Benchmarking Virtual Switches in OPNFV

draft-vsperf-bmwg-vswitch-opnfv-00

Maryam Tahhan
Al Morton

Introduction

- Maryam Tahhan
- Network Software Engineer
- Intel Corporation (Shannon Ireland).
- VSPERF project lead

Outline

- What is OPNFV and how can I join?
- What is VSPERF?
- VSPERF test specification approach
- BMWG and IPPM specs referenced by VSPERF and Matrix Coverage
- Test results repeatability considerations
- Flow Classification recommendation.
- Proposed approach for benchmarks using baselines & resource isolation

What is OPNFV?

Open Platform for NFV Project (OPNFV):

- A Linux Foundation open source project focused on accelerating the evolution of Network Functions Virtualization (NFV).
- OPNFV will establish a **carrier-grade, integrated, open source reference platform** for NFV that ensures consistency, performance and interoperability among multiple open source components.
- OPNFV will work with upstream projects to coordinate continuous integration and testing while filling development gaps.

How can I join OPNFV?

- Create a Linux Foundation account that you will use for all the tools provided by the Linux Foundation. You also need this account to contribute to OPNFV projects.
- To participate, via contribution, in any project in OPNFV, you will need to contact the project manager/lead for the project.
- Project Roles: contributor, committer, and project lead.

What is VSWITCHPERF AKA VSPERF?

- An OPNFV Project
- Goal: Characterize the performance of a virtual switch for **Telco** NFV use cases.
- Virtual switches have not typically been designed for Telco NFV use cases that require Telco grade determinism in their performance and support for latency/jitter-sensitive Telco traffic.
- This project proposes defining and executing an appropriate set of tests in order to objectively measure the current Telco characteristics of a virtual switch in the NFVI

OPNFV

VSPERF test specification approach

- Develop a performance test specification for vSwitches (called **Level Test Design or LTD document**).
- LTD should encompass all existing RFCs/Specifications that describe the testing of physical switches.
  - Allows for a fair comparison between physical and virtual network functions.
- LTD should define additional tests applicable to virtual switches, such as: noisy neighbour tests, datapath and control path coupling, CPU and memory utilization…

BMWG *and* IPPM specs referenced by VSPERF

[RFC2544] Benchmarking Methodology for Network Interconnect Devices.

[RFC2889] Benchmarking Methodology for LAN Switching

[RFC6201] Device Reset Characterization

[RFC3393] IP Packet Delay Variation Metric for IPPM

[RFC5481] Packet Delay Variation Applicability Statement

[RFC2285] Benchmarking Terminology for LAN Switching Devices

[RFC1242] Benchmarking Terminology for Network Interconnection Devices.

VSPERF Test Categories

- Throughput tests
- Packet and Frame delay distribution tests.
- Scalability tests.
- Stream performance tests.
- Control path and data path coupling tests.
- CPU and memory utilization tests.

Future tests include:

- Request/response tests.
- Noisy Neighbour tests.

Matrix Coverage of the Current LTD

	SPEED	ACCURACY	RELIABILITY	SCALE
Activation	* Throughput.RFC2889. AddressLearningRate * Throughput.RFC2889. AddressCachingCapacity *PacketLatency.Initial PacketProcessingLatency		* Throughput.RFC2544. SystemRecoveryTime * Throughput.RFC2544. ResetTime	* Throughput.RFC2889. AddressCachingCapacity
Operation	* Throughput.RFC2544. PacketLossRate * Throughput.RFC2544. PacketLossRateFrmMod * Throughput.RFC2544. BackToBackFrames * Throughput.RFC2889. ForwardingRate * Throughput.RFC2889. ForwardPressure * Throughput.RFC2889. BroadcastFrameForward * RFC2889 Broadcast Frame Latency test	* Throughput.RFC2889. ErrorFramesFiltering	* Throughput.RFC2544. Soak * Throughput.RFC2544. SoakFrameModification	
De-Activation				

OPNFV

Test results repeatability considerations

- **Hardware details:** platform, processor, memory, # enabled cores, # used cores, # pNICs (PCI slot, interrupt config details…), BIOS settings (+ version), DIMM config, PCI config …
- **Software details:** OS version, kernel, GRUB boot params, hypervisor details, vSwitch details (memory allocation, thread affinitization…), SW lib dependencies, #VNFs, #vNICs…
- **Traffic details:** type (UDP/TCP/IMIX/Other), packet sizes and profiles.

