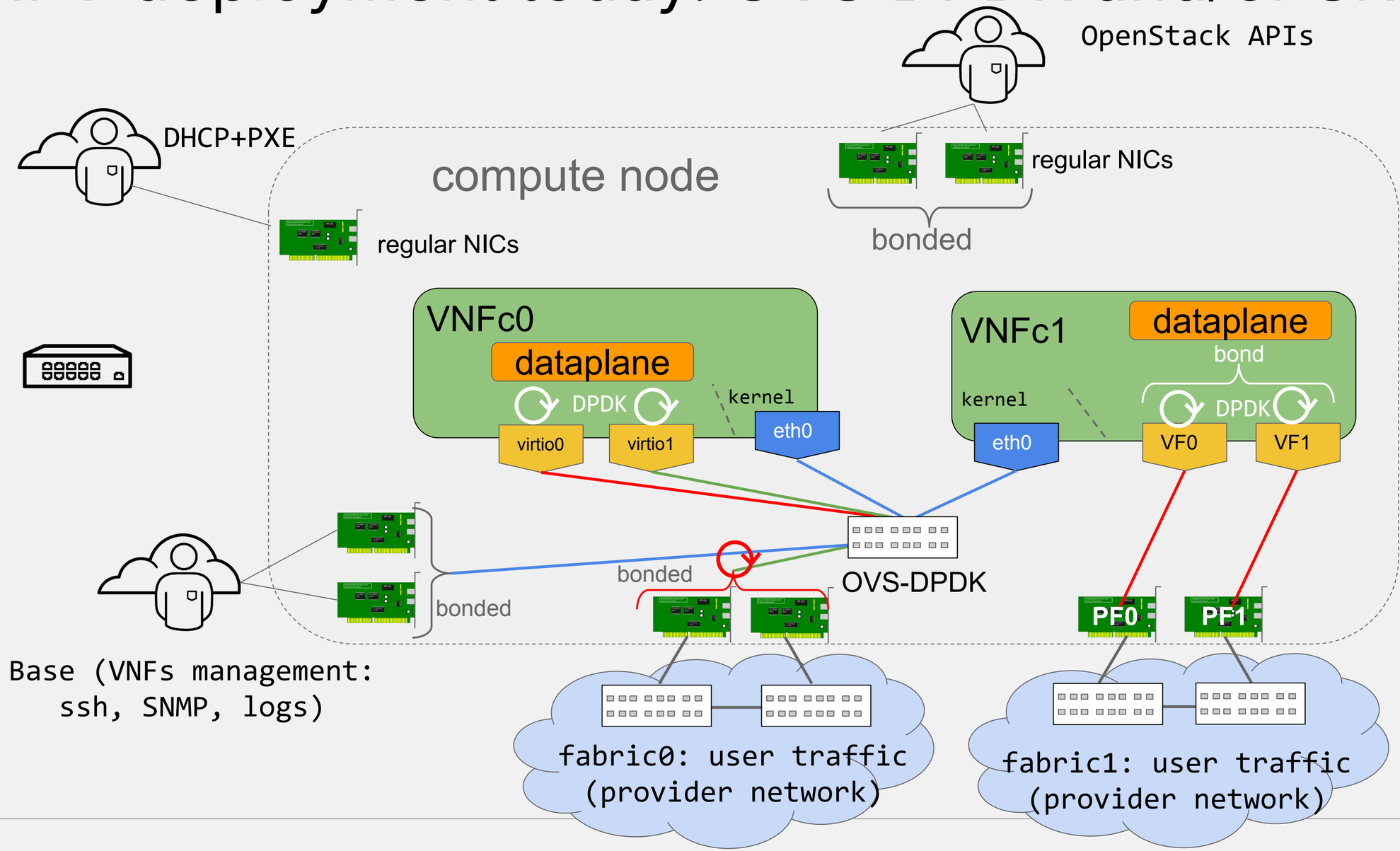
OVS-DPDK for NFV: go live feedback!

Franck Baudin, Principal Product Manager - OpenStack NFV
Anita Tragler, Product Manager - Networking/NFV Platform

