OVS, DPDK and Software Dataplane Acceleration

FOSDEM 2016
Kevin Traynor <kevin.traynor@intel.com>
Legal Disclaimer

General Disclaimer:

© Copyright 2016 Intel Corporation. All rights reserved. Intel, the Intel logo, Intel Inside, the Intel Inside logo, Intel. Experience What's Inside are trademarks of Intel. Corporation in the U.S. and/or other countries. *Other names and brands may be claimed as the property of others.

Technology Disclaimer:

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com].

Performance Disclaimers (include only the relevant ones):

Cost reduction scenarios described are intended as examples of how a given Intel-based product, in the specified circumstances and configurations, may affect future costs and provide cost savings. Circumstances will vary. Intel does not guarantee any costs or cost reduction.

Results have been estimated or simulated using internal Intel analysis or architecture simulation or modeling, and provided to you for informational purposes. Any differences in your system hardware, software or configuration may affect your actual performance.
Background

---

**DPDK Sample Applications**

- EAL
- MALLOC
- MBUF
- MEMPOOL
- RING
- TIMER
- IVSHMEM
- Core Libraries
- Platform

**Customer Applications**

- ETHDEV
- E1000
- IXGBE
- I40e
- VMXNET3
- VIRTIO
- XENVIRT
- Mellanox
- Cisco VIC
- Packet Access (PMD – Native & Virtual)

**ISV Eco-System Applications**

- LPM
- EXACT MATCH
- ACL
- Classify
- METER
- SCHED
- QoS

---

**OvS (Open vSwitch)**

- Security: VLAN isolation, traffic filtering
- Monitoring: Netflow, sFlow, SPAN, RSPAN
- QoS: traffic queuing and traffic shaping
- Automated Control: OpenFlow, OVSDB mgmt. protocol

---

**DPDK Data Plane Development Kit**

- Core Switch
- Aggregation Switch
- "Top of the Rack" Switch

---

**VMs**

- VM
- VM
- VM
- VM

---

**Applications**

- Linux Kernel
- User Space
- Customer Applications
- ISV Eco System
- System Applications

---

**Platforms**

- MALLOCS
- DISTRIB
- Extensions
- Packet Work

---

**Intel**

- KNI
- Linux Kernel
- IGB_UIO

---

**FOSDEM 2016**
Architecture
OpenvSwitch with DPDK
OVS Tables

Exact Match Cache
- Logically, Single Table per datapath thread
- Exact Match
- 8192 entries / per thread

Datapath Classifier
- Logically, Single Table per datapath thread
- Wildcard Matches
- 65536 entries

Ofproto Classifier
- Logically, Multiple (up to 255) Open Flow tables in pipeline per Open vSwitch bridge
- Wildcard Matches

Cost of lookup increasing

execute
action

rx

virtual
physical

tx

FOSDEM 2016
OVS Tables

rx cost

lookup cost

Virtual physical

tx cost

action cost
Performance
# OpenvSwitch 2.4

## Platform Performance Configuration

<table>
<thead>
<tr>
<th><strong>Item</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
</table>
| **Server Platform** | Intel® Server Board S2600WT2 DP (Formerly Wildcat Pass)  
2 x 1GbE integrated LAN ports  
Two processors per platform |
| **Chipset**    | Intel® C610 series chipset (Formerly Wellsburg)                                                                                                  |
| **Processor**  | Intel® Xeon® Processor E5-2697 v3 (Formerly Haswell)  
Speed and power: 2.60 GHz, 145 W  
Cache: 35 MB per processor  
Cores: 14 cores, 28 hyper-threaded cores per processor for 56 total hyper-threaded cores  
QPI: 9.6 GT/s  
Memory types: DDR4-1600/1866/2133,  
| **Memory**     | Micron 16 GB 1Rx4 PC4-2133MHz, 16 GB per channel, 8 Channels, 128 GB Total                                                                            |
| **Local Storage** | 500 GB HDD Seagate SATA Barracuda 7200.12 (SN:9VMKQZMT)                                                                                           |
| **PCIe**       | Port 3a and Port 3c x8                                                                                                                              |
| **NICs**       | 2 x Intel® Ethernet CAN X710-DA2 Adapter (Total: 4 x 10GbE ports) (Formerly Fortville)                                                              |
| **BIOS**       | Version: SE5C610.86B.01.01.0008.021120151325  
Date: 02/11/2015                                                                                                                                 |

---

FOSDEM 2016
OpenvSwitch 2.4
Phy-OVS-Phy Performance

Disclaimer: For more complete information about performance and benchmark results, visit www.intel.com/benchmarks and https://download.01.org/packet-processing/ONPS1.5/Intel_ONP_Server_Release_1.5_Performance_Test_Report_Rev1.2.pdf
OpenvSwitch 2.4
Phy-VM-Phy Performance

Disclaimer: For more complete information about performance and benchmark results, visit www.intel.com/benchmarks and https://download.01.org/packet-processing/ONPS1.5/Intel_ONP_Server_Release_1.5_Performance_Test_Report_Rev1.2.pdf
OpenvSwitch 2.4
Phy-OVS Tunnel-Phy Performance

Disclaimer: For more complete information about performance and benchmark results, visit www.intel.com/benchmarks and https://download.01.org/packet-processing/ONPS1.5/Intel_ONP_Server_Release_1.5_Performance_Test_Report_Rev1.2.pdf
OpenvSwitch 2.x DPDK 2.x
netdev-DPDK Performance Enhancements