Flow Classification

- A **flow** can be thought of a sequence of packets that have the same set of header field values or have arrived on the same port.
- The vSwitch will perform an action on packets that match a flow.
- Performance results can vary based on the parameters the vSwitch uses to match for a flow.
- The recommended flow classification parameters for any vSwitch performance tests are: the **input port**, the **source IP address**, the **destination IP address** and the **Ethernet protocol type** field.

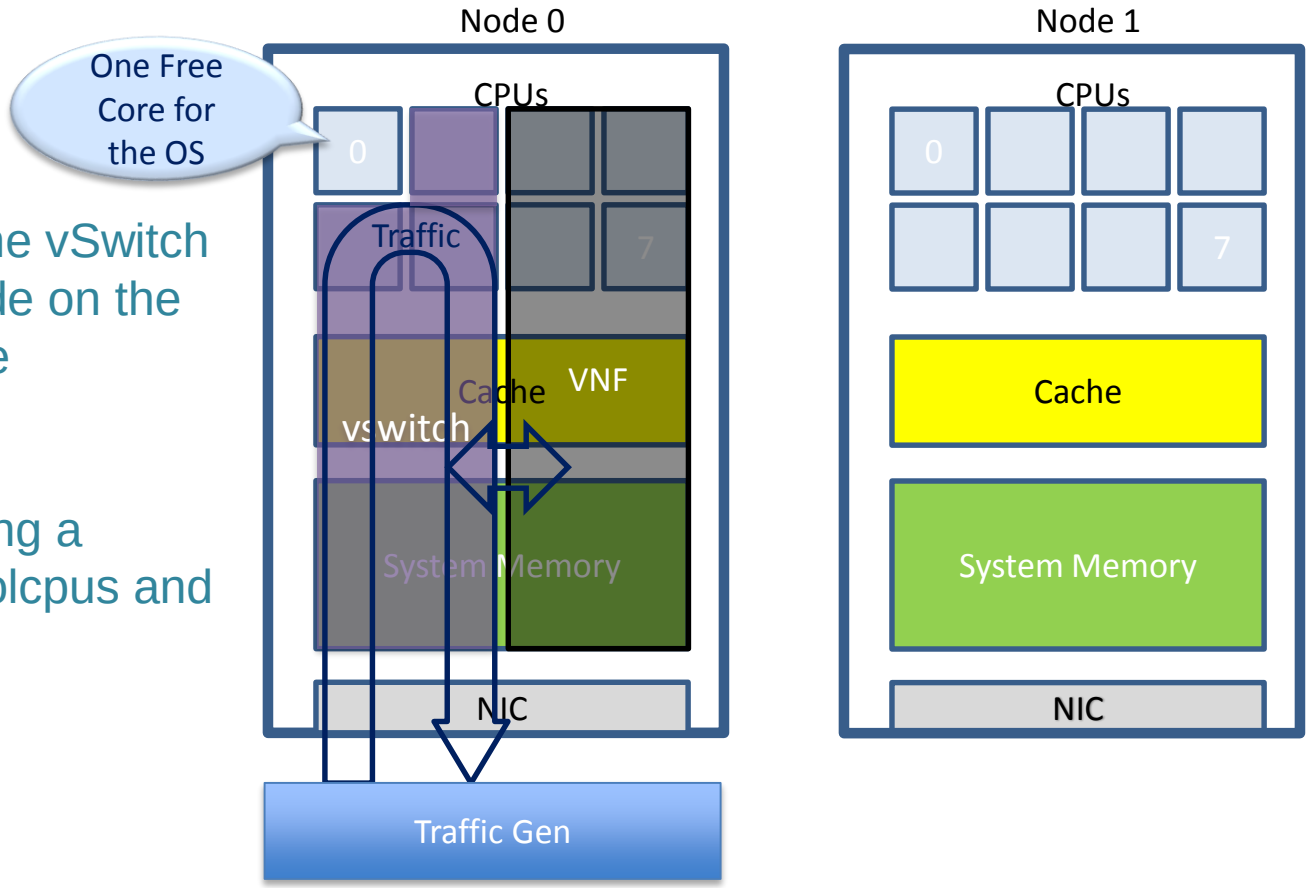
Benchmarks using baselines & resource isolation

- Baseline:
  - Optional: Benchmark platform forwarding capability.
  - Benchmark VNF forwarding capability.
  - Benchmarking with isolated resources alone, with other resources (both HW&SW) disabled.
    - Example, vSw and VM are SUT
  - benchmarking with isolated resources alone, leaving some resources unused.
  - Isolated resources and all resources occupied.

