr>
<td></td>
<td>▪ OVS is remotely managed using proprietary management plane.</td>
</tr>
<tr>
<td><strong>Custom tunneling protocols</strong></td>
<td>▪ Uses custom tunneling protocols instead of Vxlan, GRE.</td>
</tr>
<tr>
<td></td>
<td>▪ Uses custom management plane. Control of OVS via network or from the local host.</td>
</tr>
<tr>
<td><strong>IPSec support for VM payload</strong></td>
<td>▪ IKE runs in VM; xfrm offload module allows SA offload to LiquidIO.</td>
</tr>
<tr>
<td></td>
<td>▪ LiquidIO can perform IPsec operation for VM traffic and encapsulate in tunnel as per OVS action.</td>
</tr>
<tr>
<td><strong>Custom core vswitch</strong></td>
<td>▪ Uses custom data plane with modified flow tables.</td>
</tr>
<tr>
<td></td>
<td>▪ Openflow is still used to apply rules.</td>
</tr>
<tr>
<td></td>
<td>▪ Control plane applications are also customized.</td>
</tr>
</tbody>
</table>
**Netperf TCP bi-directional (8 flows) with Vxlan**

LiquidIO VF in PCI passthrough, Intel with OVS+DPDK using virtio.
LiquidIO OVS Offload – UDP performance

**UDP Performance**

<table>
<thead>
<tr>
<th>THROUGHPUT (GBPS)</th>
<th>PACKET SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>64</td>
</tr>
<tr>
<td>20</td>
<td>128</td>
</tr>
<tr>
<td>15</td>
<td>360</td>
</tr>
<tr>
<td>10</td>
<td>512</td>
</tr>
<tr>
<td>5</td>
<td>1024</td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

LiquidIO OVS Bridge
Intel OVS+DPDK Bridge
LiquidIO OVS Vxlan
Intel OVS + DPDK Vxlan

**DPDK testpmd running Rx mode in VM (100 flows)**

LiquidIO VF in PCI passthrough, Intel with OVS+DPDK using virtio.

**Setup**

- **Dell T630 with 14 cores E5-2690v4 @ 2.6 Ghz**
- **Host OS: CentOS 7.3**
- **Guest OS: CentOS 7.3**
- **Single VM with 4 vCPU**
- **Adapters:**
  - Intel X710 10G – DA4
  - LiquidIO 2360-210
- **Intel tests with OVS+DPDK on 4 host CPUs.**
- **Tested with UDP traffic on 2 x 10G ports.**
- Uses VF representors in host to allow Neutron server to bind LiquidIO VF to compute node VMs.
- Uses relay agent to allow Neutron server to reach OVS control plane running on LiquidIO.
- Integration bridge is implemented in the LiquidIO adapter itself.
- Slow path continues to be within the LiquidIO but VF stats are updated for the VF representor in the hypervisor.
- No changes to Neutron for this model.
- Proof of concept completed for ODL & OVN with Pike based RDO.
LiquidIO OVS offload – pros & cons

• Advantages
  ▪ Slow path avoids PCI overhead.
  ▪ Host isolation – vswitch can be remotely managed.
  ▪ Migration to new OVS version without kernel or host OS changes.
  ▪ Supports all tunneling protocol available with OVS – Vxlan, NVGRE, GENEVE.
  ▪ Support for conntrack + NAT.

• Limitations
  ▪ VMs connect to the data network using a single adapter.
  ▪ Openstack components lacks flexibility to support full offload.
  ▪ Data plane requires tweaks to support new OVS features.
  ▪ Live migration requires attach to VM using virtio.
    ▪ PCI passthrough to VM cannot support Live Migration.
LiquidIO OVS Offload – looking ahead

- Add support for OVS 2.8
- Infrastructure to support TC/Flower based offload
- Support next generation ARM based LiquidIO adapters
- Improve conntrack implementation
- Add support for ebpf offload
- How can the community help with full offload?
  - Allow control and data plane to exist in different domains. Some support for TCP based connections but this could be fully extended.
  - OVN currently expects vswitch to run on the local processor.
Thank You!