An Comparison of Fd.io and OVS/DPDK

Thomas F Herbert
SDN Group
Red Hat
Contents

● Introduction to Fd.io/VPP
  ○ Fd.io Project and Community
  ○ VPP Architecture and Performance

● Review of OVS/DPDK
  ○ OVS/DPDK Architecture and Performance

● Fd.io compared to OVS/DPDK

● Looking Forward

● Conclusion
Fd.io Introduction

- **Community**
  - Open Source Project
  - Linux Foundation
  - Open Governance Model
  - Active and Growing Community

- **Licensing**
  - Apache 2.0

- **History**
  - Internal of Cisco product: vRouter
    - v9000 ACL products
  - Code dump: Cisco January 2016

- **Projects**
  - VPP - Core Engine
  - CSIT
  - NSH/SFC
  - ONE -- Overlays, LISP
  - Honeycomb
  - VPP Sandbox -- Bootstrapping hosting
  - More to Come
Fd.io

Full Layer 2 and Layer 3 Functionality

IP
- Complete IPv4 and IPv6 Stacks
- GRE, vxLAN, IPSEC, DHCP
- Neighbor Discovery, Router Advertisement
- Segment Routing
- MAP/LW46
- ARP Termination and Proxy ARP
- ...

MPLS
- MPLS over Ethernet with Deep label stacks

Layer 2
- Vlans -- single and double
- Mac Learning
- Flooding
- Input ACLs
- ...

Counters for Everything
- Input Checks
- TTL, Header Checksum
- ARP
VPP Architecture -- How VPP Works

Packet Vector is Input to DPDK

Nodes on Graph Pipeline Process Packets According to Registered Dependencies

Additional Plugins can be Introduced to Create New Nodes

Examples:

VxLAN - NSH VTEP
Fd.io (VPP) Architecture

VPP reads the largest available vector of packets from the network IO layer.

VPP then processes the vector of packets through a Packet Processing graph.
Fd.io (VPP) Architecture
Fd.io (VPP) Architecture
Open vSwitch Architecture
Open vSwitch Architecture
Open vSwitch Acceleration

Performance Optimizations

DPDK User Space Summit 2015 Dublin
Open vSwitch Acceleration
Challenge of Comparing Performance

- Perception that VPP “Fixes” OVS/DPDK Scalability Problem.
- Need Real World Use Cases
  - Open Stack Deployments of Both to Compare
  - The hope is that OPNFV/Vsperf will compare
  - Fds vs ovs/dpdk
- As TOR VPP May Scale Better
  - But Deployed in Compute Node or Hosting NFVs?
Raw DPDK Performance

DPDK Host L3 forwarding on Intel® Xeon® processor E5-2695 v4

Throughput (packets/second)

Packet size (bytes)

Intel® Open Network Platform Release 2.1 Performance Test Report
Open vSwitch Performance w/DPDK

DPDK Can Scale OVS performance significantly particularly when multiple cores are used.
Open vSwitch Performance

DPDK Can Accelerate OVS Well with Small Packet Sizes.
Open vSwitch Performance

When passed through VM with vhost-user, DPDK/OVS may scale OK with simple forwarding with relatively small number of flows.
OVS and FD.IO Similarities

- DPDK based
- Implemented in Software on Commodity Processors
  - Intel, ARM, Power 8
- Deployable in Compute node of Open Stack
- Host to VM uses vhost-user
- Container Support is through kernel via TAP
  - Limitations of Kernel Networking
  - Will User Space Netlink Be Faster
    - Security Issues
Architecture: Apples vs Oranges

**Open vSwitch**

- Open Flow Protocol matches and actions
  - Fast Path -- Linux Kernel Module
  - Fast Path -- Accelerated by DPDK
  - Fast Path Exact Match Cache -- Fastest
  - About 8K matches using linear search
  - Misses go to Slower MiniFlow Match -- Slower
  - Misses then go to OFP Match Slowest

**FD.IO/VPP**

- VPP fd.io
- Graph based vector Processing Engine
- Consumes vector of input packets
- Extensible via plugins
  - Synchronized for parallel operation
- Multiple nodes
  - Optimized for parallelism
  - Hardware like -- Use branch prediction.