Proposed Approach

SUT (vSwitch + VNF) typical configuration

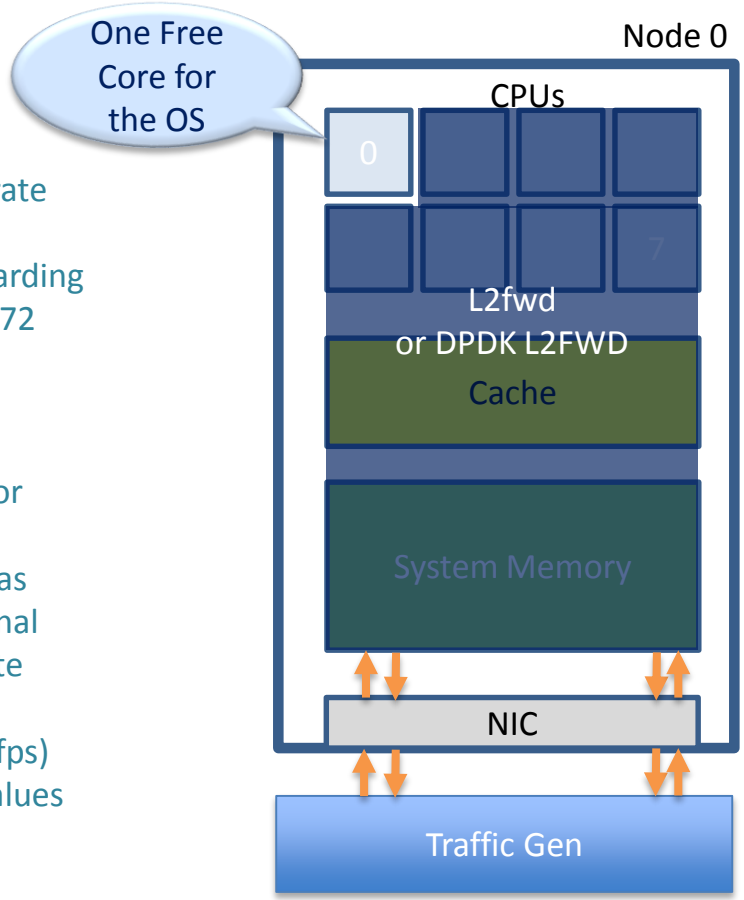
- NUMA – Ideally the vSwitch and the VNF reside on the same NUMA node
- CPU pinning – accomplished using a combination of isolcpus and taskset.



Benchmark platform forwarding capability

OPTIONAL

- Run RFC 2889 Maximum forwarding rate test
- Transmit traffic at line rate/max forwarding rate (whichever is higher) for at least 72 hours, measure throughput (fps) and latency.
- Traffic should be **bidirectional**.
- Establish a baseline forwarding rate for what the platform can achieve.
- Additional validation: After the test has completed for 72hours run bidirectional traffic at the maximum forwarding rate once more to see if the system is still functional and measure throughput (fps) and latency. Compare the measure values with what is expected.

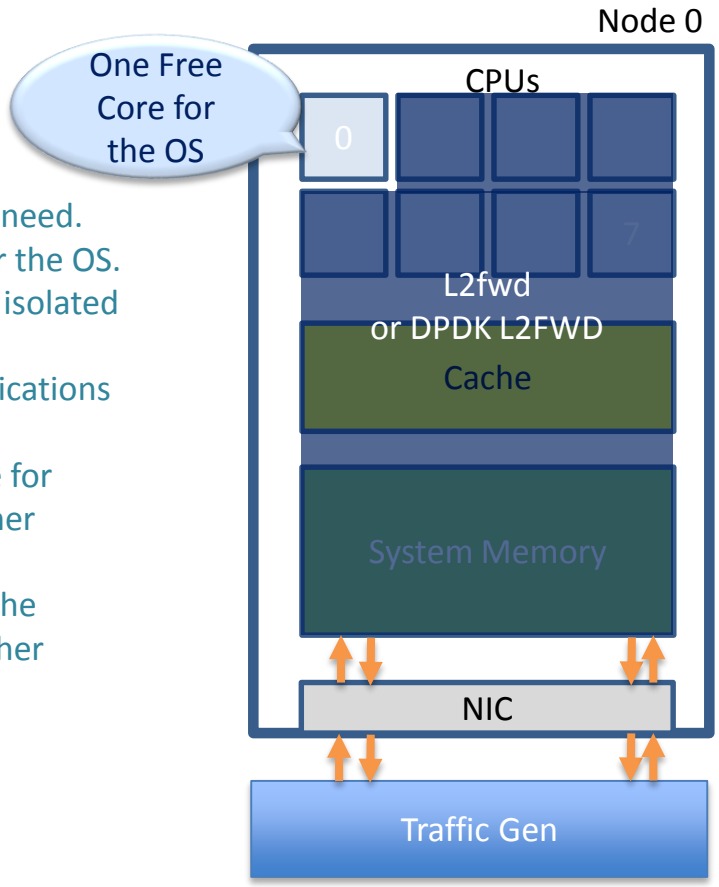


Benchmark platform forwarding capability Settings

OPTIONAL

Settings:

