# Considerations for Benchmarking Network Performance in Containerized Infrastructure

**draft-dcn-bmwg-containerized-infra-10**

Minh-Ngoc Tran (Soongsil University), Sridhar Rao (The Linux Foundation),

Jangwon Lee, Younghan Kim (Soongsil University)

# Introduction

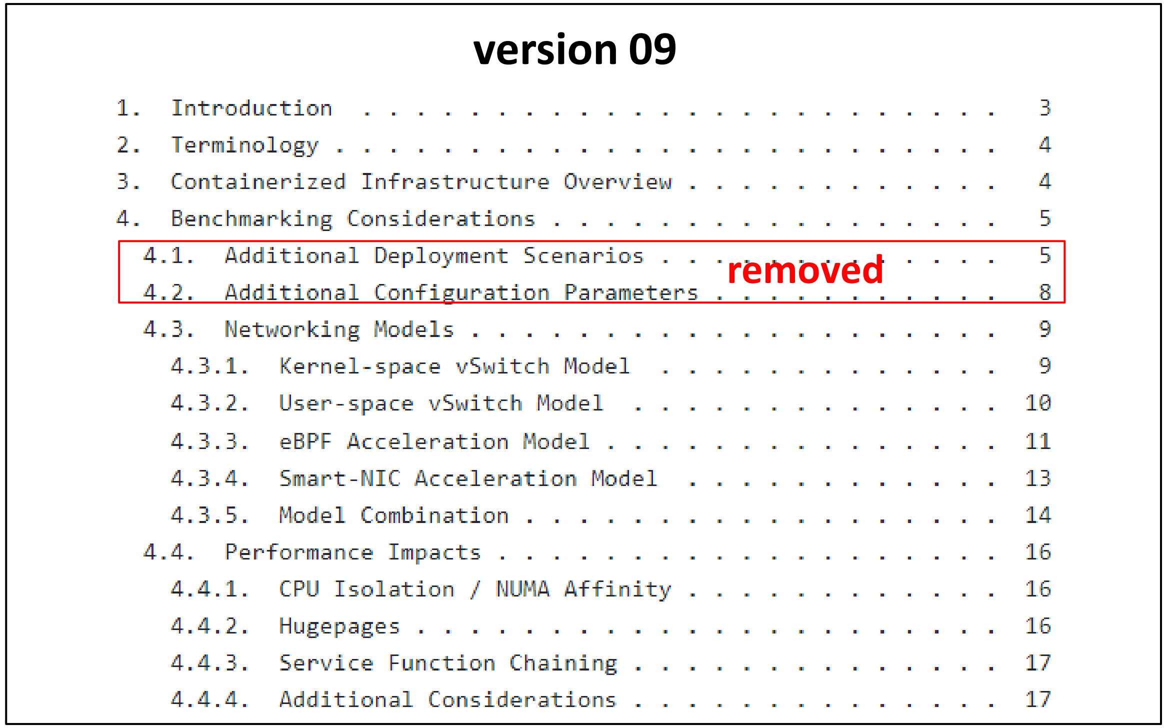
- Thanks Al and Sridhar for reviewing our draft before meeting!

- This draft aims to provide additional considerations as specifications to guide containerized infrastructure benchmarking, compared with previous benchmarking methodology of common NFV infrastructure

- The considerations include:
  - Investigation of **different container networking models** based on the usage of different packet acceleration techniques
  - Investigation of **different resources configuration settings** (NUMA, hugepages, etc.) **that might make performance impacts** on network performance

