
DPDK
DATA PLANE DEVELOPMENT KIT
Userspace 2015 | Dublin
OVS, DPDK and
Software Dataplane Acceleration

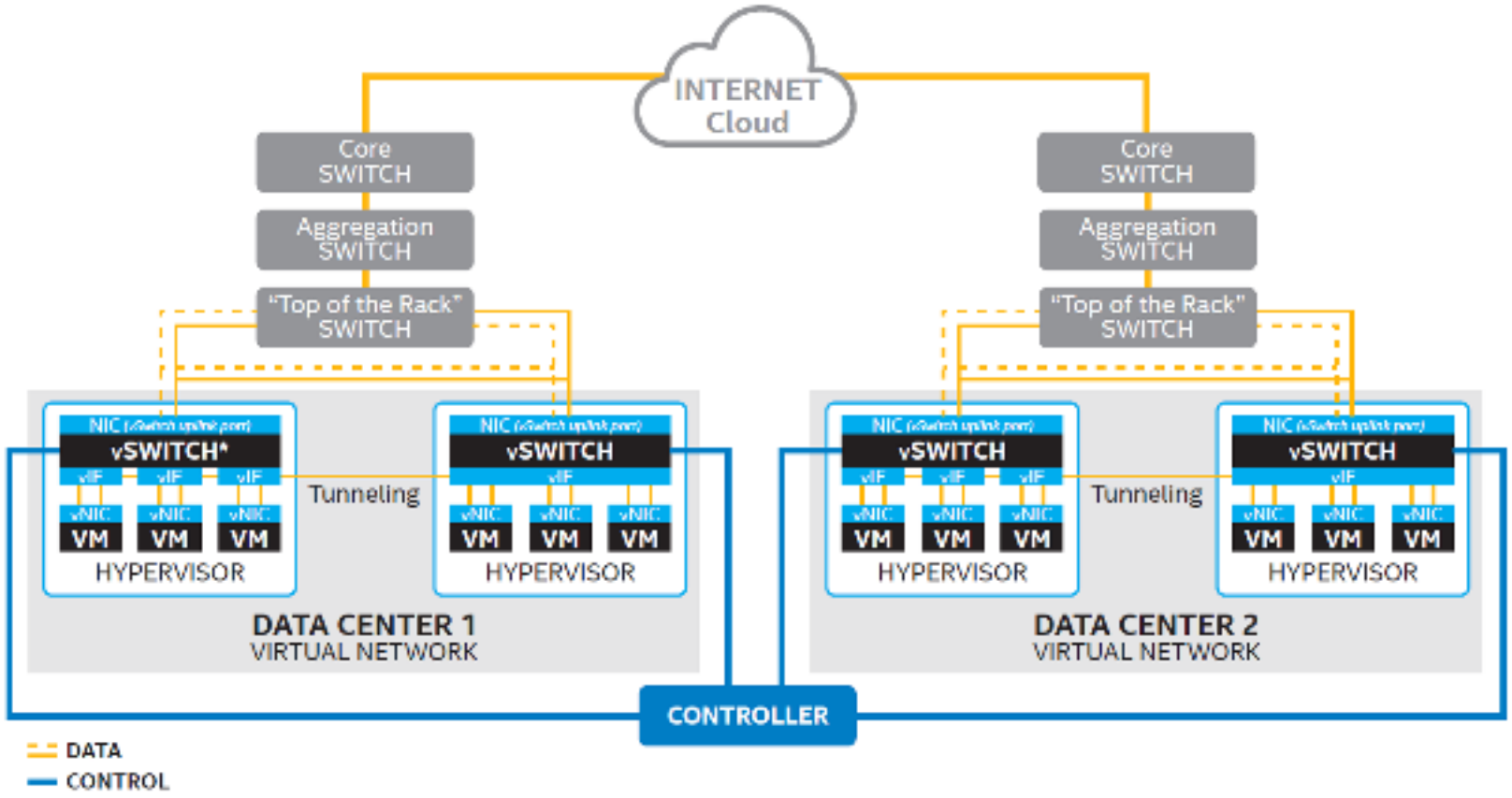
Who we are?

DPDK Userspace 2015

- Thomas F. Herbert
  - Red Hat
  - therbert@redhat.com
- Kevin Traynor
  - Intel
  - kevin.traynor@intel.com
- Mark Gray
  - Intel
  - mark.d.gray@intel.com

What is a Virtual Switch?

