Agenda

• DPDK vHost User Introduction/Refresh
• Time Line of DPDK vHost User in OVS
• Recent Improvements
  • NUMA Awareness
  • Client Mode & Reconnect
• Future Improvements
  • vHost User PMD
  • Zero Copy
What is DPDK vHost User?
What is DPDK?
What is DPDK?

- Data Plane Development Kit
What is DPDK?

• Data Plane Development Kit

• Userspace drivers & libraries for accelerated network I/O
What is DPDK?

• Data Plane Development Kit
• Userspace drivers & libraries for accelerated network I/O
• Integrated into OVS in v2.2
What is DPDK?

- Data Plane Development Kit
- Userspace drivers & libraries for accelerated network I/O
- Integrated into OVS in v2.2
What is DPDK?

- Data Plane Development Kit
- Userspace drivers & libraries for accelerated network I/O
- Integrated into OVS in v2.2
Accelerated guest access method offered by DPDK capable of outperforming traditional methods by >8x*

* Platform Configuration and Test Result in Backup
Timeline of vHost User in OVS

- **Kernel vHost**: OVS Release 2.2
- **vHost User**: OVS Release 2.4
- **NUMA Awareness**: OVS Release 2.5
- **Client Mode & Reconnect**: OVS Release 2.6
- **Zero Copy**: TBD

- **vHost Cuse**: OVS Release 2.4
- **Multi-queue**: OVS Release 2.4
- **Cuse removal**: OVS Release 2.6
- **vHost PMD**: OVS Release 2.6
- **Other**: TBD

- Black circle = Functional Improvement
- White circle = Performance Improvement
Timeline of vHost User in OVS

- Kernel vHost
- vHost User
- NUMA Awareness
- Client Mode & Reconnect
- Zero Copy

vHost Cuse
Multi-queue
Cuse removal
vHost PMD
Other

OVS Release Version
2.2

= Functional Improvement
= Performance Improvement
Timeline of vHost User in OVS

- Kernel vHost
- vHost User
- NUMA Awareness
- Client Mode & Reconnect
- Zero Copy

OVS Release Version

- 2.4

vHost Cuse

- Multi-queue
- Cuse removal
- vHost PMD
- Other

= Functional Improvement

= Performance Improvement
Timeline of vHost User in OVS

- Kernel vHost User
- NUMA Awareness
- Client Mode & Reconnect
- Zero Copy

- vHost Cuse
- Multi-queue
- Cuse removal
- vHost PMD
- Other

= Functional Improvement
= Performance Improvement

OVS Release Version: 2.4
Timeline of vHost User in OVS

- Kernel vHost
- vHost User
- NUMA Awareness
- Client Mode & Reconnect
- Zero Copy

OVS Release Version

- vHost Cuse
- Cuse removal
- vHost PMD
- Other

Multi-queue

= Functional Improvement
= Performance Improvement
Timeline of vHost User in OVS

- Kernel vHost
- vHost User
- Multi-queue
- Cuse removal
- vHost PMD
- Zero Copy

OVS Release Version

vHost Cuse

NUMA Awareness

- 2.6

Client Mode & Reconnect

Zero Copy

Other

= Functional Improvement
= Performance Improvement
Timeline of vHost User in OVS

- **Kernel vHost**
- **vHost User**
- **NUMA Awareness**
- **Client Mode & Reconnect**
- **Zero Copy**

**OVS Release Version**

- **vHost Cuse**
- **Multi-queue**
- **vHost PMD**
- **Other**

**Cuse removal**

- **2.6**

- **= Functional Improvement**
- **= Performance Improvement**
Timeline of `vHost User` in `OVS`
Timeline of vHost User in OVS

- Kernel vHost
- vHost User
- NUMA Awareness
- Client Mode & Reconnect
- Zero Copy
- vHost Cuse
- Multi-queue
- Cuse removal
- vHost PMD
- Other

Symbols:
- Black circle = Functional Improvement
- White circle = Performance Improvement

OVS Release Version

TBD
Timeline of vHost User in OVS

Kernel vHost

vHost User

NUMA Awareness

Client Mode & Reconnect

Zero Copy

vHost Cuse

Multi-queue

Cuse removal

vHost PMD

TBD

OVS Release Version

= Functional Improvement

= Performance Improvement

Other
NUMA Awareness

Kernel vHost

vHost Cuse

vHost User

Multi-queue

Cuse removal

Client Mode & Reconnect

vHost PMD

Zero Copy

Other

OVS Release Version

2.6
NUMA Awareness

QEMU, DPDK & OVS vHost memory need to be co-located for optimal performance.
QEMU, DPDK & OVS vHost memory need to be co-located for optimal performance.
QEMU, DPDK & OVS vHost memory need to be co-located for optimal performance.
NUMA Awareness