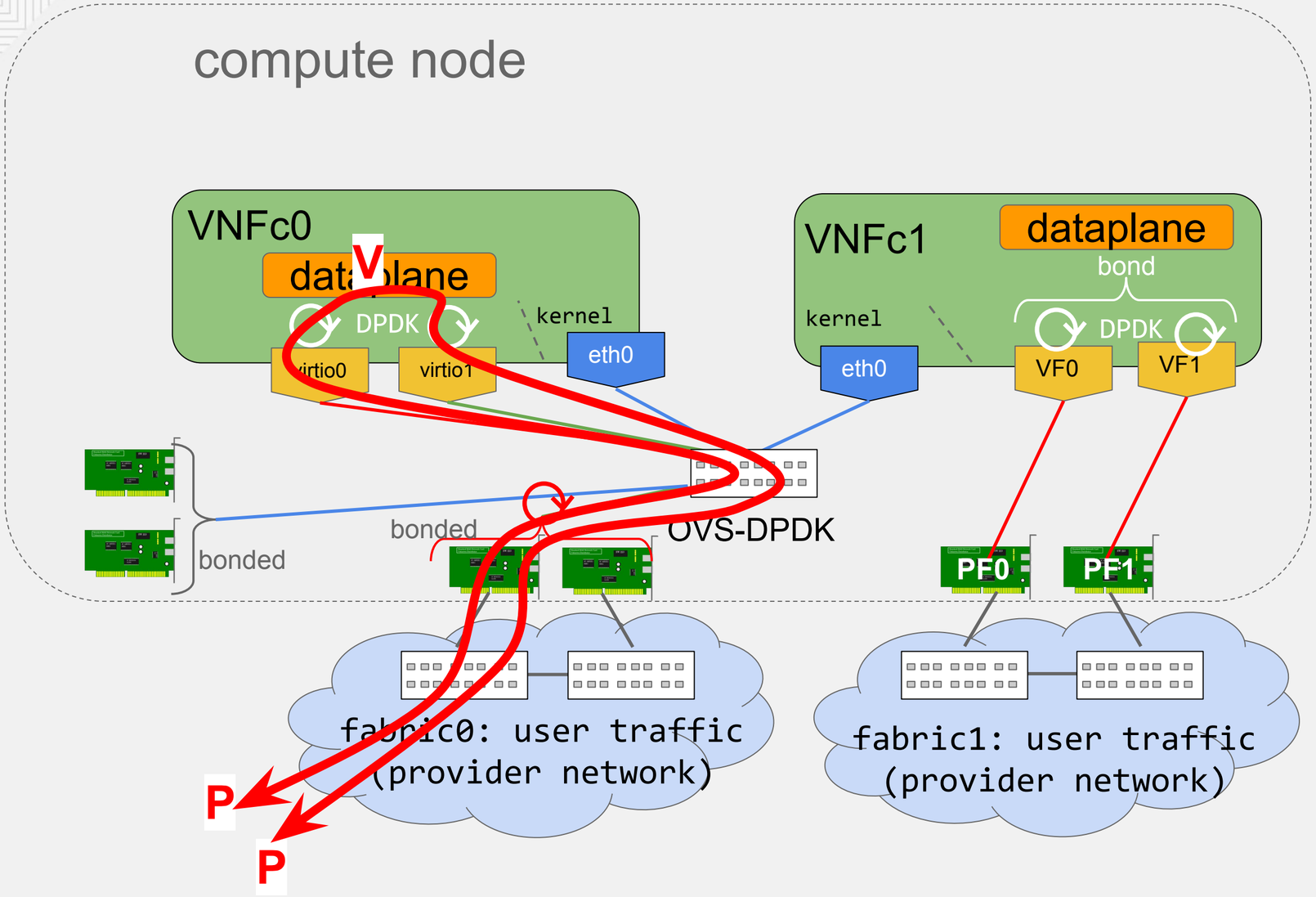
November, 2017 - OVS Conference

OVS-DPDK typical NFV deployment

NFV deployment today: OVS-DPDK and/or SR-IOV



Today’s subscriber traffic pattern: PVP*



10 Gbps chunk of “real Mobile traffic”**

Average frame size: 600 Bytes

Throughput : 4 Mpps

1 M of established flows***

200 k/s new flows

200 k/s destroyed flows

** Physical - Virtual - Physical*

***Numbers for a 10 Gbps chunk:*

multiply the numbers by 2.5 for 25 Gbps

divide by 10 for 1Gbps

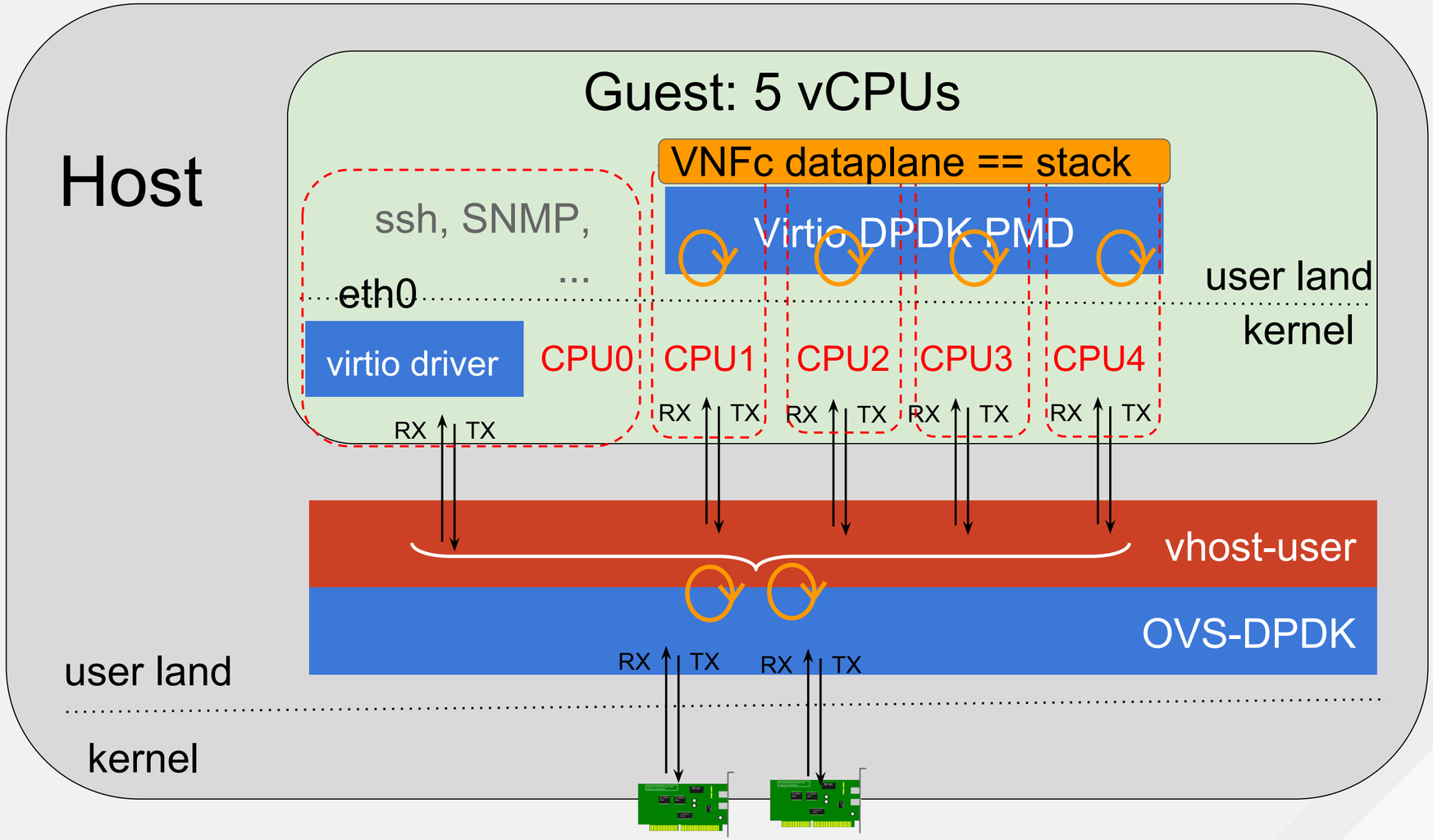
**** ~contrack, bi-directional 5-tuples*