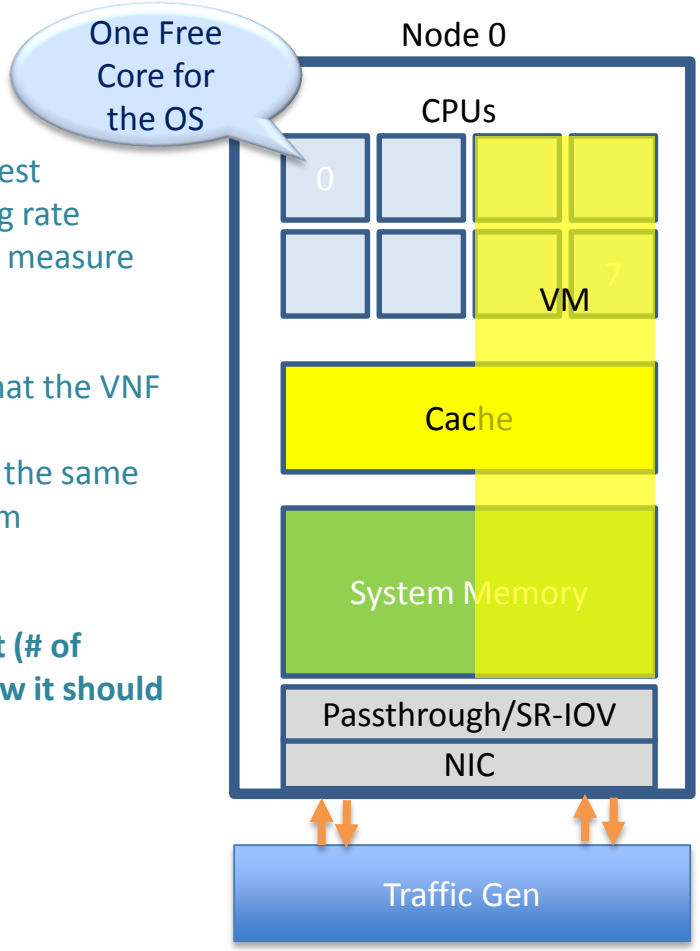
- maxcpus = only use exactly what you need.
- Isolcpus = everything bar one core for the OS.
- Taskset the forwarding application to isolated cores.
- Limit the amount of background applications that are running.
- Only enable HW that you need to use for your test – to ensure there are no other interrupts on the system.
- Configure NIC interrupts to only use the cores that are not allocated to any other process.
- Set OS to boot to runlevel 3
- **Disable other sockets in BIOS**



Benchmark VNF forwarding capability

- Run RFC 2889 Maximum forwarding rate test
- Transmit traffic at line rate/max forwarding rate (whichever is higher) for at least 72 hours, measure throughput (fps) and latency.
- Traffic should be bidirectional.
- Establish a baseline forwarding rate for what the VNF can achieve.
- Additional validation can be carried out in the same manner as specified in Benchmark platform forwarding capability.