QEMU, DPDK & OVS vHost memory need to be co-located for optimal performance.
NUMA Awareness

Previous limitation:
All DPDK vHost memory must come from the same NUMA node.
NUMA Awareness

Previous limitation:
All DPDK vHost memory must come from the same NUMA node.
Previous limitation:
All DPDK vHost memory must come from the same NUMA node.
Previous limitation:
All DPDK vHost memory must come from the same NUMA node.
Solution:
DPDK vHost memory relocated to correct NUMA node on VM boot.
NUMA Awareness

Previous limitation:
All PMDs servicing vHost ports must come from the same NUMA node.
Solution:
mbufs and servicing PMD in OVS are moved to correct NUMA during DPDK callback.
Solution:
mbufs and servicing PMD in OVS are moved to correct NUMA during DPDK callback.
NUMA Awareness

Can achieve >50% improvement in second socket VM2VM performance*

* Platform Configuration and Test Result in Backup
NUMA Awareness

Without NUMA Awareness

With NUMA Awareness

Can achieve >50% improvement in second socket VM2VM performance*

Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset
Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset

OVS by default acts as the socket server
Client Mode & Reconnect

Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset

OVS by default acts as the socket server
Client Mode & Reconnect

Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset

OVS by default acts as the socket server

Default Mode (Server)

QEMU (client mode)

Guest
eth0

Host
vhost0

OVS DPDK (server mode)

🌟 = creates/manages/destroys sockets
Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset

OVS by default acts as the socket server
Client Mode & Reconnect

Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset

OVS by default acts as the socket server
Client Mode & Reconnect

Previous Limitation:
VMs cannot easily regain connectivity if OVS DPDK crashes or is reset

OVS by default acts as the socket server
Solution:
QEMU creates the socket and acts as the server instead.
Solution: QEMU creates the socket and acts as the server instead.
Solution:
QEMU creates the socket and acts as the server instead.

VMs can reconnect to OVS.

New Mode (Client)

QEMU (server mode)

Guest

eth0

Host

vhost0

OVS DPDK (client mode)

= creates/manages/destroys sockets
Solution:
QEMU creates the socket and acts as the server instead

VMs can reconnect to OVS

vHost PMD

libraries

librte_mbuf

librte_malloc

librte_mempool

librte_vhost
vHost PMD

libraries
- librte_mbuf
- librte_malloc
- librte_mempool
- librte_vhost

drivers
- ixgbe
- i40e
- mlx
vHost PMD

Core Libraries

libraries
- librte_mbuf
- librte_malloc
- librte_mempool
- librte_vhost

drivers
- ixgbe
- i40e
- mlx
vHost PMD

Core Libraries

libraries
- librte_mbuf
- librte_malloc
- librte_mempool
- librte_vhost

drivers
- ixgbe
- i40e
- mlx

librteEther DPDK API
vHost PMD

Core Libraries

- librte_mbuf
- librte_malloc
- librte_mempool
- librte_vhost

Drivers

- ixgbe
- i40e
- mlx

Libraries

- librte_ether

DPDK API

Open vSwitch
vHost PMD

Core Libraries

- librte_mbuf
- librte_malloc
- librte_mempool
- librte_vhost

DPDK API

- ixgbe
- i40e
- mlx
- vhost

- Simplified code path
- Little difference in usability/performance
- Easier future vHost feature integration in OVS

librteEther

Open vSwitch
Zero Copy

- Kernel vHost
- vHost User
- NUMA Awareness
- Client Mode & Reconnect
- vHost PMD
- Other

TBD

OVS Release Version

vHost Cuse
Multi-queue
Cuse removal
vHost PMD
Other
DPDK 16.11 performance improvement
DPDK 16.11 performance improvement

Both dequeue (rx) and enqueue (tx) paths usually incur a copy.
Dequeue path involves copying a packet from the VM to the host.
Zero Copy

Dequeue path involves copying a packet from the VM to the host
Dequeue path involves copying a packet from the VM to the host
Dequeue path involves copying a packet from the VM to the host.
Enqueue path involves copying a packet from the host to the VM.
Enqueue path involves copying a packet from the host to the VM.
Enqueue path involves copying a packet from the host to the VM.
Enqueue path involves copying a packet from the host to the VM
Zero copy is possible for dequeue if the mbuf references the virtio descriptor buffer **directly**.
Zero copy is possible for dequeue if the mbuf references the virtio descriptor buffer directly.
Zero Copy

Not suitable for small packet sizes (~ < 512B)
Zero Copy

Can achieve >15% increase in throughput for 1518B packets for this use case*

(vHost ➔ OVS-DPDK ➔ vHost)

* Platform Configuration and Test Result in Backup
Other Future Improvements

- Kernel vHost
- vHost User
- NUMA Awareness
- Client Mode & Reconnect
- Zero Copy

- vHost Cuse
- Multi-queue
- Cuse removal
- vHost PMD

= Functional Improvement
= Performance Improvement

OVS Release Version

Other
• Virtio User (16.11)
  • New “PMD”
  • Method of using vHost User in containers
Other Future Improvements

• Virtio User (16.11)
  • New “PMD”
  • Method of using vHost User in containers
• Mergeable buffers path improvement (16.11)
Other Future Improvements

- Virtio User (16.11)
  - New “PMD”
  - Method of using vHost User in containers
- Mergeable buffers path improvement (16.11)
- vHost PCI (POC)
  - VM2VM path performance enhancement
  - vHost vEth pair
Since it’s introduction to OVS in 2015, many incremental improvements to DPDK vHost User have been added.