OVS-DPDK: virtio, vhost-user, virtio PMD

DPDK
DATA PLANE DEVELOPMENT KIT

ACTIVE LOOP

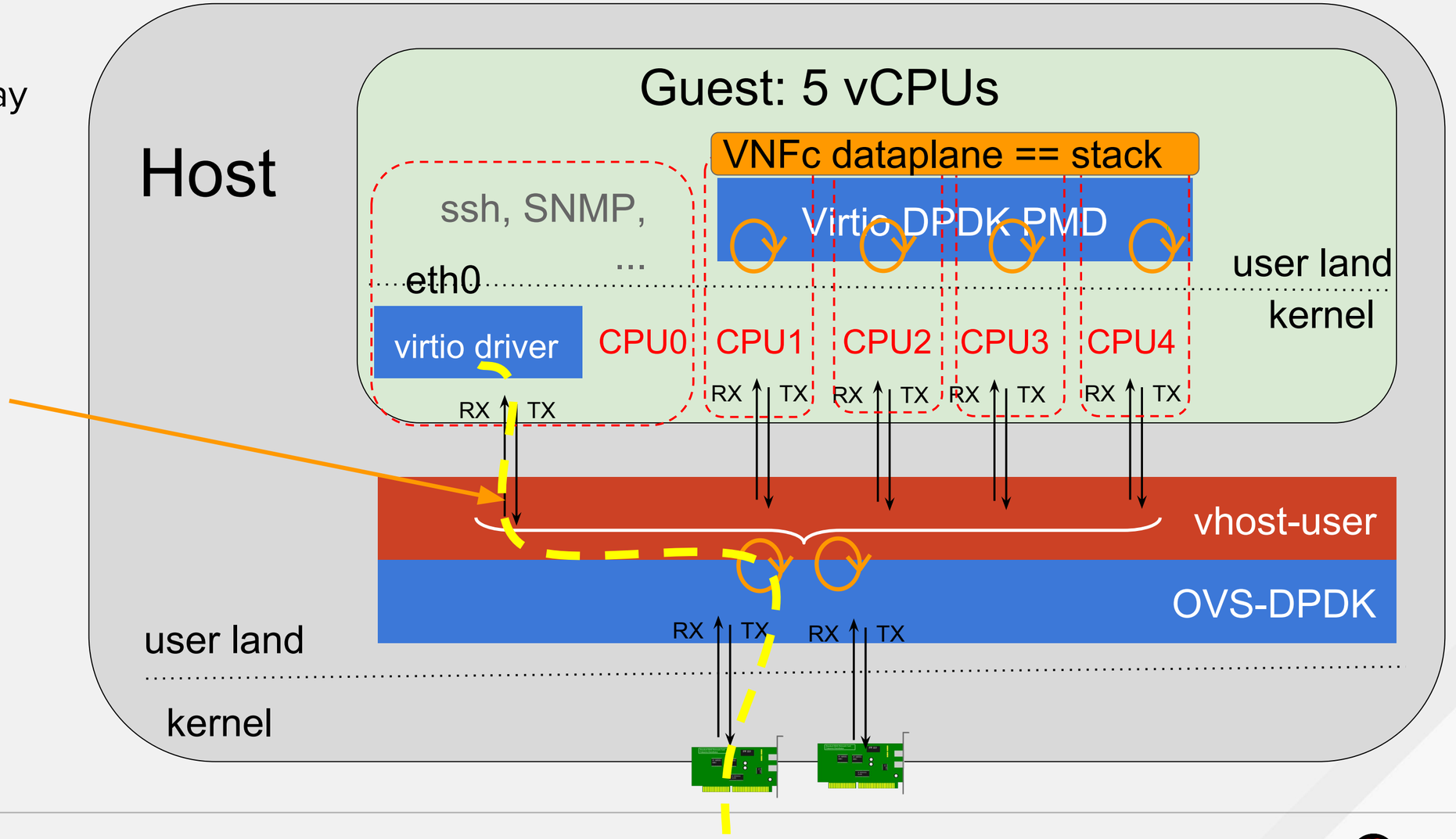
```
while (1) {
    RX-packet()
    forward-packet()
}
```



redhat.