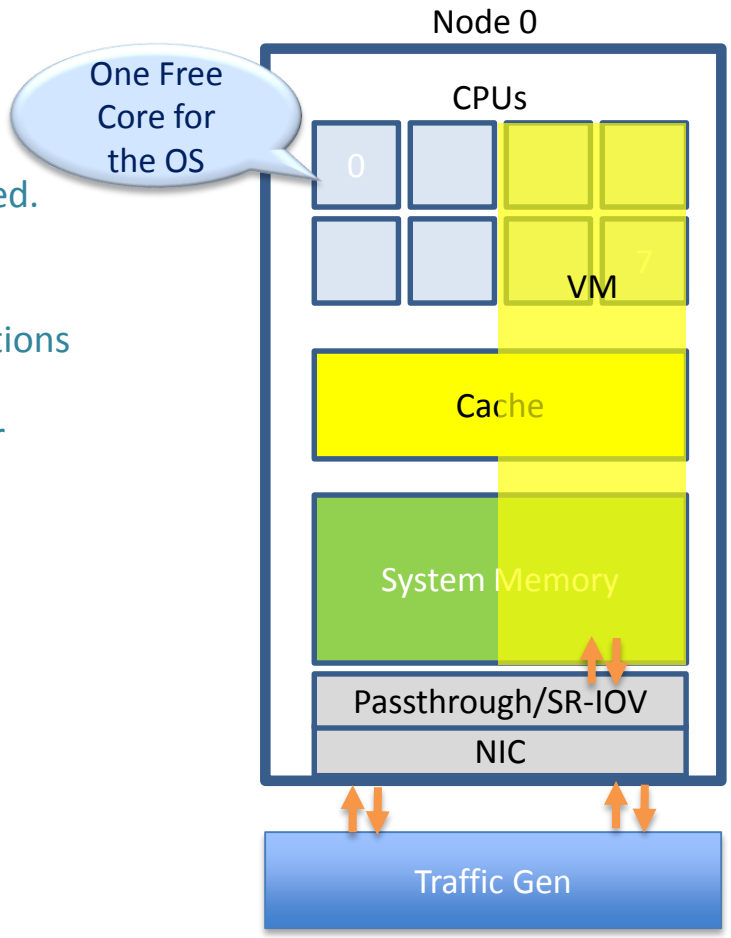
NOTE: How we configure the VM for this test (# of vCPUs, vNICs, Memory, affinitization…) is how it should be configured for every test after this.



Benchmark VNF forwarding capability Settings

Settings:

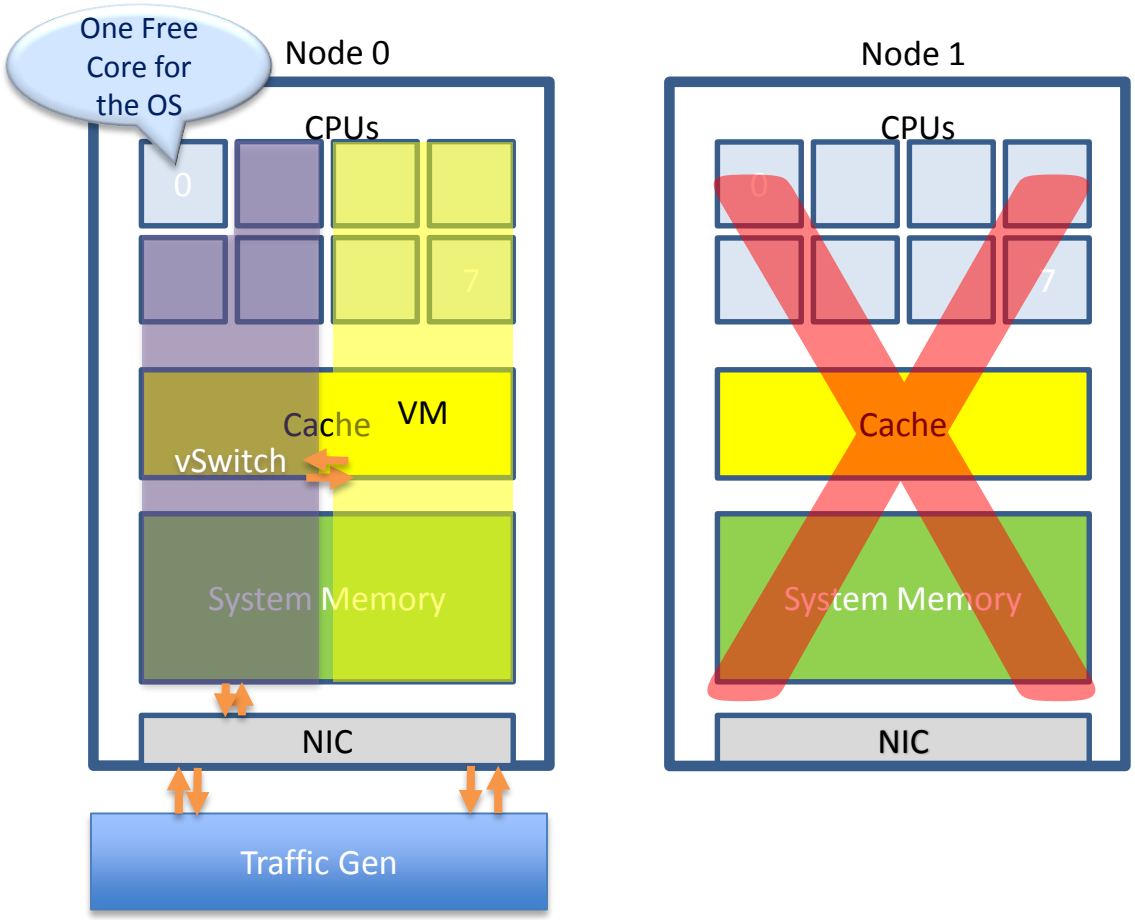
- maxcpus = only use exactly what you need.
- Isolcpus = Leave one core for the OS.
- Taskset the VM to isolated cores.
- Limit the amount of background applications that are running.
- Only enable HW that you need to use for your test – to ensure there are no other interrupts on the system.
- Configure NIC interrupts to only use the cores that are not allocated to any other process.
- Affinitize vCPUs to separate cores.
- Set OS to boot to runlevel 3
- **Disable other sockets in BIOS**