DPDK
Userspace 2015

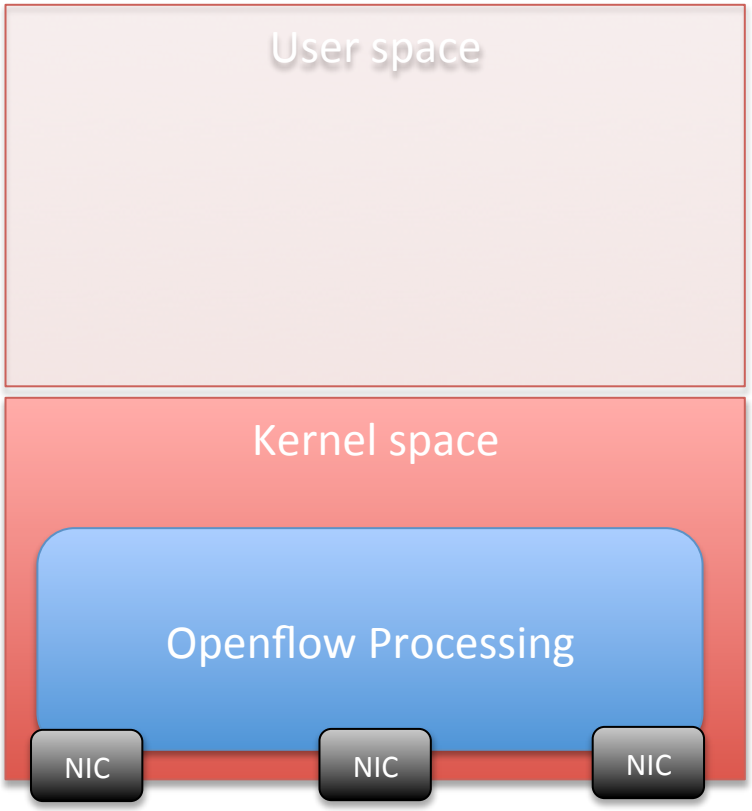


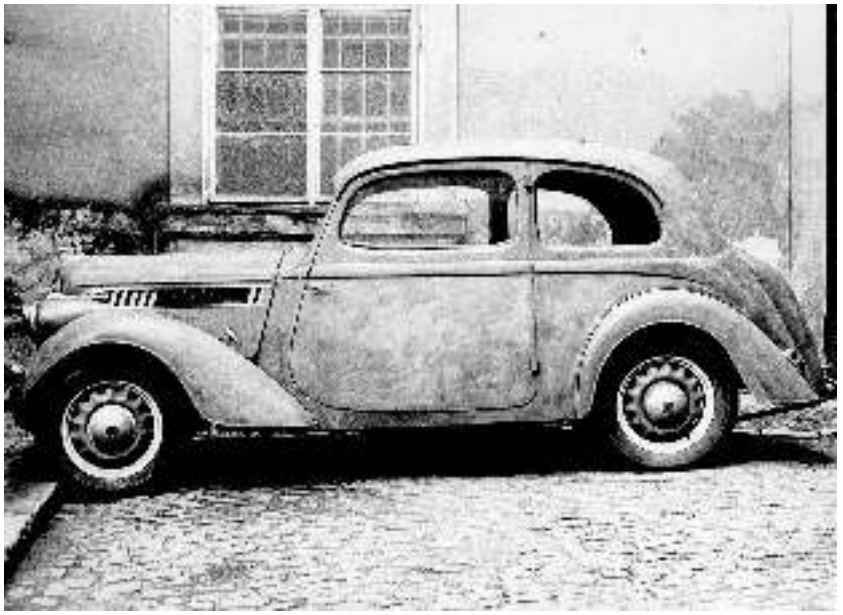
DPDK

Architecture

OVS Architectural Evolution

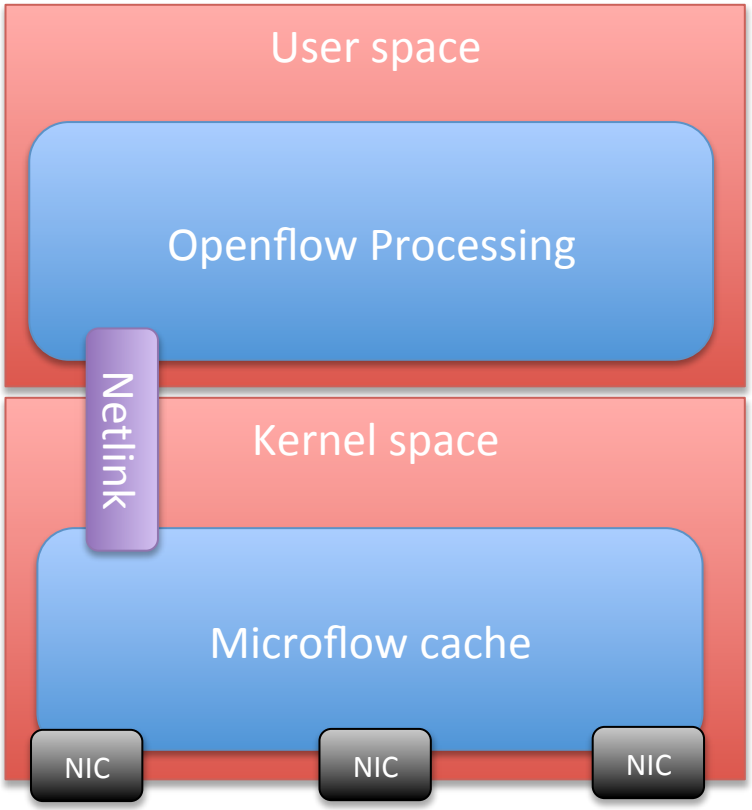
DPDK
Userspace 2015





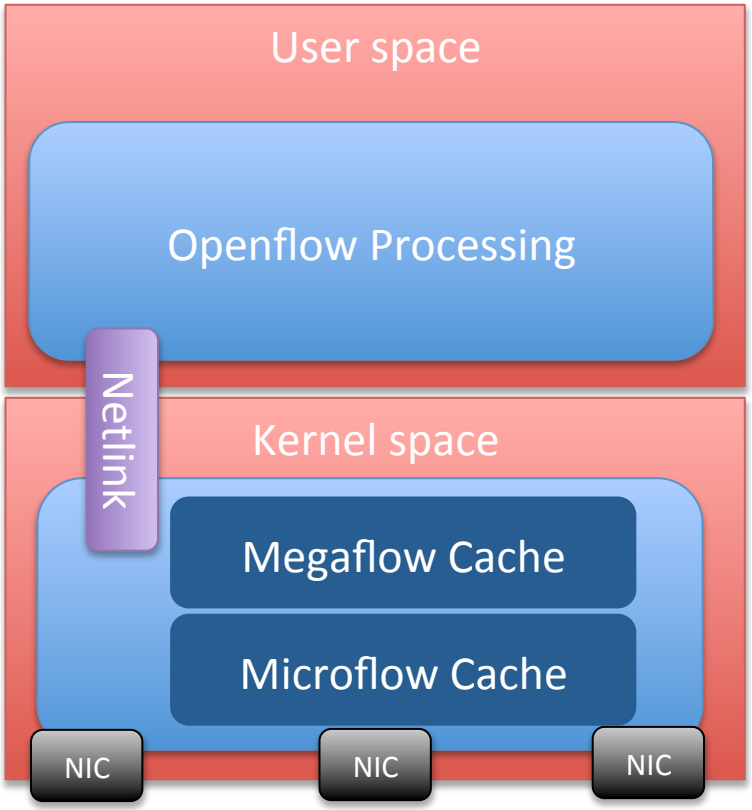
OVS Architectural Evolution

DPDK
Userspace 2015



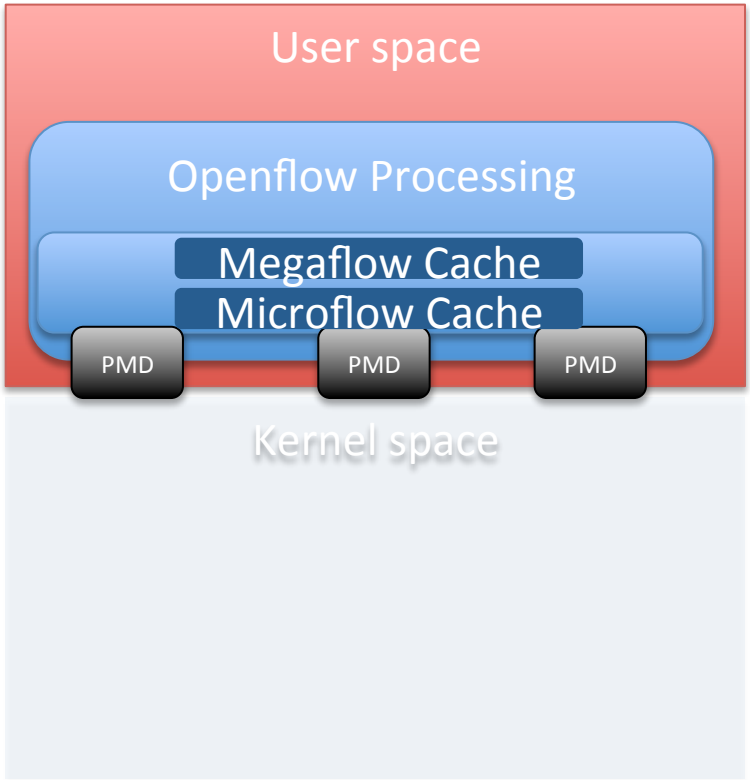
OVS Architectural Evolution

DPDK
Userspace 2015



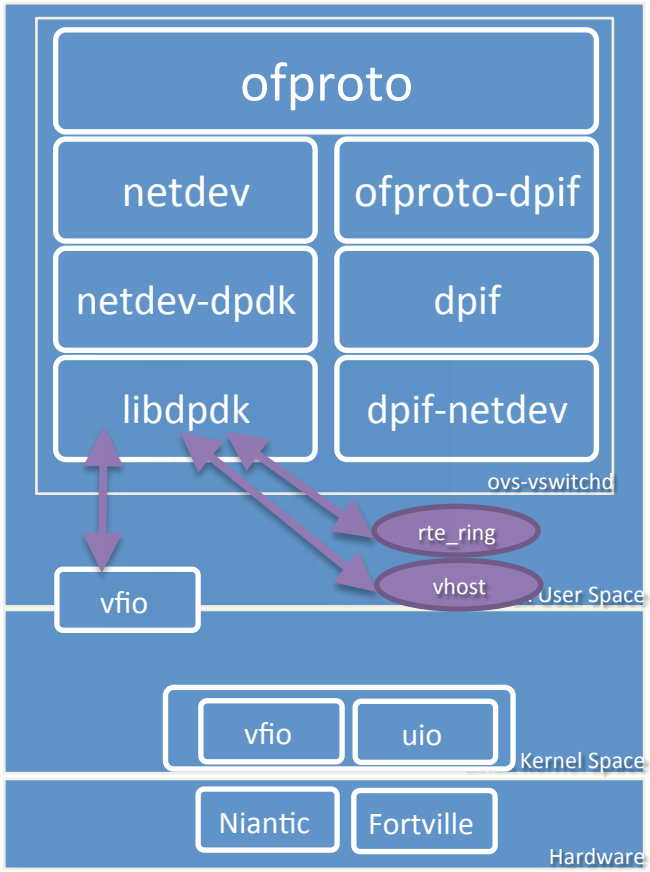
OVS Architectural Evolution

DPDK
Userspace 2015



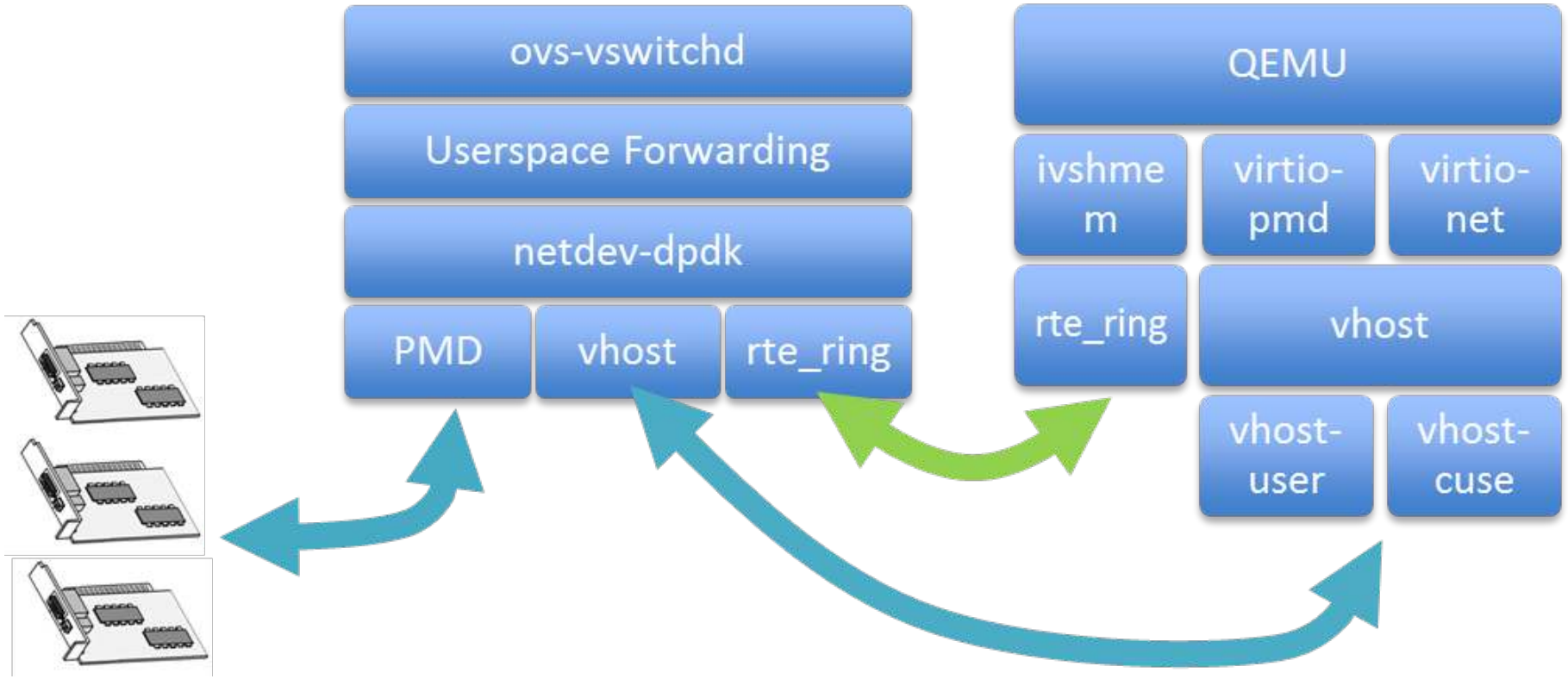
OVS Architecture

DPDK
Userspace 2015



OVS Guest Interfaces

DPDK
Userspace 2015



DPDK

Usability

DPDK – Open vSwitch

- DPDK - Popular Software Accelerated Data Plane
  - Fast Packet Forwarding for the Cloud
    - Virtualized Network Functions
  - Use of Commodity Hardware
  - For Basic OpenFlow Switch Functions
    - Behaves Identical to Linux Kernel
- Advantages and Disadvantages WRT Linux Kernel
- Linux Data Plane Has
  - 20 years of development
  - Rich Debugging Options
    - Packet Dumping
  - Access to Wide Variety of Network IF's and VF's
  - Full set of device statistics
  - More Tunnel Endpoints
- DPDK Data Plane
  - Much Faster Packet Forwarding
    - Up to 12X for small packets

DPDK/OVS User Perspective

DPDK Userspace 2015

- How About the “User” of OVS/DPDK
  - Controllers using OF and OVSDB protocols
  - People Using OVS CLI Tools
  - Network Engineers building complex topologies
  - Cloud Deployements
  - Programmers - Application Developers of
    - Other Packet consumers, DPI, Classifiers
    - Infrastructure – Routers, Firewalls, Services
    - Other Packet consumers, DPI, Classifiers
- One person’s experience
- What do user’s want
  - Expectations of DPDK/OVS vs Linux Kernel/OVS

DPDK/OVS Usability Story

DPDK Userspace 2015

- In the Beginning: My user Story starts in 2013
  - Inspired by Intel presentation of DPDK at ONS 2013
- On Team Developing SDN Network Threat Analyzer controlled
  - Integrated Open vSwitch
    - First with Linux Kernel Data Path
  - Traffic shaping, threat blocking and mitigation