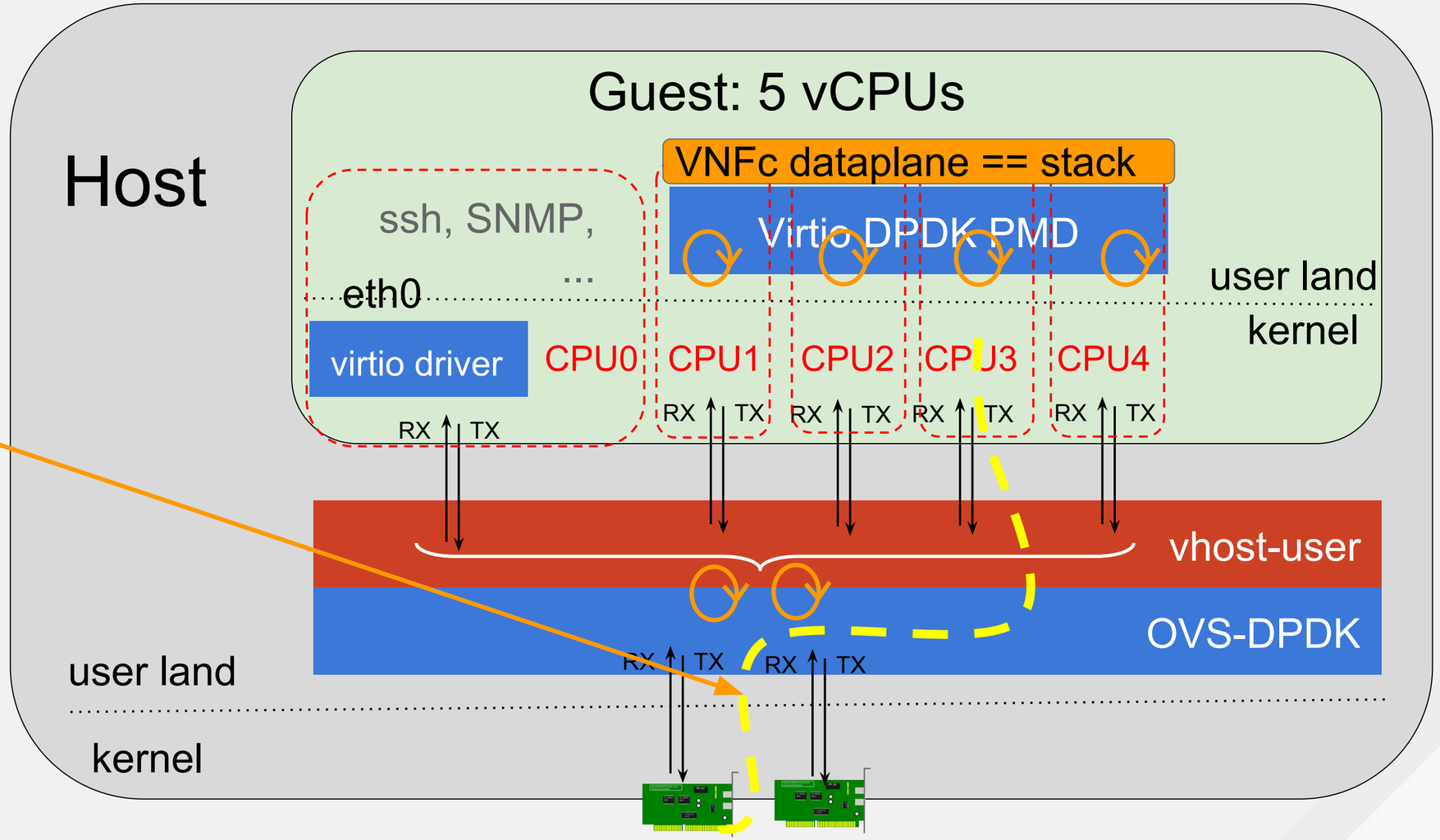
OVS-DPDK: VM management interface

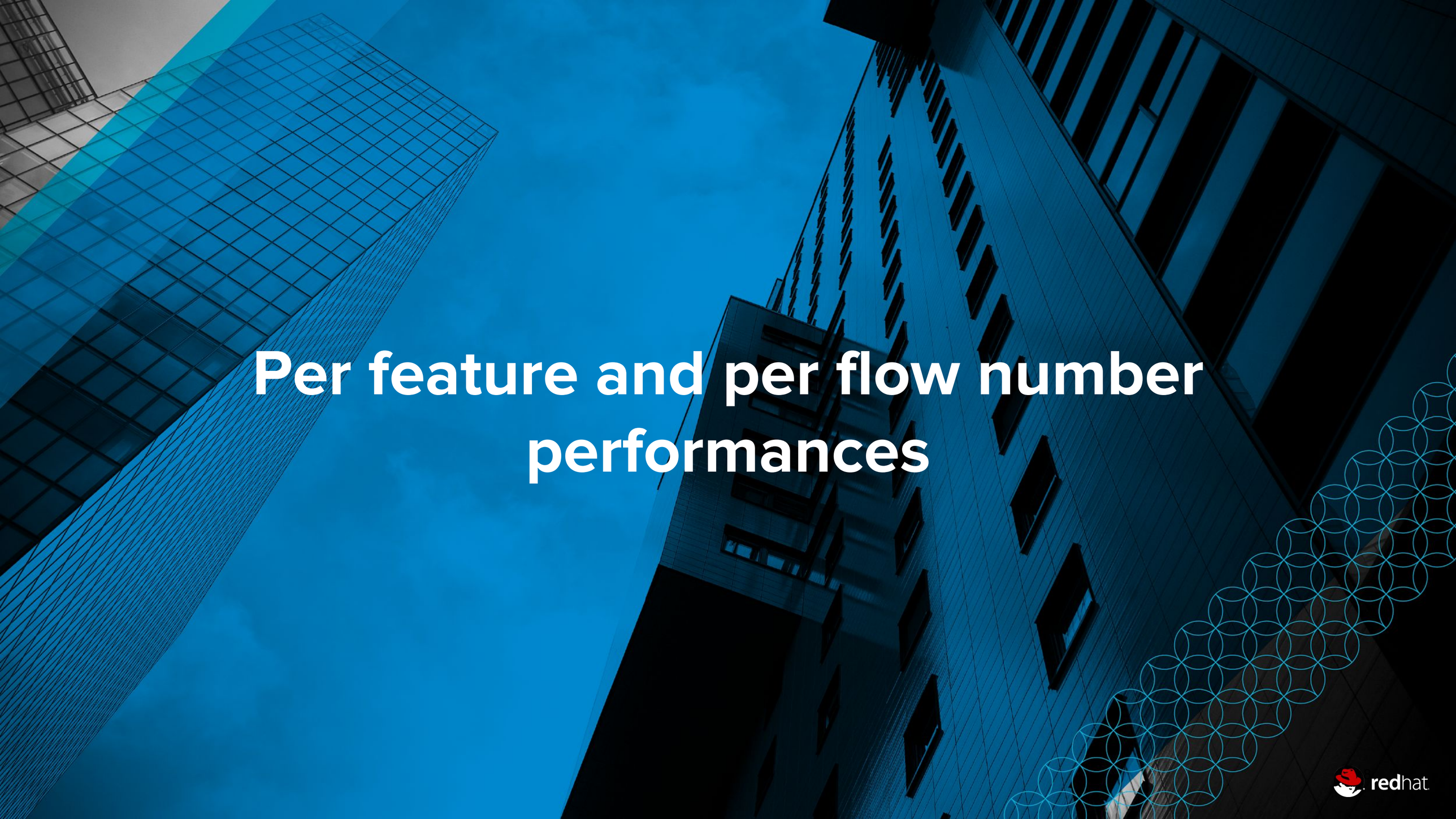
- Usually VLAN underlay
  - Sometime VxLAN
- Security groups “on”
  - Stateless
  - Stateful/conntrack
- Usually low traffic
- Dedicated NICs
  - Don’t mix with subscribers traffic



OVS-DPDK: dataplane interfaces

- Usually VLAN underlay
  - MPLS often requested
  - Few requests for VxLAN
- Security groups “off”
  - Use a VNF firewall
  - Few requests to add some ACLs on VMs (not user traffic but on VM to VM)
- Bonded NICs
  - Often with LACP
  - Often 9K MTU



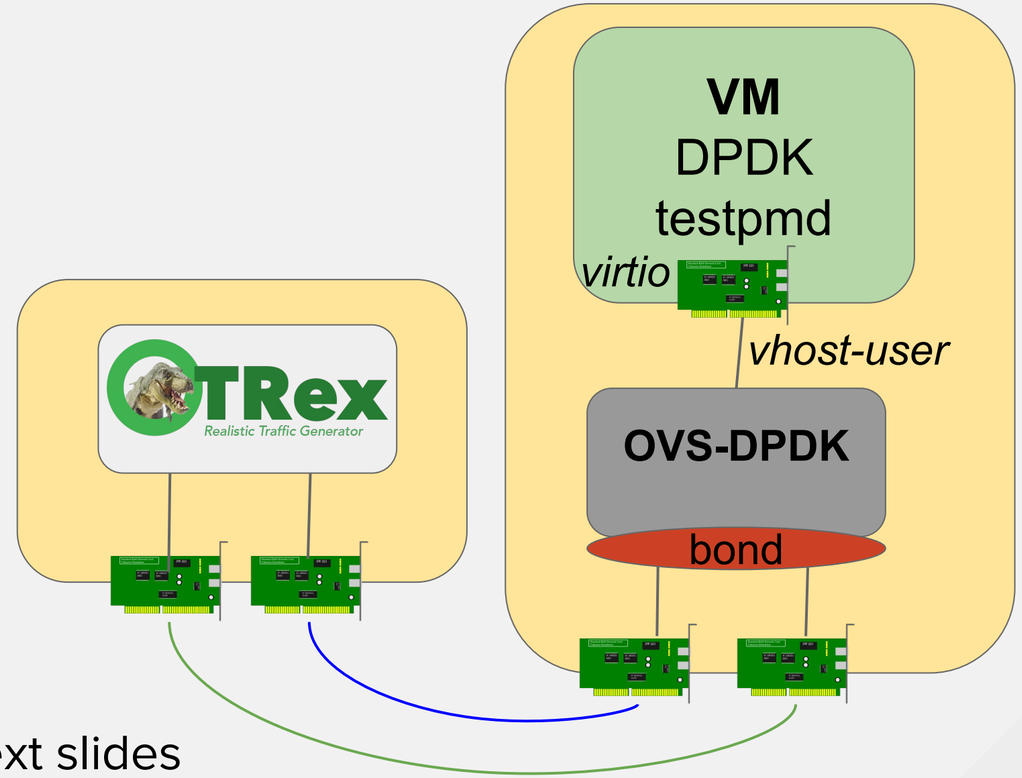

Per feature and per flow number performances

Measurement methodology overview

All tests developed within OPNFV VSperf project

All tests (next slides) done with:

- OVS-DPDK - OVS 2.7 (DPDK 16.11)
- IPv4 traffic
- Same NUMA (VM, DPDK PMDs and NIC)
- RFC2544, 0% acceptable loss rate, 2 mins iterations
- UDP flows, 5 Tuple match, referred as “flows” in the next slides
- DPDK testpmd in the VM, so the VM is never the bottleneck (verified)



EMC (Exact Match Cache) performances impact?