Benchmark with isolated resources **alone**

Settings:

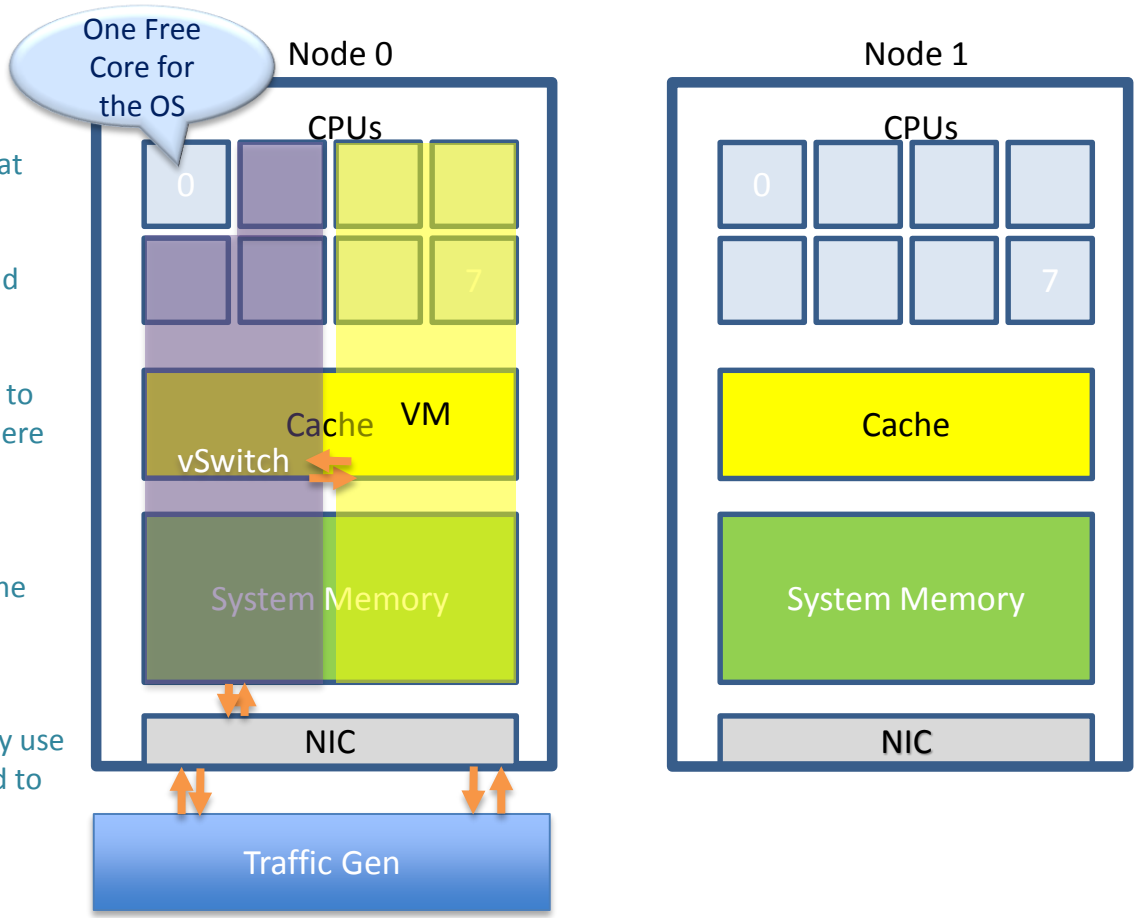
- maxcpus = only use exactly what you need.
- Limit the amount of background applications that are running
- Only enable HW that you need to use for your test – to ensure there are no other interrupts on the system.
- Isolcpus = Leave one core for the OS.
- Set OS to boot to runlevel 3.
- Configure NIC interrupts to only use the cores that are not allocated to any other process (VNF).
- **Disable other sockets in BIOS**