  - Requirement: 10Gb without Adding $10K to $20K on custom HW Switch Fabric.
    - DPDK is the Answer?
    - How to prove the OVS/DPDK Claim?
- At first Started with DPDK 1.7.1
- Scary: poor integration --Not integrated with OVS
  - Compilation issues, conflicting APIs. ABIs, OVS Versions
  - Three Confusing Forks:
    1. DPDK.org
    2. DPI Fork with custom API
    3. 01.org
  - Then came DPDK 1.8
    - Integrated: Master Branch OVS
    - I Ran DPDK on guest with VirtIO/ VMXnet3 saw 2.5X perf gain
    - Developed App using DPDK-ring to feed DPI
  - Now we are up to DPDK 2.1 with OVS.
    - Much much improved!

The Netdev Interface to OVS

- Transparency of Data Plane
- Netdev – API Between Data Plane and OVS
  - Generic network device API independent from data plane implementation.
  - Similar to network driver interface in BSD
  - Netdev Abstracts forwarding of packets in data plane
- Conceptually like any Network device driver
  - With Start, Stop, Private Data Area, Queue Management
- Struct netdev Holds the interface Specific Function Pointers
  - Includes the generic part followed by private part for use by driver.
  - Constructur for netdev provider
  - Dpdk Creates dpdk personality of struct netdev
    - Multiple rx queues Managed by OVS

Improving DPDK/OVS

- Is DPDK really still Experimental?
  - Is it time for this patch?

```
--- a/INSTALL.DPDK.md
+++ a/INSTALL.DPDK.md
@@ -5,8 +5,8 @@ Open vSwitch can use Intel® DPDK lib to operate entirely in
Userspace. This file explains how to install and use Open vSwitch in such a mode.

-The DPDK support of Open vSwitch is considered experimental.
-It has not been thoroughly tested.

This version of Open vSwitch should be built manually with `configure` and `make`.
```

- Issues with DPDK:
  - How to Improve?
    - This thread, **http://openvswitch.org/pipermail/dev/2015-August/058814.html**
  - Some Suggestions from Thread
    - Device management:
    - Udev/systemd – (Flavio Leitner)
      - Device creation, binding, destruction – handled by Host OS

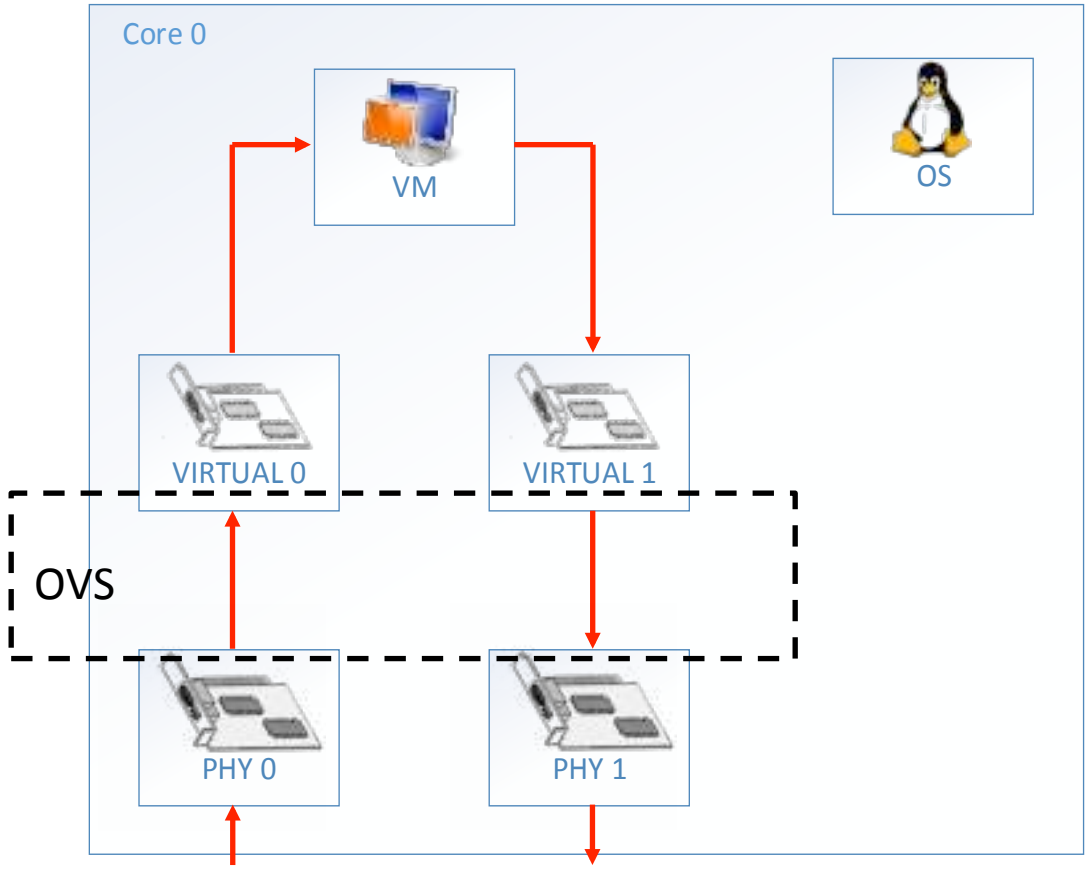
Improving DPDK/OVS Contd.