Exact Match Cache

Tests matrix

- EMC enabled default value (=100) and disabled
- For 1k, 10k, 100k flows
- OpenFlow pipeline
  - Baseline - port cross-connection (no CT)
  - Conntrack w/ various matches (up to 5 tuple)

OVS-DPDK sees no significant difference in performance with or without EMC

1M flows to be tested soon...

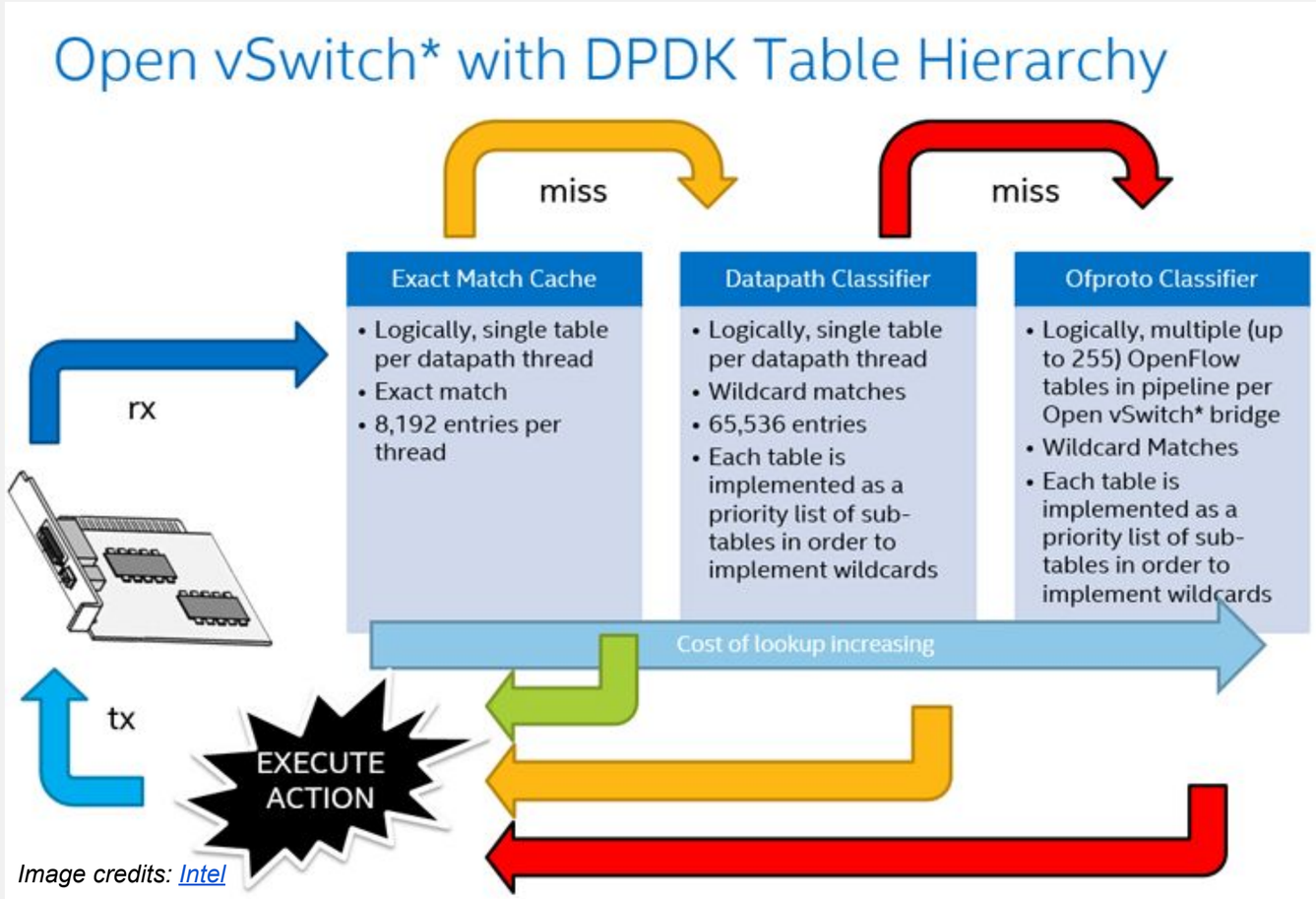


Image credits: Intel

Flow (subscriber traffic) count impact

With and without **stateless** firewall

1k and 10k flows: baseline performances are the same

100k flows: 50% degradation in baseline

1M flows measurements to come

LLC/cache consumption by OVS-DPDK PMD increases with flow count

Individual features impact

For reference, absolute numbers measured with 4-PMDs/4-Hyperthreads/2Cores/1-NUMA-node

From Baseline 7 Mpps without firewall (conntrack) and 1k flows..

- This is with same NUMA for VNF, PMD threads and DPDK NIC
  - Cross-numa shows ~50% performance reduction
- This is without tunneling, no VxLAN
  - VXLAN encapsulation adds 30% performance hit
- This is without conntrack
- This is without QoS
- This is without LACP bonding
- This is with a friendly VM, not competing for LLC/RAM

Performance Drops to 1.6 Mpps (80% drop) with firewall (CT) and 100k flows.