Benchmark with isolated resources

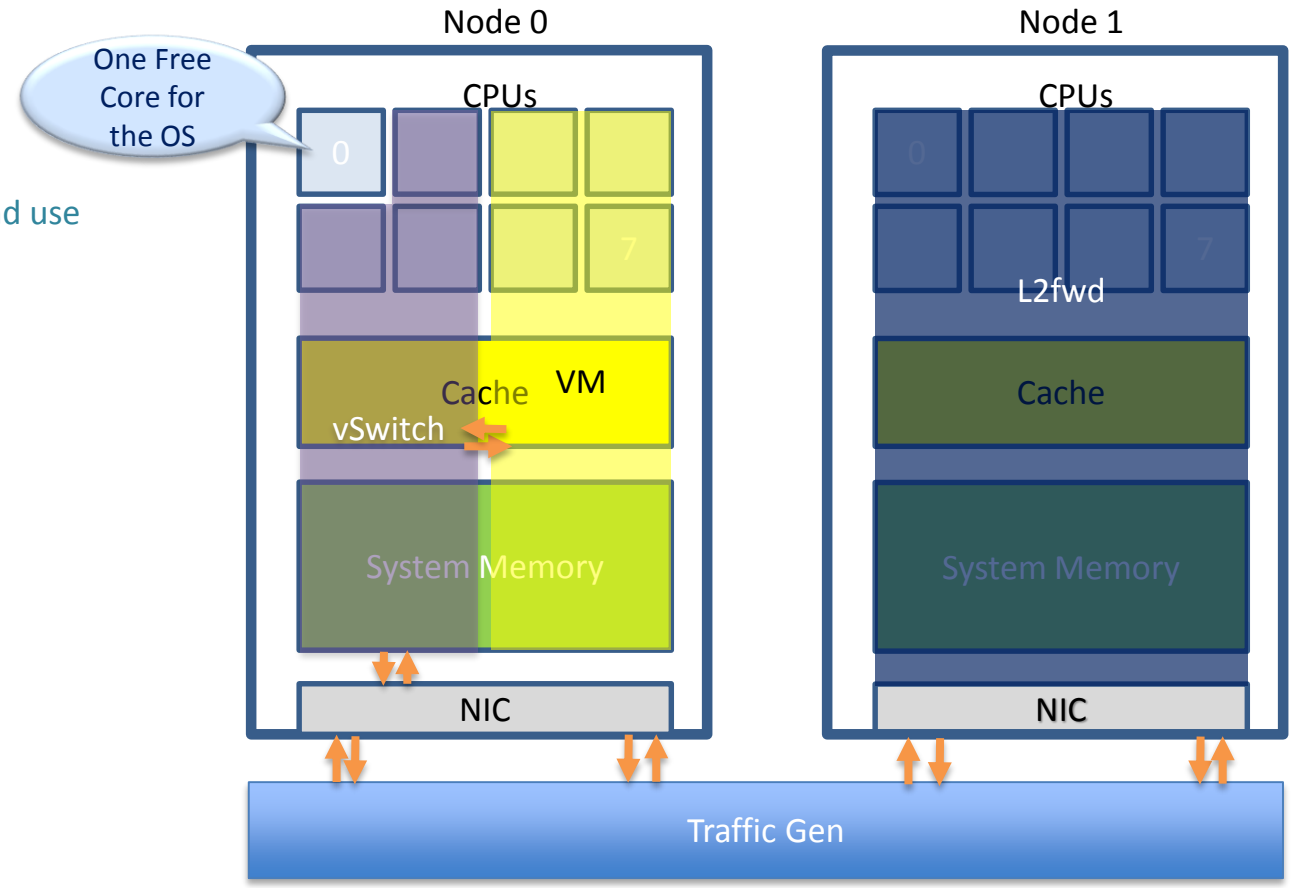
Settings:

- maxcpus = only use exactly what you need.
- Limit the amount of background applications that are running.
- Only enable HW that you need to use for your test – to ensure there are no other interrupts on the system.
- Isolcpus = Leave one core for the OS.
- Set OS to boot to runlevel 3.
- Configure NIC interrupts to only use the cores that are not allocated to any other process (VNF).
- **Enable the 2nd socket in BIOS.**



Benchmark with isolated resources and all resources occupied

Enable and use all CPUs.