- Vector Tuple Extractor
- DPDK Hash
- DPDK ACL tables

- Control Path
- Slab Path
- User Space Forwarding
- Tunnels
- netdev
- TAP
- socket

- VM
- DPDK
- ivshmem
- qemu
- Rings
- vHost

- VM
- virtio
- qemu

- Offloads
  - Wildcards

- Tunnel processing

- Virtio ordering
- vHost Bulk Alloc/Free
- Multiqueue vhost-user
usable

*adjective*  |  usable  | ˈyū-za-bəl*

**Simple Definition of Usable**

: capable of being used : in good enough condition to be used

**Full Definition of Usable**

1 : capable of being used

2 : convenient and practicable for use

—*usabil-i-ty* ˈyū-zə-bəl-i-tē noun

—*us-able-ness* ˈyū-zə-bəl-nəs noun

—*us-ably* ˈbliə adverb
Usability
Usability Examples

Cmd Line args - ovs-vswitchd --dpdk -c 0x40 -n 4 --socket-mem 1024,0


Testing - VSPERF – OPNFV project

- https://wiki.opnfv.org/characterize_vswitch_performance_for_telco_nfv_use_cases

DPDK Device Management - Driverctl


DPDK Debug/tcpdump – Several ideas proposed


Documentation

- https://github.com/openvswitch/ovs/blob/master/INSTALL.DPDK.md

Out of the box Performance

- ovs-vsctl --no-wait set Open_vSwitch . other_config:pmd-cpu-mask=f
Availability: Distro Packages and Git

- ISV and OSV recognizing the progress of OVS with DPDK
- Centos7: 7.4: DPDK 2.1; 7.2: OVS 2.4
- Fedora: F23; F22 updates DPDK 2.0; F24: DPDK 2.1
- Fedora Copr repo for latest: https://copr.fedoraproject.org/coprs/pmatilai/dpdk/
- Red Hat OSP8:
  - OVS 2.4/DPDK 2.0 Integrated with Neutron
  - Ubuntu: 15.10: OVS with DPDK package
- OVSNFV OPNFV Project planning deployment in future OPNFV releases
  - https://01.org/packet-processing/intel%C2%AE-onp-servers
  - git clone http://dpdk.org/git/dpdk
  - git clone https://github.com/openvswitch/ovs.git
Wrap-up

- Feel free to join us in the OVS-DPDK lane...
  - Performance
  - Usability
  - Testing
    - http://openvswitch.org/mlists/
    - http://dpdk.org/ml
- But not like this…

I don't always violate the HOV lane law...but when I do, I get a $124 ticket! We'll give him an A for creativity! 😎👍
Backup
OVS-DPDK Setup #1

# Build DPDK
export RTE_SDK=/home/ktraynor/vswitch/ovs/code/dpdk_210
cd /home/ktraynor/vswitch/ovs/code/dpdk_210
make install T=x86_64-native-linuxapp-gcc CONFIG_RTE_BUILD_COMBINE_LIBS=y

# Build OVS
./boot.sh
./configure --with-dpdk=/home/ktraynor/vswitch/ovs/code/dpdk_210/x86_64-native-linuxapp-gcc
make 'CFLAGS=-g -Ofast -march=native' ; make install

# Mount Hugepages
mkdir -p /mnt/huge
mount -t hugetlbfs nodev /mnt/huge
echo 64 > /sys/devices/system/node/node0/hugepages/hugepages-2048kB/nr_hugepages

# Bind ports to DPDK
modprobe uio
insmod $dpdk_dir/x86_64-native-linuxapp-gcc/kmod/igb_uio.ko
$dpdk_dir/tools/dpdk_nic_bind.py -b igb_uio 05:00.0 05:00.1
OVS-DPDK Setup #2

# Run vswitchd
```bash
cd /usr/local
ovs-vswitchd --dpdk -c 0x40 -n 4 --socket-mem 1024,0 -- unix:/usr/local/var/run/openvswitch/db.sock
```

# Set Forwarding cores
```bash
ovs-vsctl --no-wait set Open_vSwitch . other_config:pmd-cpu-mask=f
```

# Add Bridge, Ports and Rule
```bash
ovs-vsctl add-br br0 -- set bridge br0 datapath_type=netdev
ovs-vsctl add-port br0 dpdk0 -- set Interface dpdk0 type=dpdk
ovs-vsctl add-port br0 dpdk1 -- set Interface dpdk1 type=dpdk
```
```bash
ovs-ofctl add-flow br0 in_port=1,action=output:2
```

# Debug
```bash
ovs-ofctl add-flow br0 in_port=1,action=output:2, LOCAL
tcpdump -i <ip of LOCAL port>
```
OVS Setup

# Build OVS
./boot.sh
./configure
make ; make install
# Add OVS kernel module
modprobe openvswitch
# Run vswitchd
ovs-vswitchd unix:/usr/local/var/run/openvswitch/db.sock
# Add Bridge, Ports and Rule
ovs-vsctl add-br br0
ovs-vsctl add-port br0 p3p1
ovs-vsctl add-port br0 p3p2
ovs-ofctl add-flow br0 in_port=1,action=output:2
# Debug
tcpdump -i p3p2
Paths to the guest

QEMU

Guest OS

Virtio Driver

DPDK vhost user

OVS Datapath

DPDK x

UDS

Kernel Space

QEMU

Guest OS

OVS client

DPDK Ring API

PCI dev (04:00.0)

BAR2

ivshmem

Memory

mempool

Kernel Space

OVS Datapath

DPDK PMD

DPDK Ring API

UDS

1GB