Other performances aspect

LACP with balance-tcp, active/backup

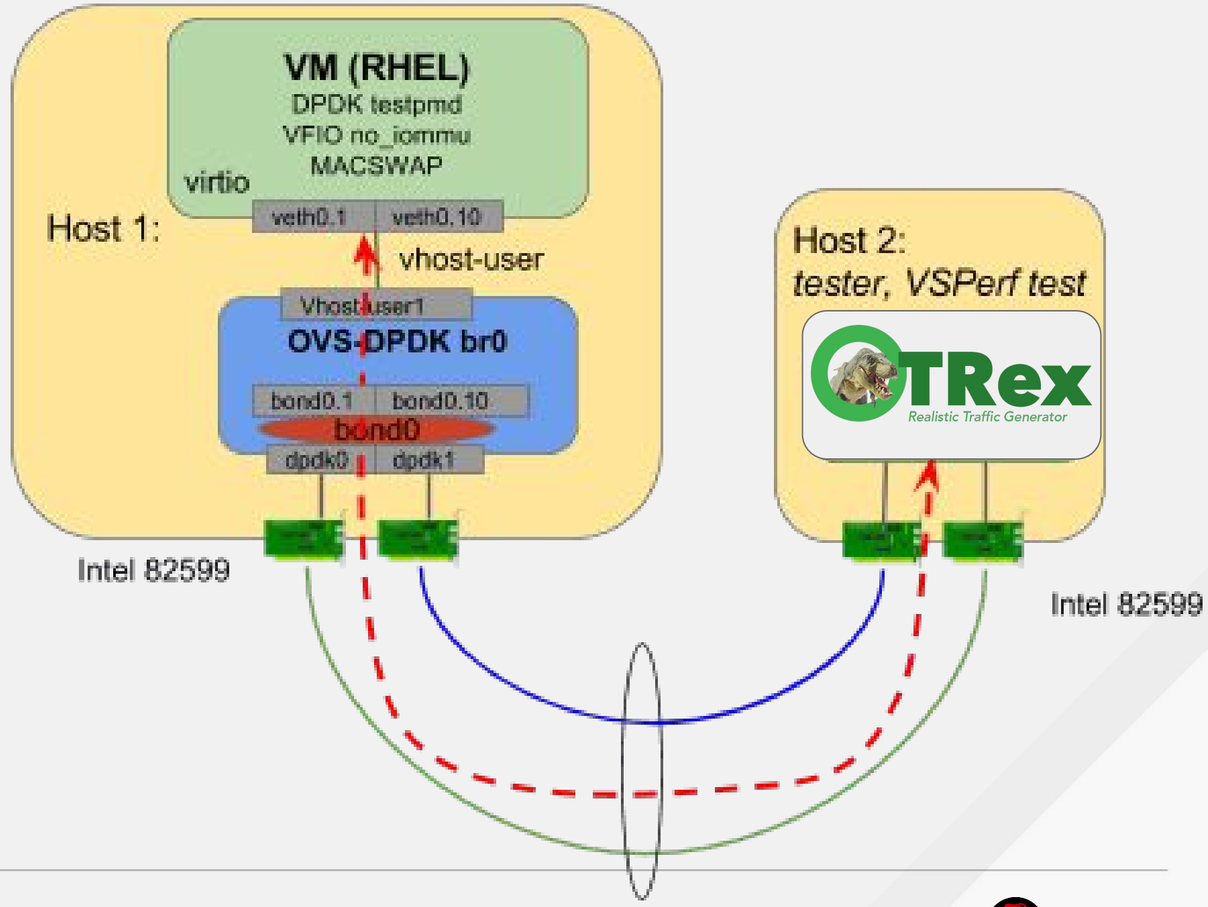
Also part of OPNFV VSPerf suite

Failover time measurement at various pps, 1k flows

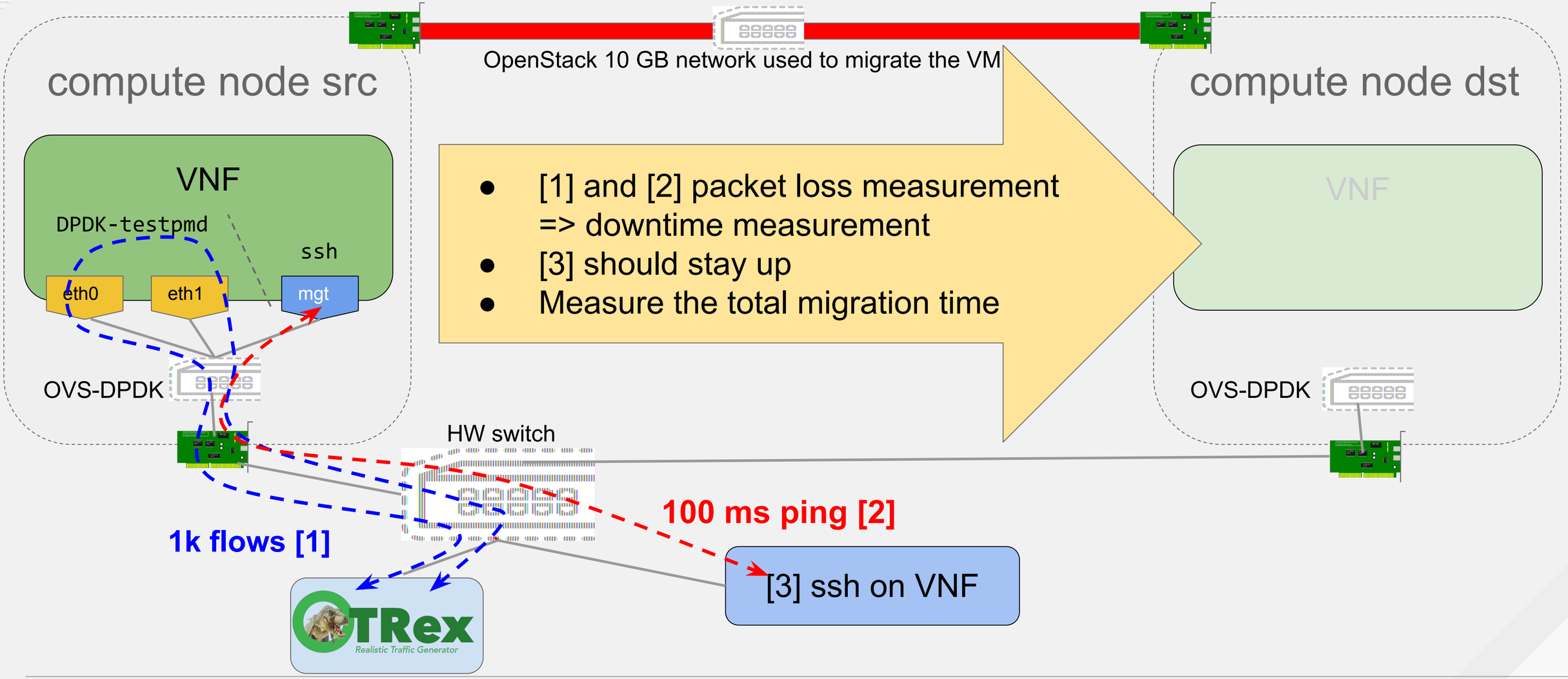
1000 pps: 312ms (312 packets dropped)

100 kpps: 376ms (37595 packets dropped)

1 Mpps: 449ms (448642 packets dropped)



OVS-DPDK Live Migration 1/2



- [1] and [2] packet loss measurement
  => downtime measurement
- [3] should stay up
- Measure the total migration time