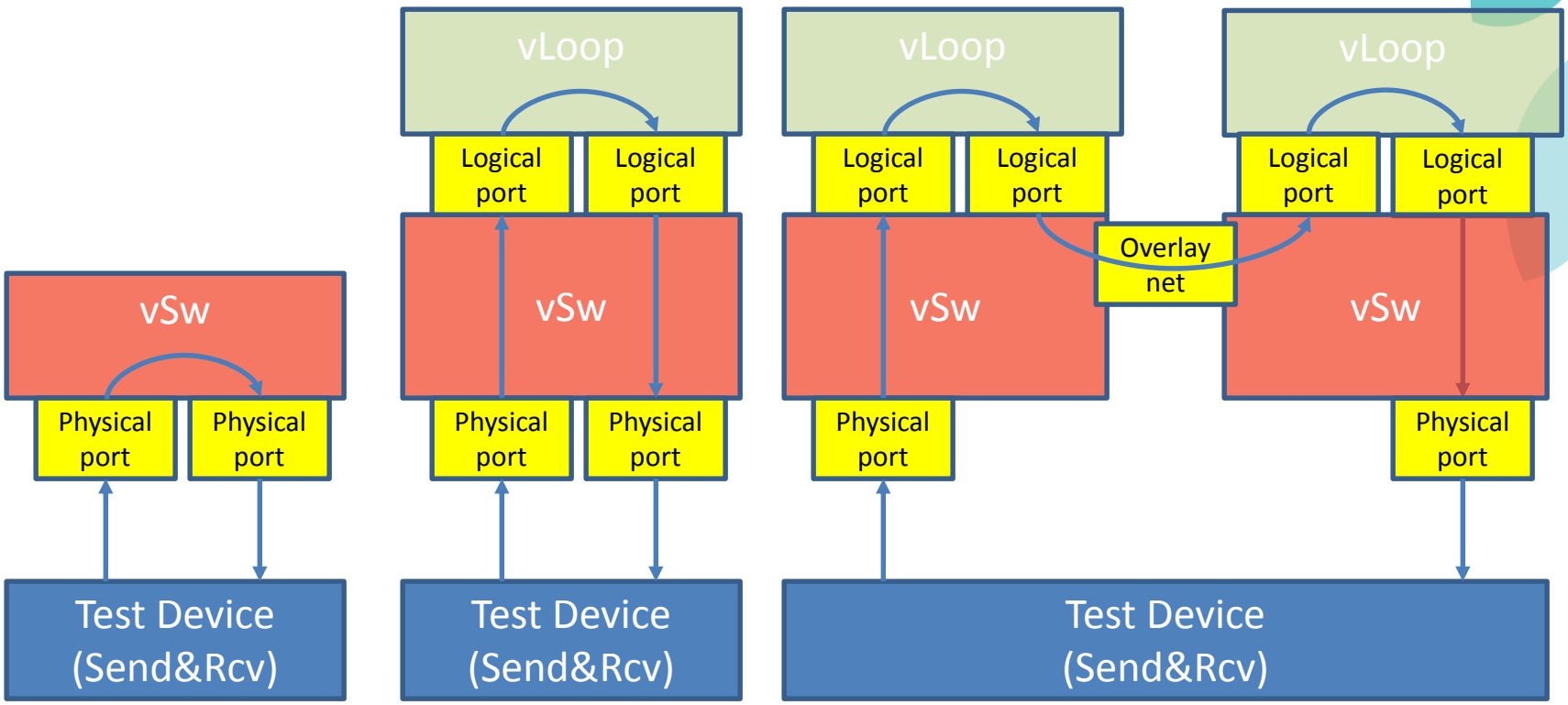
Additional Notes and Summary

- TOIT – Open Source test framework that implements the LTD
- Full LTD spec Available

- Authors are Interested in BMWG opinion on further development and next steps:
  - We've seen/heard/read enough, thanks
  - Some may want to join OPNFV work
  - Keep developing the draft as a Summary & Pointer to LTD+tools
  - Expand the draft to cover full LTD with BMWG review (WG)
  - Other thoughts

BACKUP

Additional Test Setups (single traffic direction shown)



vSwitch deployment scenarios

- Physical port → vSwitch → physical port .
- Physical port → vSwitch → VM → vSwitch → physical port .
- Physical port → vSwitch → VM → vSwitch → VM → vSwitch → physical port .
- Physical port → vSwitch → VM.
- VM → vSwitch → physical port.
- VM → vSwitch → VM.

Please note a Physical port is connected to a traffic generator. A VM is connected to the vSwitch through a logical port.