**DPDK**
## Open vSwitch Control Plane

**Two control plane protocols**

- **Open Flow Protocol**
  - Matches and Actions
  - 44 Packet Match Fields Plus Metadata
  - CLI
    - ovs-ofctl

- **OVSDB Protocol**
  - Ports, Bridges, Tunnels, Config
  - Highly configurable
  - CLI
    - ovs-vsctl

## VPP Control Plane

- **Basic VPP has**
  - CLI API
  - Restconf IF

- **Extensible via plugins**
  - Each Plugin has has
    - CLI API
    - Restconf IF

- **Includes complete IPv4 and IPv6 stack**
  - Other vRouter Features
    - CLI API
    - Restconf IF
Control Plane Differences - Open Stack

- Neutron ML2
- Netconf
- ODL NetConf Plugin
- ODL OpenFlow - OVSDB-Netvirt
- Open vSwitch Vswitchd ovsdb-server
- VPP Plugin API
- VPP
- HoneyComb RestConf API

Data Plane

Control Plane

VPP
<table>
<thead>
<tr>
<th>Open vSwitch</th>
<th>FD.IO/VPP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specified by Open Flow as “Switch” 1.0 - 1.5 No one predicted NFV Cloud Use Case</td>
<td>Designed as vRouter Configured by IOS/CLI</td>
</tr>
<tr>
<td>4 to 5 years of experience</td>
<td>Super Configurable Statistics Everywhere</td>
</tr>
<tr>
<td>Works Well with Conceptualized Networks</td>
<td>Historical Use in Real World TOR Switch -- Infrastructure Router</td>
</tr>
<tr>
<td>1+ Years with NFV Experience</td>
<td>Now Utilized in Complex Multi-Use? Compute Node Host HyperVisor</td>
</tr>
<tr>
<td>Observed Performance Scaling May not be as Good: Deteriorates with many Flows</td>
<td>Not originally designed as “SDN” as defined by Open Flow</td>
</tr>
</tbody>
</table>

**Performance:** Apples vs Oranges
VPP vs OVS Performance?

- OVS may not scale well when the number of flows exceed 8192 due to size of EMC.
- According to NEANTC testing, VPP scales close to linearly.
- Also, the VPP architecture may hold promise for accelerating vxLAN/NSH in the VTEP case for SFC.
- But...

Is testing on Number of MAC Addresses Sufficiently Realistic?

FD.io VPP OVS/DPDK Outlook

- VPP shows promise
  - Pipelined Approach
  - Well Suited for DPDK
- Dual Approach Possible
  - OVS in Control Plane
- FD.io: Alternative to OVS for some deployment scenarios

- Performance Validation
  - Real world scenarios: OpenStack
  - Via OPNFV FDS
  - Real World Use Cases
    - NFV
    - Open Stack - OPNFV Deployments
- vxLAN - NSH/SFC
  - Likely to be merged upstream sooner than OVS/DPDK
<table>
<thead>
<tr>
<th>Open vSwitch</th>
<th>FD.IO/VPP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Upstream</strong></td>
<td><strong>Upstream</strong></td>
</tr>
<tr>
<td>vHost multi-queue -- perf performances</td>
<td>vxLAN - NSH for Alternative to OVS</td>
</tr>
<tr>
<td>Container Acceleration</td>
<td>HoneyComb Yang</td>
</tr>
<tr>
<td>DPDK Development</td>
<td>CSIT -- Continuous Testing Infrastructure</td>
</tr>
<tr>
<td>DPDK “Usability”</td>
<td>Performance Validation</td>
</tr>
<tr>
<td>Get rid of “Experimental”</td>
<td>Container Performance</td>
</tr>
<tr>
<td>vxLAN NSH and VTEP Acceptance</td>
<td><strong>OPNFV and MidStream</strong></td>
</tr>
<tr>
<td></td>
<td><strong>OPNFV and MidStream</strong></td>
</tr>
<tr>
<td><strong>OPNFV and MidStream</strong></td>
<td><strong>OPNFV and MidStream</strong></td>
</tr>
<tr>
<td>Deploy OVS/DPDK C Release</td>
<td>FDS - Fast Data Stacks</td>
</tr>
<tr>
<td>Apex and Fuel Installers</td>
<td>Apex Installer</td>
</tr>
<tr>
<td>vxLAN NSH and VTEP</td>
<td>“D” Release - Real World NFV Deployment</td>
</tr>
<tr>
<td>Vsperf</td>
<td>Vsperf</td>
</tr>
<tr>
<td>OVS4NFV - QoS Perf Req</td>
<td>Security Groups Policy QoS</td>
</tr>
</tbody>
</table>
References and Credits

VPP fd.io

Open Source Project: fd.io


https://fd.io/technology

fd.io performance


Open vSwitch/DPDK performance


http://www.slideshare.net/harryvanhaaren/ovs-and-dpdk-tf-herbert-k-traynor-m-gray