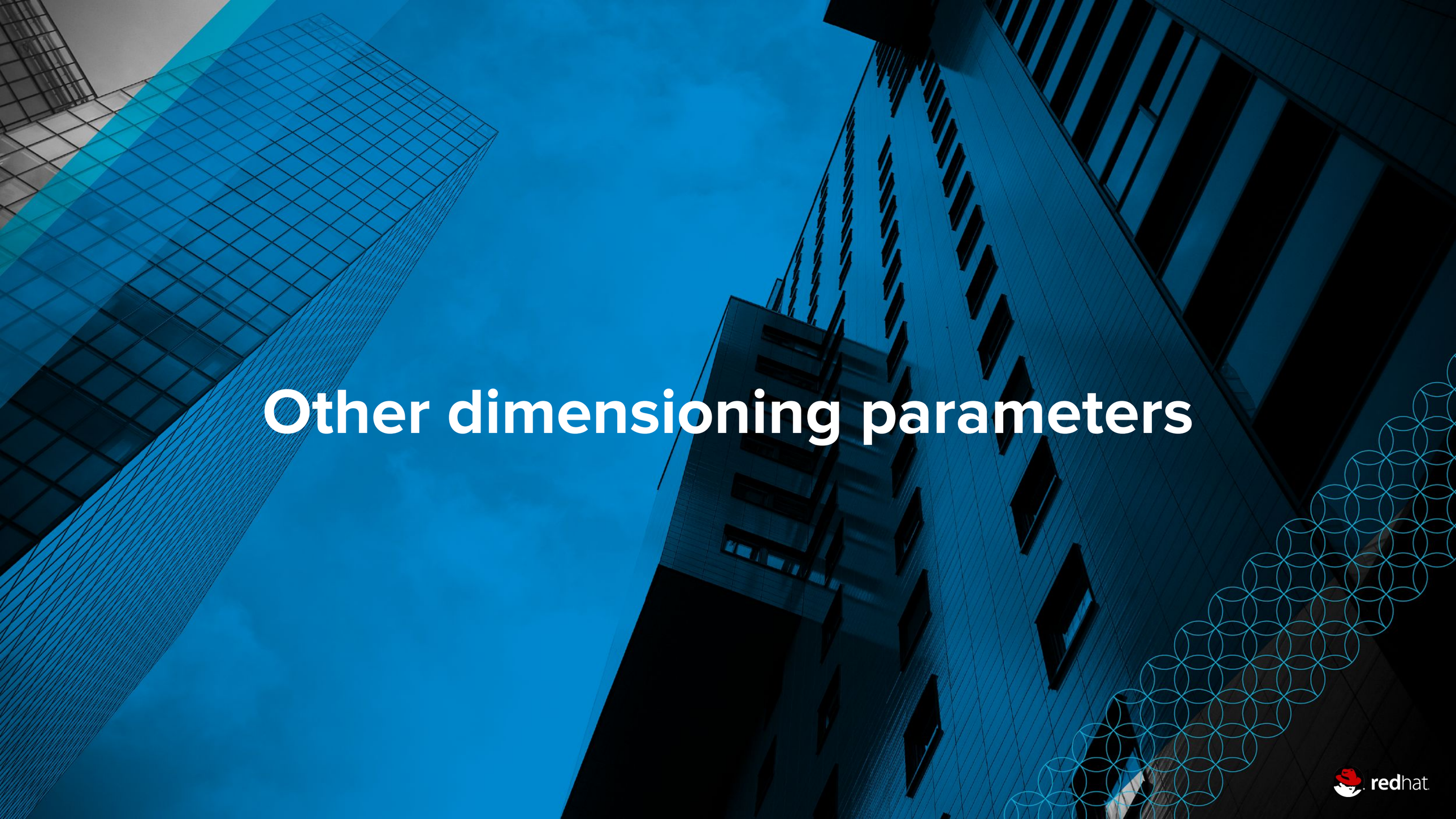
OVS-DPDK Live Migration 2/2

Test parameters

- 1 to 5 Mpps subscriber traffic [1]
- OpenFlow pipeline based on NORMAL
  - MAC learning
  - Gratuitous ARP
- 8 GB guest, 2M and 1G huge pages
- Friendly/optimistic parameters
  - testpmd use a single 1GB and few 2M huge pages: a realistic VNF would trash/use way more. Proposal to add such behavior to testpmd posted on DPDK mailing list.
  - No security groups (conntrack needs to migrate as well?), no QoS, no VxLAN, ...
  - 1k flows

Migration time: between 12s and 17s

Service downtime (dropped subscriber traffic): between 100ms and 150ms


Other dimensioning parameters

OVS-DPDK Host/VNFs guests resources partitioning

Typical 18 cores per node dual socket compute node (E5-2599 v3)

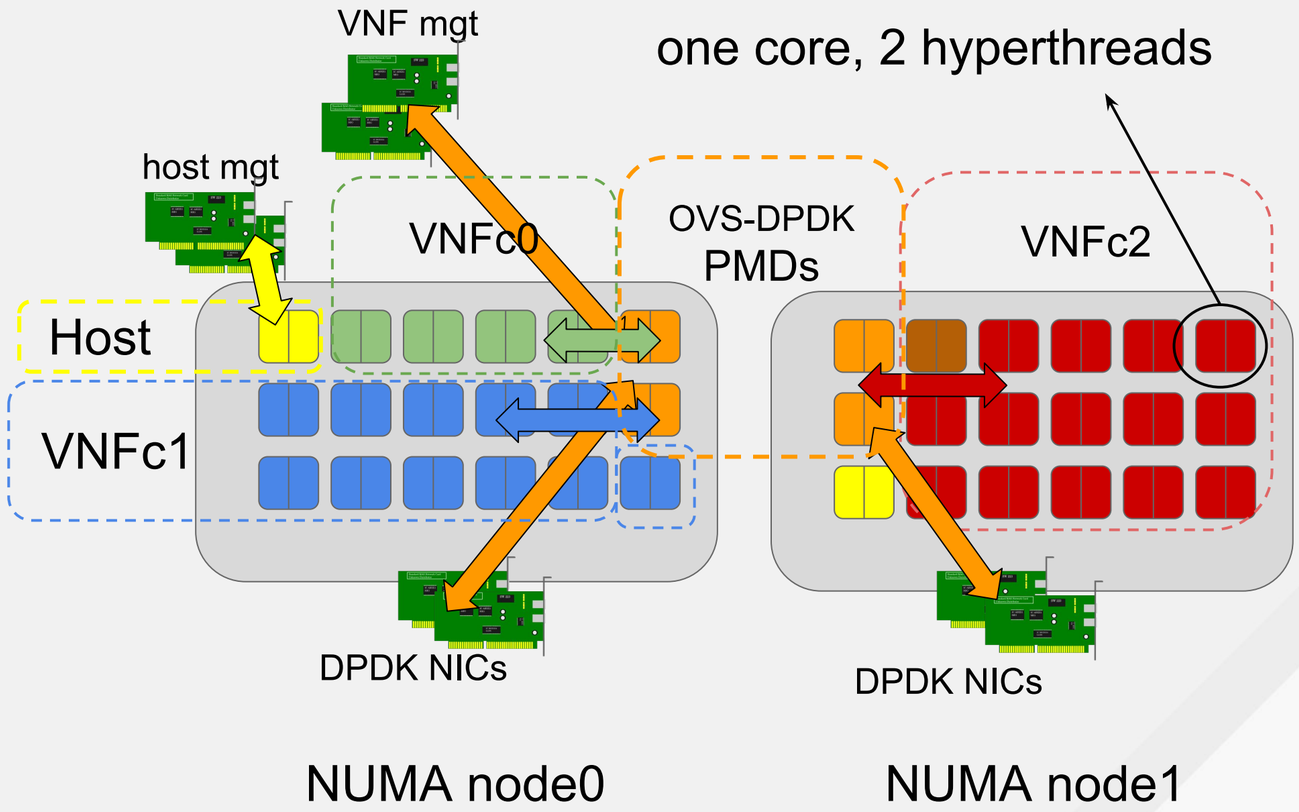
All host IRQs routed on host cores: the first core of each NUMA node will receive IRQs, per HW design