Many more improvements to look forward to.
Questions?
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.
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
</tr>
</thead>
</table>
| Server Platform | Intel® Server Board S2600WTT (Formerly Wildcat Pass)  
2 x 1GbE integrated LAN ports  
Two processors per platform |
| Chipset    | Intel® C610/X99 series chipset (Formerly Wellsburg)                                                                                        |
| Processor  | Intel® Xeon® Processor E5-2695 v3 (Formerly Haswell)  
Speed and power: 2.30 GHz, 120 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                                                                               |
| NICs       | 2 x Intel® Ethernet CAN X710 Adapter (Total: 4 x 10GbE ports)  
(Formerly Fortville)                                                                  |
| BIOS       | Version: SE5C610.86B.01.01.0008.021120151325  
Date: 02/11/2015                                                                            |
| OS         | Fedora 22                                                                                                                                 |
| Software   | DPDK - v2.2.0, OVS – v2.5.0 pre-release (commit 522aca), QEMU – 2.3.0, Linux kernel – 4.0.6-300.fc22.x86_64                                |

### Guest Access Method

<table>
<thead>
<tr>
<th>Method</th>
<th>Packets per Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtio-net</td>
<td>51131</td>
</tr>
<tr>
<td>vhost-net</td>
<td>406515</td>
</tr>
<tr>
<td>vhost-user</td>
<td>3366374</td>
</tr>
</tbody>
</table>
## Platform Configuration & Test Results

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Packets per Second</th>
</tr>
</thead>
</table>
| **Server Platform** | Intel® Server Board S2600WTT (Formerly Wildcat Pass)  
2 x 1GbE integrated LAN ports  
Two processors per platform | Without NUMA Awareness: 2545945  
With NUMA Awareness: 3831019 |
| **Chipset**   | Intel® C610/X99 series chipset (Formerly Wellsburg)                                                                                                                                                           |--------------------|
| **Processor** | Intel® Xeon® Processor E5-2695 v3 (Formerly Haswell)  
Speed and power: 2.30 GHz, 120 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**    | Micron 16 GB 1Rx4 PC4-2133MHz, 16 GB per channel, 8 Channels                                                                                                                                                   |--------------------|
| **NICS**      | 2 x Intel® Ethernet CAN X710 Adapter (Total: 4 x 10GbE ports)  
(Formerly Fortville)                                                                                                                                                                                         |--------------------|
| **BIOS**      | Version: SE5C610.86B.01.01.0008.021120151325  
Date: 02/11/2015                                                                                                                                                                                                |--------------------|
| **OS**        | Fedora 22                                                                                                                                                                                                   |--------------------|
| **Software**  | DPDK – v16.07, OVS – v2.6.0 (commit 136e425df951), QEMU – 2.7.0, Linux kernel – 4.2.8-200.fc22.x86_64                                                                                                          |--------------------|
## Platform Configuration & Test Results

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Packets per Second</th>
</tr>
</thead>
</table>
| **Server Platform** | Intel® Server Board S2600WTT (Formerly Wildcat Pass)  
2 x 1GbE integrated LAN ports  
Two processors per platform |                     |
| **Chipset**  | Intel® C610/X99 series chipset (Formerly Wellsburg)                         |                     |
| **Processor** | Intel® Xeon® Processor E5-2695 v3 (Formerly Haswell)  
Speed and power: 2.30 GHz, 120 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                |                     |
| **NICs**     | 2 x Intel® Ethernet CAN X710 Adapter (Total: 4 x 10GbE ports)               |                     |
| **BIOS**     | Version: SE5C610.86B.01.01.0008.021120151325  
Date: 02/11/2015                        |                     |
| **OS**       | Fedora 22                                                                   |                     |
| **Software** | DPDK – v16.11-rc2, OVS – v2.6.0 (commit 136e425df951, patched to enable feature), QEMU – 2.7.0, Linux kernel – 4.2.8-200.fc22.x86_64 |                     |

<table>
<thead>
<tr>
<th></th>
<th>Without zero copy</th>
<th>2094554</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>With zero copy</td>
<td>2415784</td>
</tr>
</tbody>
</table>