- How to Improve?
  - Debugging?
    - TcpDump like capability
    - Use “Mirroing” of packets to pmd/libpcap or libpcap-ng
  - Testing
  - Add CI for Data Plane Testing
  - Vsperf Project – To Test Against Goals
  - Support Only One type of vhost device
    - Drop Vhost - Cuse
    - Vhost-user only
  - Better Documentation
    - Recent Patch to INSTALL.DPDK.md
  - Training
    - From lstopo to Optimized Data Plane

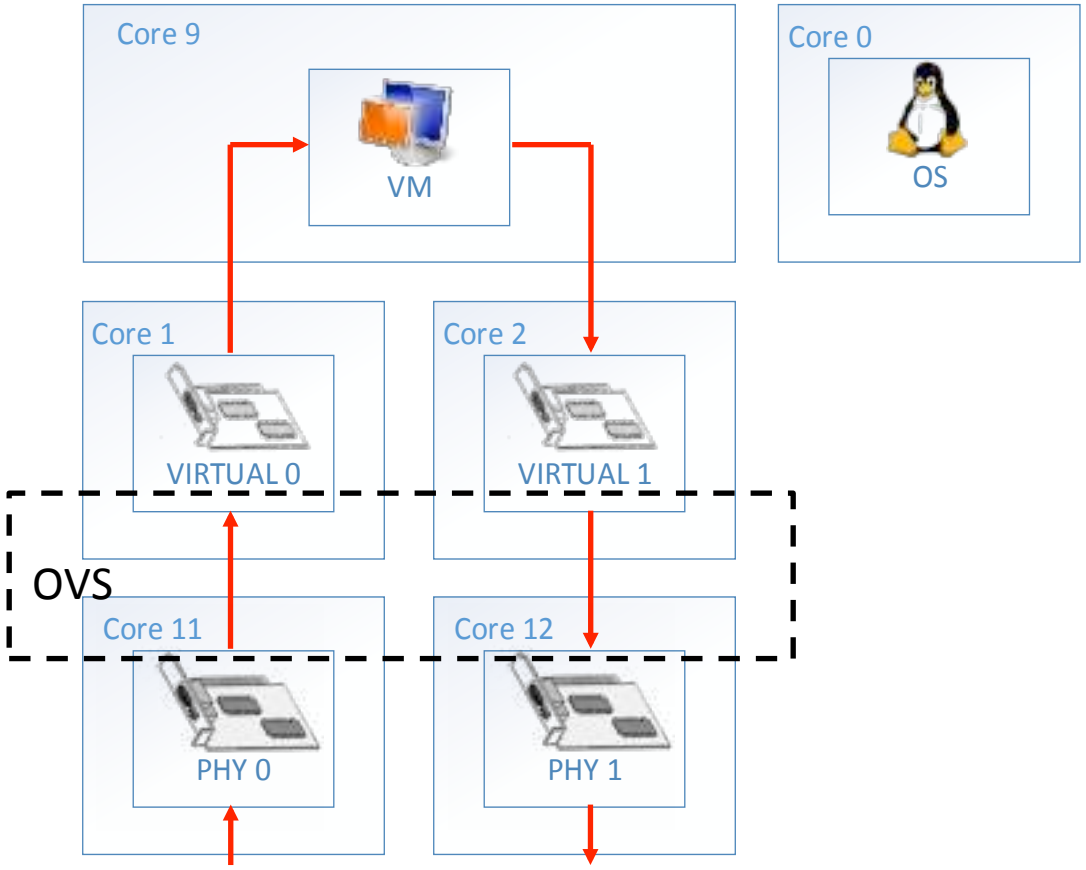
DPDK