All VNFx cores dedicated to VNFx

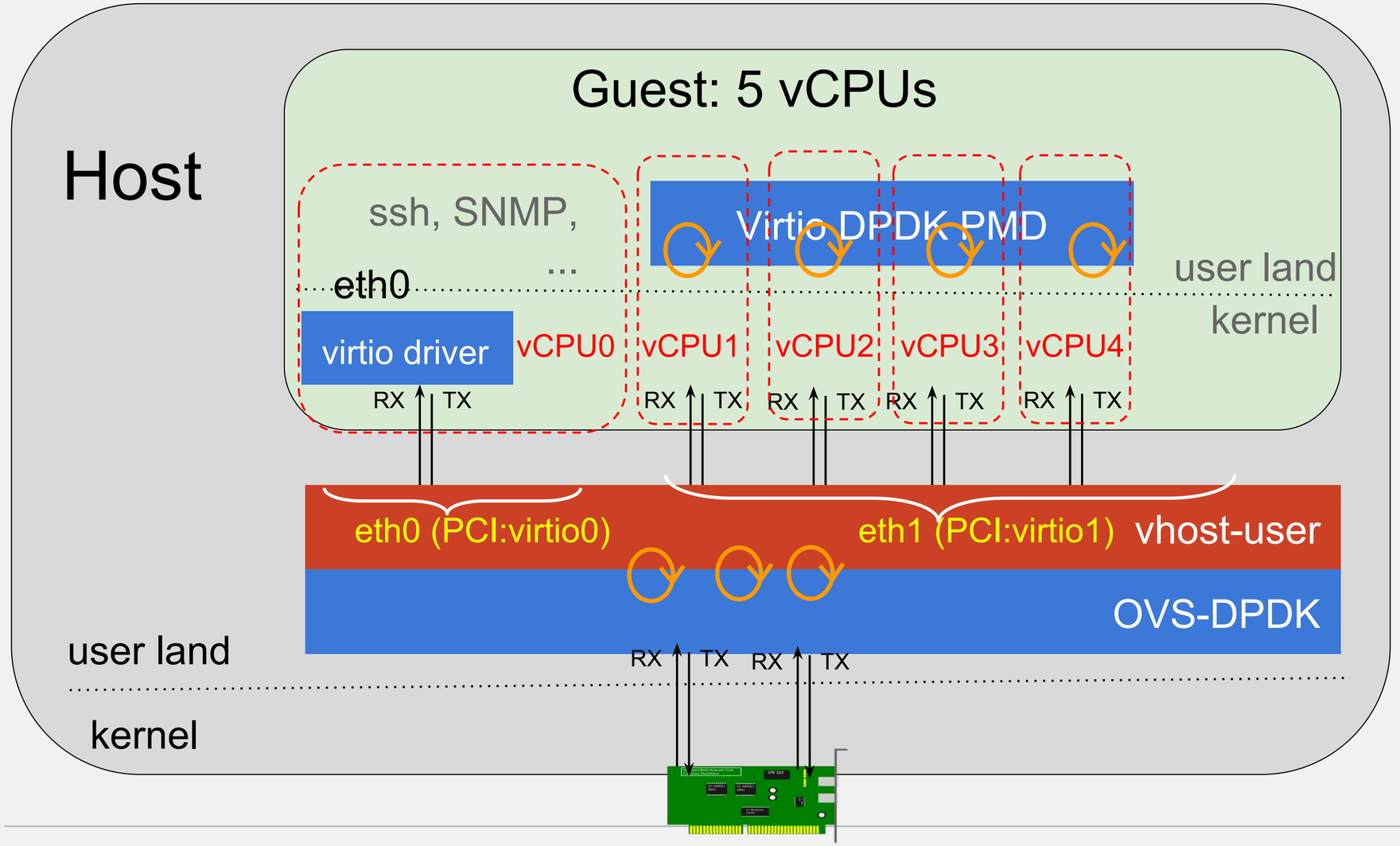
- Isolation from others VNFs
- Isolation from the host

ovs-vswitchd parameters

- PMDs threads: per user configuration
- dispatcher/revalidator: as many as host CPUs
- Hugepages number: depends on MTUs, NICs numbers, queues number … formula under documentation… start with 4GB per NUMA, look at ovs-vswitch logs, and don’t hesitate to double!



Challenge: balance the queues among PMD threads



Queues are distributed in a round-robin fashion among the PMDs of the proper NUMA node

=> Could work if all queues were equally loaded... but some queues can even not be used by the guest!

=> Could work if the queues number is way greater than the PMD numbers... but rarely the case

commit cd995c739a330dbcaee6433c08ecaad62791a56b

Author: Kevin Traynor <ktraynor@redhat.com>

Date: Fri Aug 25 00:54:26 2017 -0700

Future: auto-rebalancing

First version landed, testing/experimentation starting!

```
If pmd-rxq-affinity is not set for rxqs, they will be
assigned to pmds (cores) automatically. The processing
cycles that have been stored for each rxq will be used
where known to assign rxqs to pmd based on a round robin
of the sorted rxqs.

For example, in the case where here there are 5 rxqs and
3 cores (e.g. 3,7,8) available, and the measured usage of
core cycles per rxq over the last interval is seen to be:

- Queue #0: 30%
- Queue #1: 80%
- Queue #3: 60%
- Queue #4: 70%
- Queue #5: 10%
```

```
The rxqs will be assigned to cores 3,7,8 in
the following order:

Core 3: Q1 (80%) |
Core 7: Q4 (70%) | Q5 (10%)
core 8: Q3 (60%) | Q0 (30%)

Rxq to pmds assignment takes place whenever
there are configuration changes or can be
triggered by using::

    $ ovs-appctl
dpif-netdev/pmd-rxq-rebalance
```

Final thoughts

With OpenStack Newton OVS-ML2

Without any feature like Security Groups or QoS, with 1k flows

Out of the box: 2Mpps/ NUMA socket

- 1 core (2HT) per NUMA socket

VM NUMA aware tuning: 4Mpps/ NUMA socket

- 1 core (2HT) per NUMA socket, NUMA awareness workaround (not supported by OpenStack yet)

Very advanced tuning: 4Mpps/core scaling with the number of cores

- Requires to properly balance the queues manually, until automated queues rebalancing

OVS-DPDK go live challenges

NFV go-live with OVS-DPDK are taking-off, but they require OVS-DPDK experts support

- More experts needed!!
- Simplification/usability improvement in progress

Per feature performance impact has to be known

- OPNFV VSPerf welcome help!!

All test cases, CI, can be reused as-is for any vSwitch/vRouter, including OVS HW offload

- OPNFV VSPerf welcome any vSwitch/vRouter with or without HW offload

Some cool features still need to be coded, for instance: dynamic and automatic queues-rebalancing without any packet drop (but we can start by measuring the drops…)

redhat.

Thank you!

fbaudin@redhat.com
atragler@redhat.com

plus.google.com/+RedHat

facebook.com/redhatinc

linkedin.com/company/red-hat

twitter.com/RedHatNews

youtube.com/user/RedHatVideos