Performance

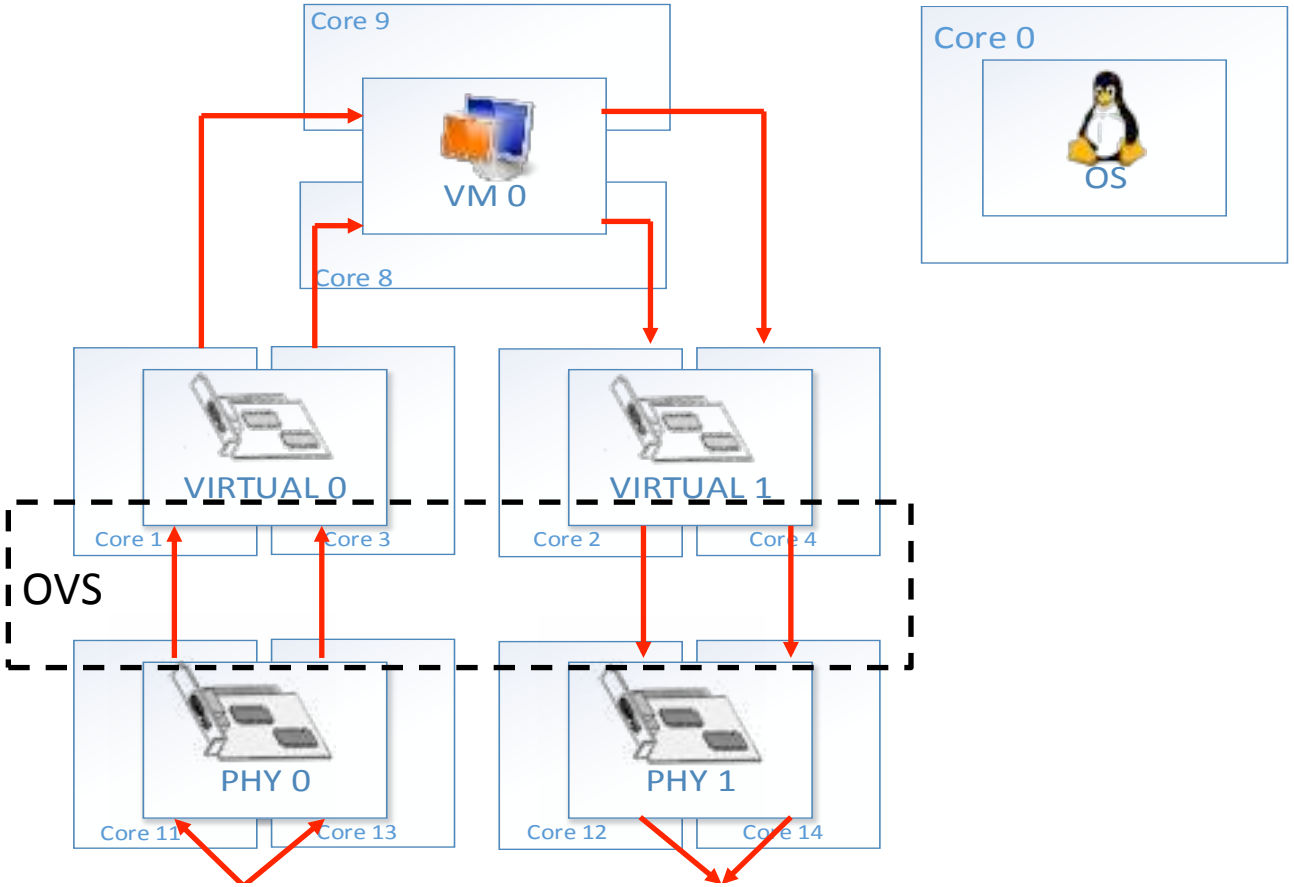
Tuning - Multiple Threads



Tuning - Multiple Threads



Tuning - Multiple Queues

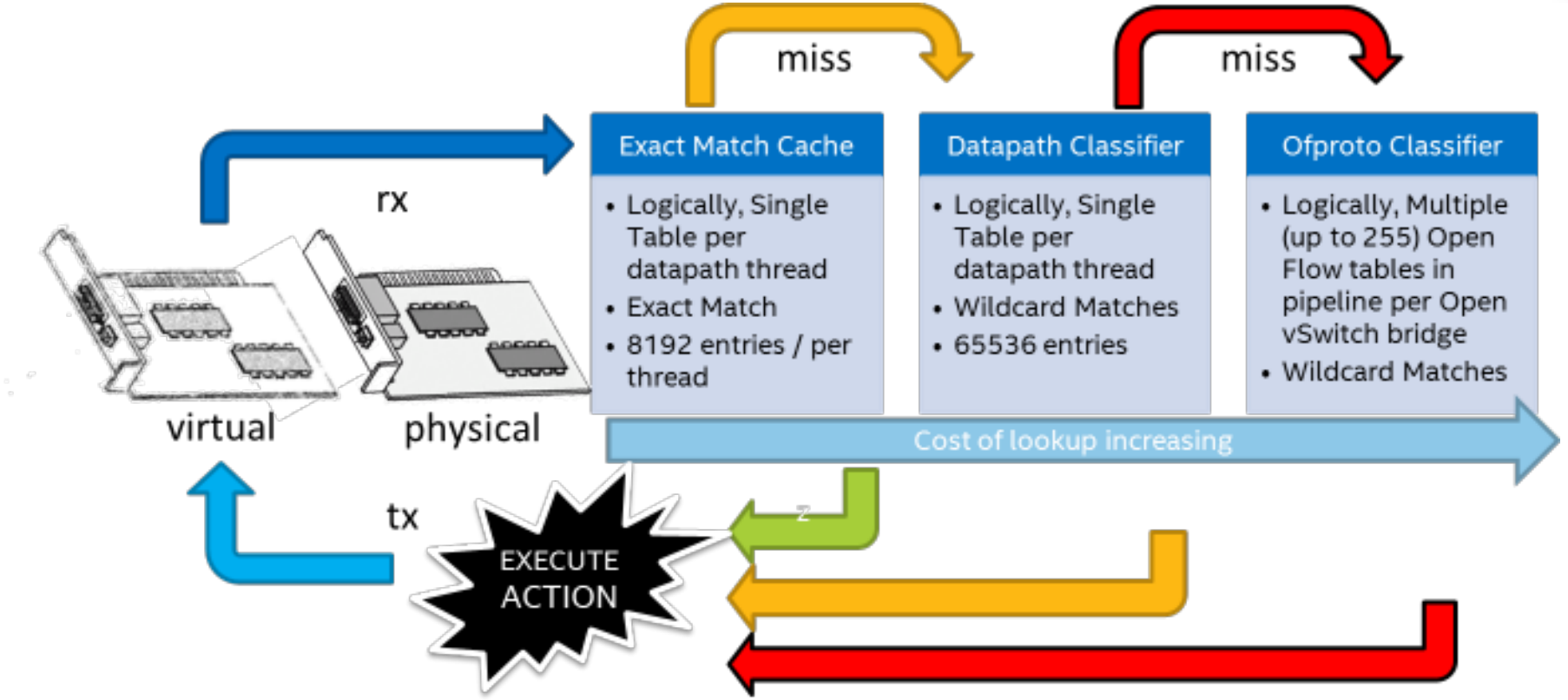


DPDK

Performance Optimizations

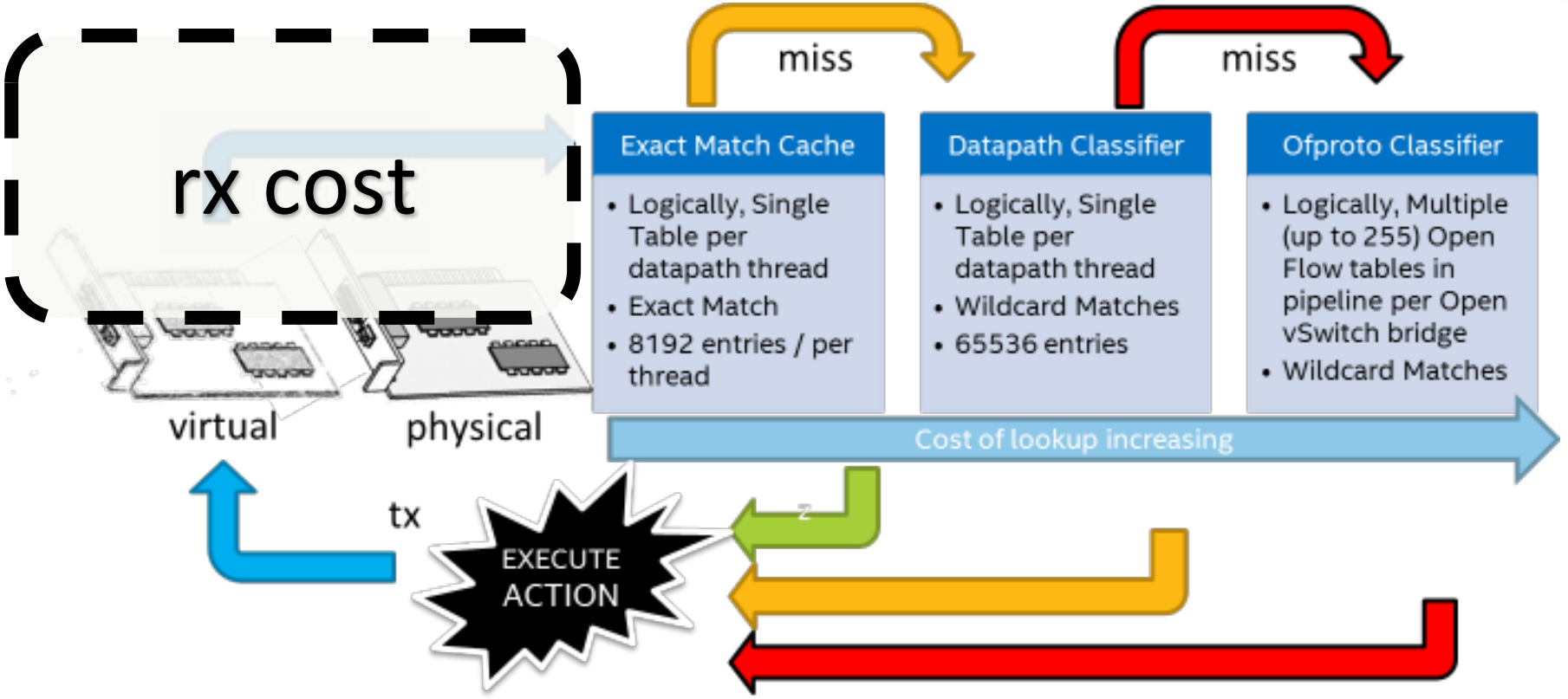
Performance Optimizations

DPDK Userspace 2015



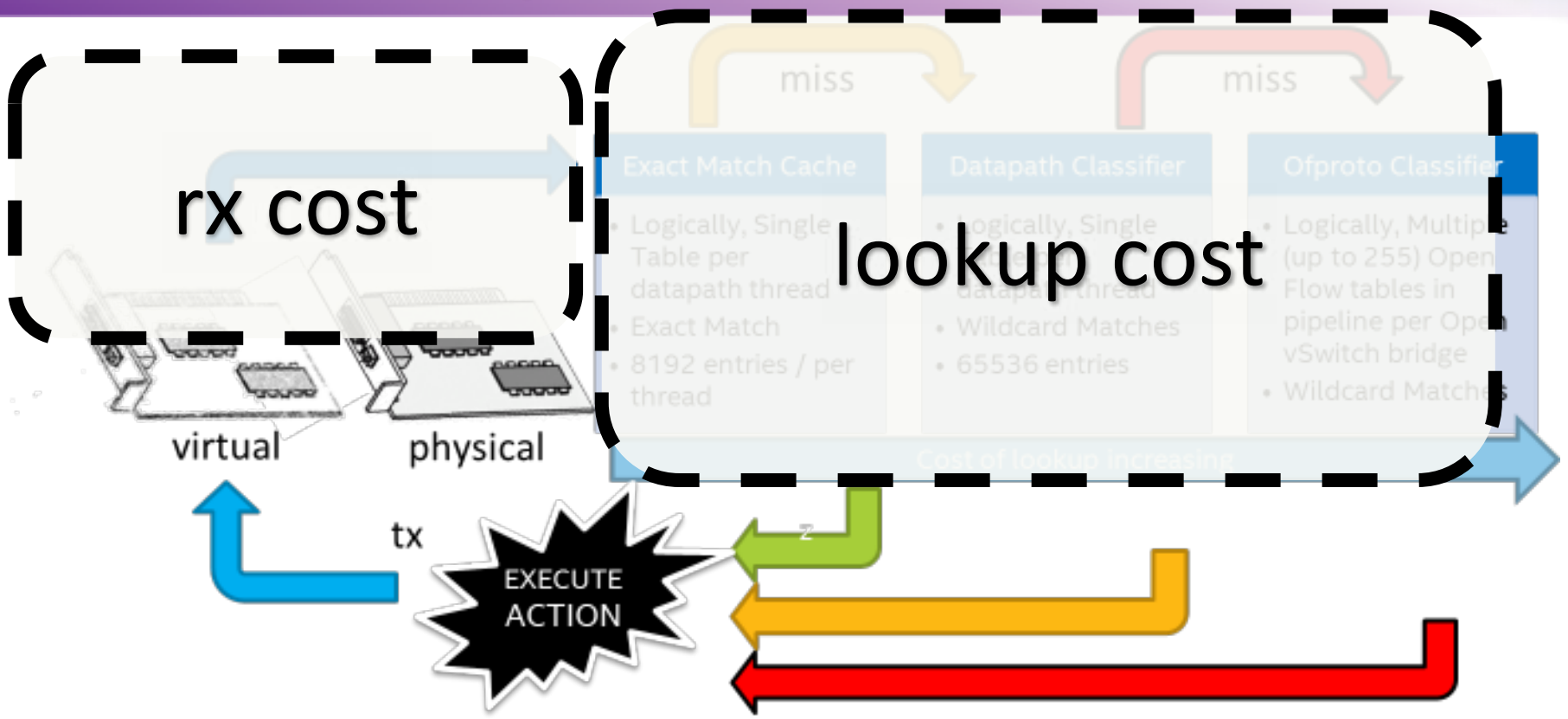
Performance Optimizations

DPDK
Userspace 2015



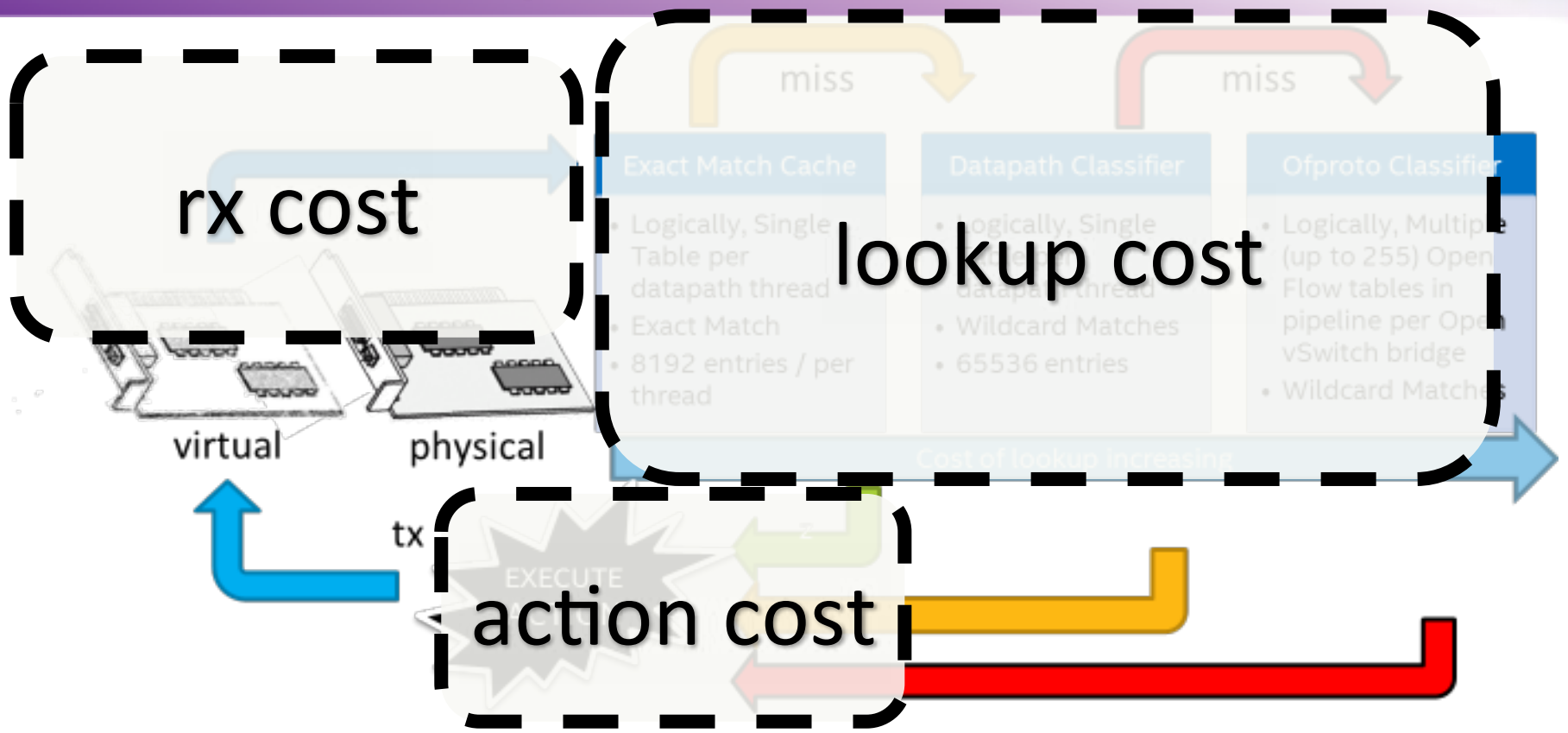
Performance Optimizations

DPDK
Userspace 2015



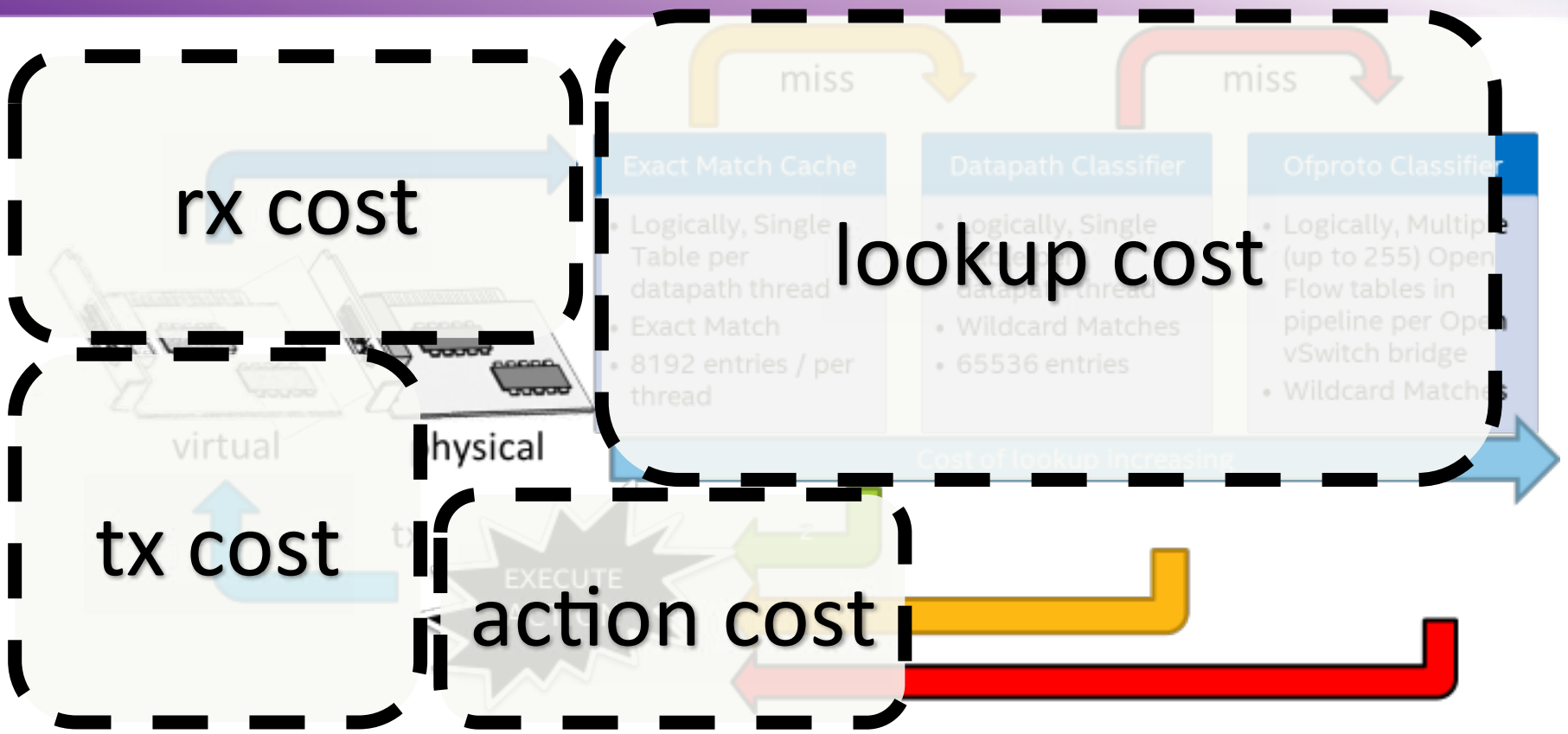
Performance Optimizations

DPDK
Userspace 2015



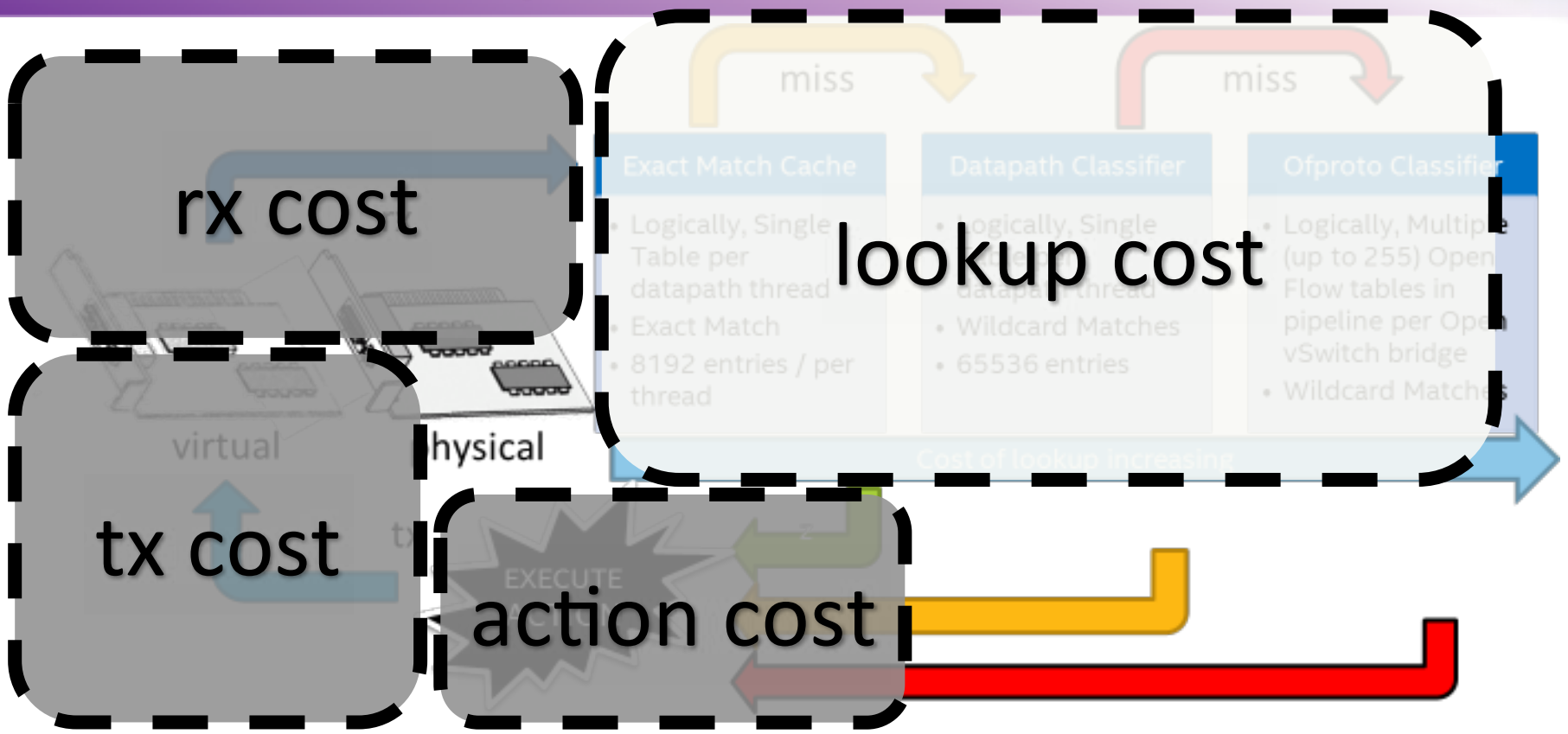
Performance Optimizations

DPDK
Userspace 2015



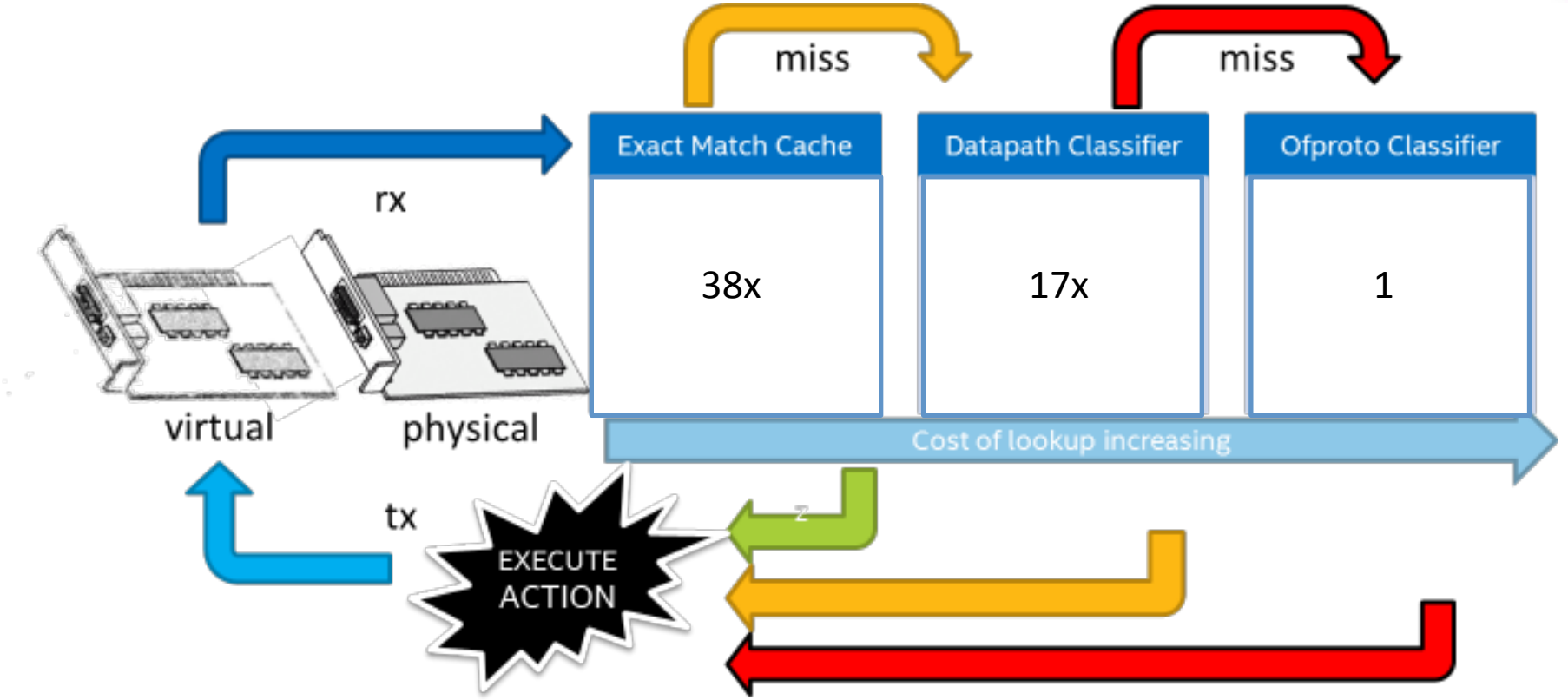
Performance Optimizations

DPDK
Userspace 2015



Performance Optimizations

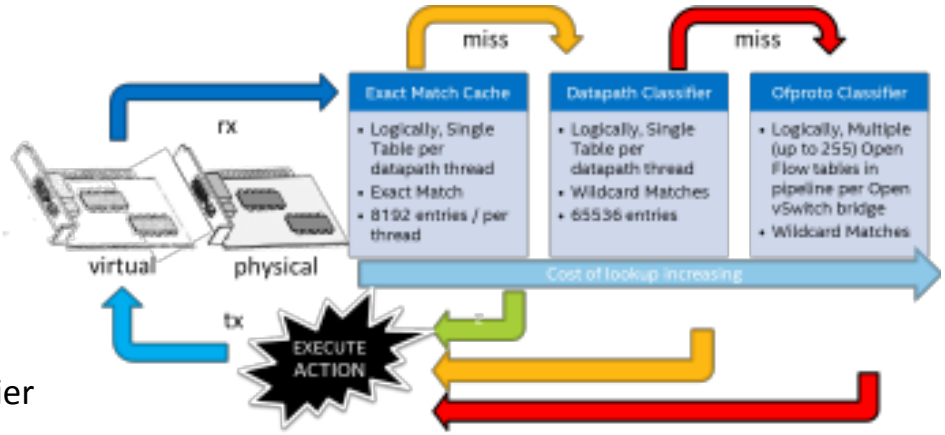
DPDK
Userspace 2015



Performance Optimizations

DPDK Userspace 2015

- Physical
  - Offload hash to NIC hardware
  - Multiple Rx Queues using RSS
  - DPDK Tx/Rx use of Intel's Advanced Vector Extensions
  - DPDK Bulk allocation and batching
- Exact Match Cache
  - Increased size
  - DPDK hash integration - needs variable key size
  - Native EMC optimization
- Datapath Classifier
  - Investigating use of DPDK ACL table
  - Different usage model than OVS datapath classifier
- Virtual
  - vhost library – prefetching and bulk operations
  - virtio pmd vectorization
  - Multiqueue vhost

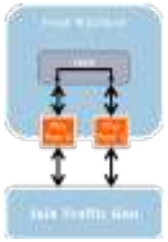


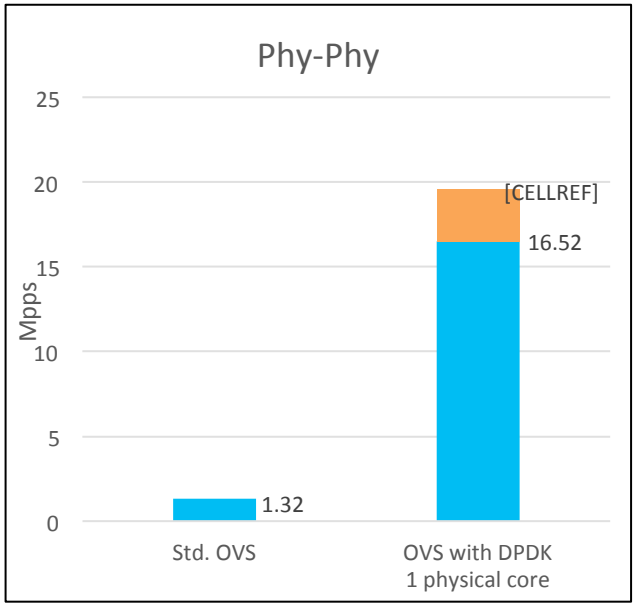
DPDK

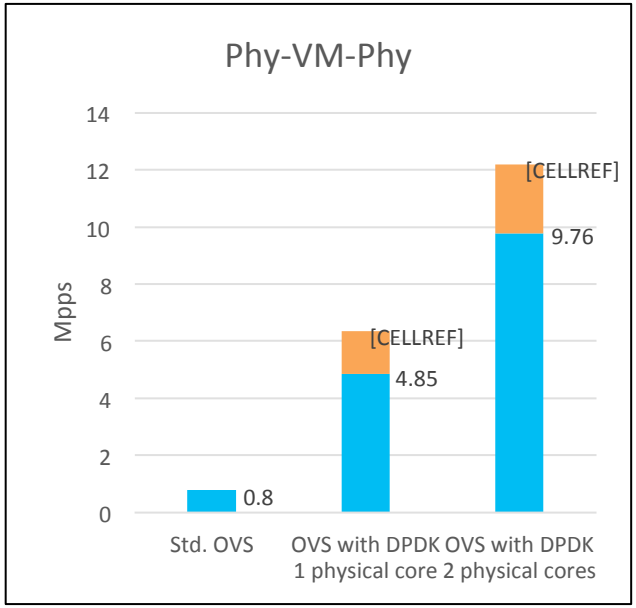
Performance Results

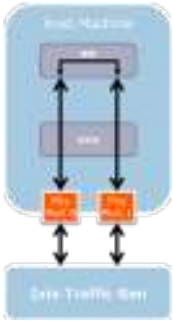
Performance Results

DPDK Userspace 2015







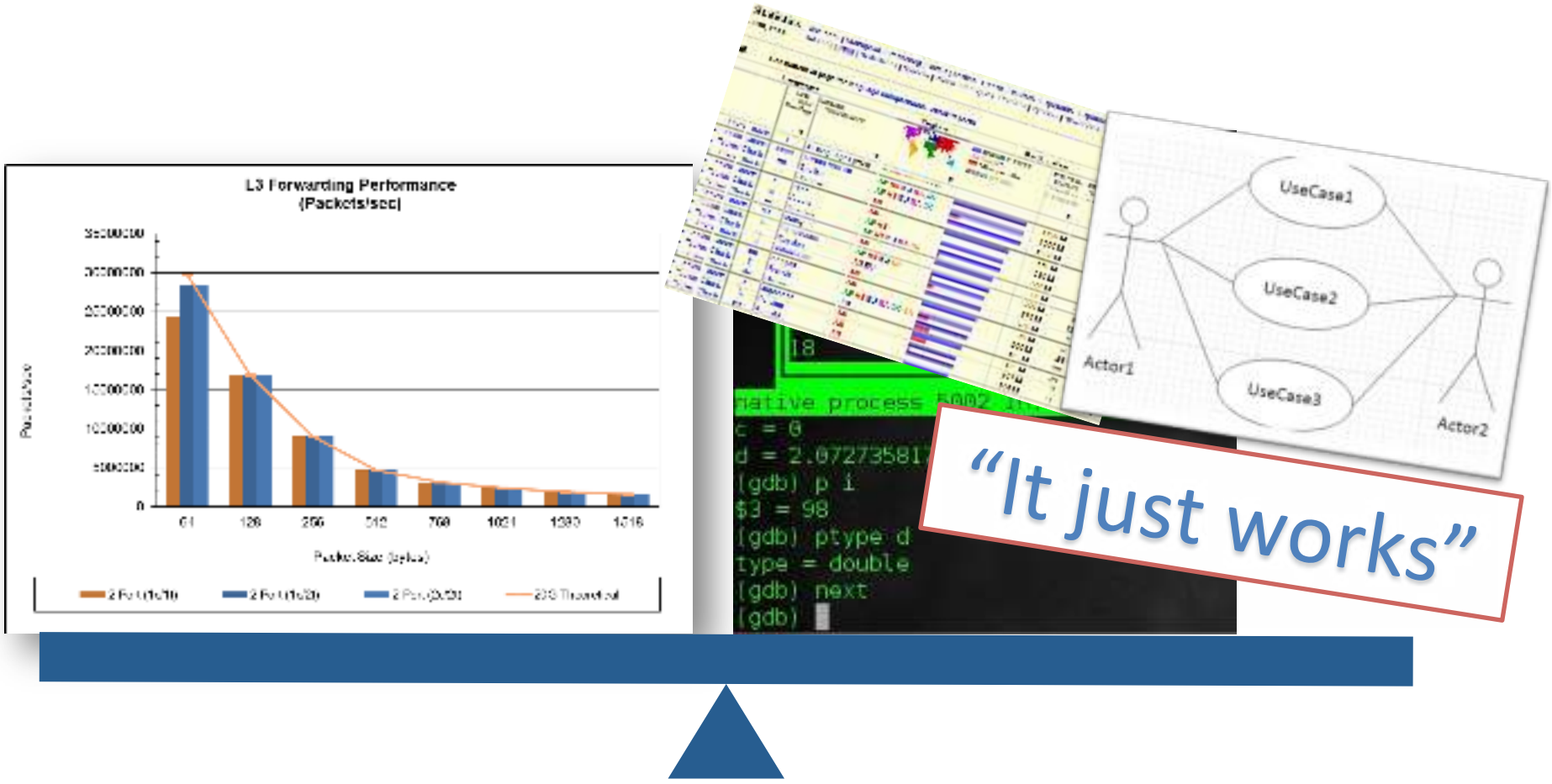


E5-2680v2 2.8GHz core - 64 byte packets - OVS commit id f82313 - DPDK 2.0

DPDK

Conclusion

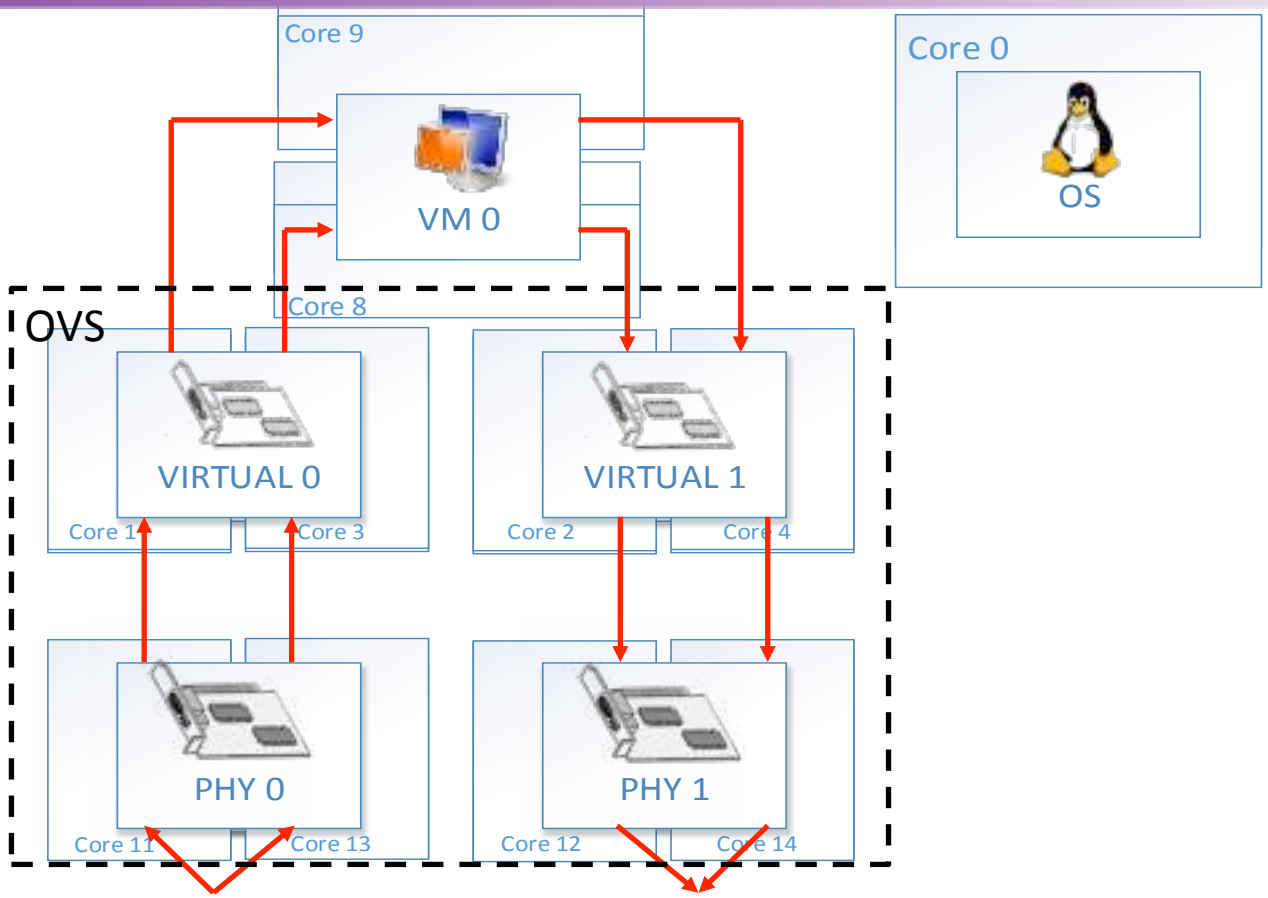

L3 Forwarding Performance
(Packets/sec)
UseCase1
UseCase2
UseCase3
Actor1
Actor2
"It just works"


L3 Forwarding Performance
(Packets/sec)
UseCase1
UseCase2
UseCase3
Actor1
Actor2
(gdb) p i
$3 = 98
(gdb) ptype d
type = double
(gdb) next
"It just works"

DPDK

Backup

Tuning - Multiple Queues



OVS Architectural Evolution

DPDK Userspace 2015