# Updates Summary (from v9 to v10)

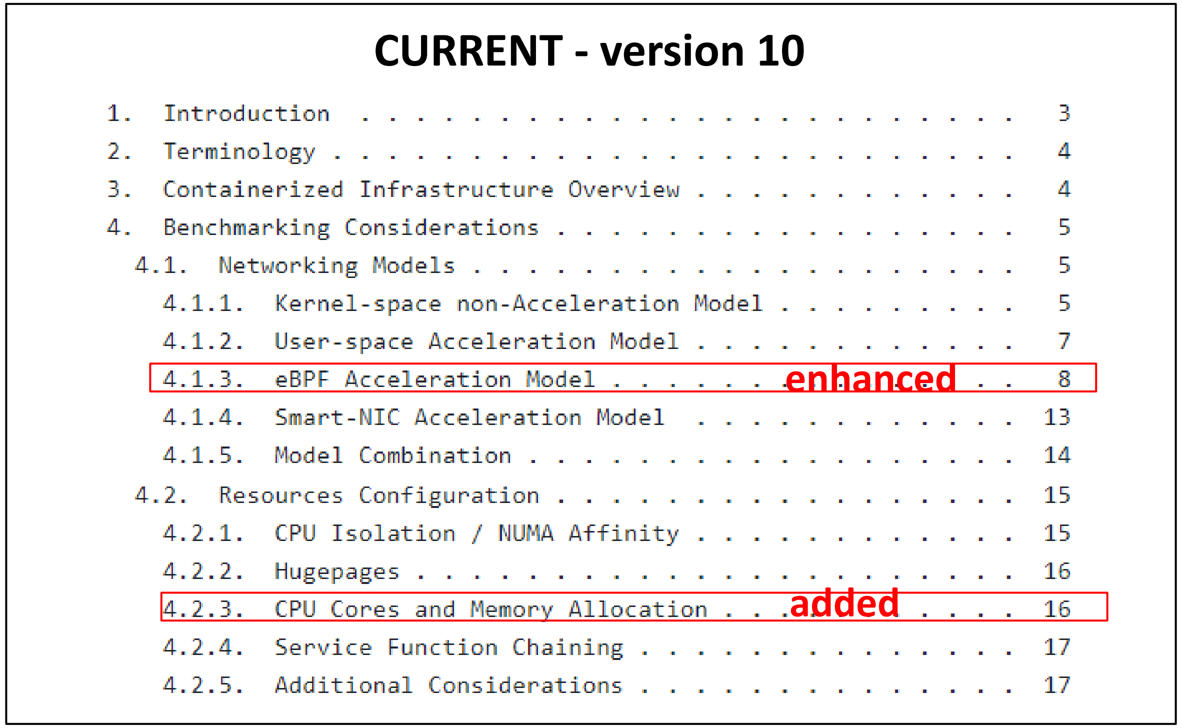
- After discussing our draft with VinePerf from Anuket Project – The Linux Foundation (Sridhar Rao and Al Morton):
  - Removed Sections: Additional Deployment Scenarios and Additional Configuration Parameters (ver 09 – section 4.1, 4.2)
  - Enhanced Section: eBPF Acceleration Model (ver 10 – section 4.1.3)
  - Added Section: CPU Cores and Memory Allocation (ver 10 – section 4.2.3)



**version 09**

```
1.  Introduction  . . . . . . . . . . . . . . . . . . . . . . . .   3
2.  Terminology . . . . . . . . . . . . . . . . . . . . . . . . .   4
3.  Containerized Infrastructure Overview . . . . . . . . . . . .   4
4.  Benchmarking Considerations . . . . . . . . . . . . . . . . .   5
  4.1.  Additional Deployment Scenarios . . . . . . . . . . . . .   5
  4.2.  Additional Configuration Parameters . . . . . . . . . . .   8
  4.3.  Networking Models . . . . . . . . . . . . . . . . . . . .   9
    4.3.1.  Kernel-space vSwitch Model  . . . . . . . . . . . . .   9
    4.3.2.  User-space vSwitch Model  . . . . . . . . . . . . . .  10
    4.3.3.  eBPF Acceleration Model . . . . . . . . . . . . . . .  11
    4.3.4.  Smart-NIC Acceleration Model  . . . . . . . . . . . .  13
    4.3.5.  Model Combination . . . . . . . . . . . . . . . . . .  14
  4.4.  Performance Impacts . . . . . . . . . . . . . . . . . . .  16
    4.4.1.  CPU Isolation / NUMA Affinity . . . . . . . . . . . .  16
    4.4.2.  Hugepages . . . . . . . . . . . . . . . . . . . . . .  16
    4.4.3.  Service Function Chaining . . . . . . . . . . . . . .  17
    4.4.4.  Additional Considerations . . . . . . . . . . . . . .  17
```

removed



**CURRENT - version 10**

```
1.  Introduction  . . . . . . . . . . . . . . . . . . . . . . . .   3
2.  Terminology . . . . . . . . . . . . . . . . . . . . . . . . .   4
3.  Containerized Infrastructure Overview . . . . . . . . . . . .   4
4.  Benchmarking Considerations . . . . . . . . . . . . . . . . .   5
  4.1.  Networking Models . . . . . . . . . . . . . . . . . . . .   5
    4.1.1.  Kernel-space non-Acceleration Model . . . . . . . . .   5
    4.1.2.  User-space Acceleration Model . . . . . . . . . . . .   7
    4.1.3.  eBPF Acceleration Model . . . . . . . . . . . . . . .   8
    4.1.4.  Smart-NIC Acceleration Model  . . . . . . . . . . . .  13
    4.1.5.  Model Combination . . . . . . . . . . . . . . . . . .  14
  4.2.  Resources Configuration . . . . . . . . . . . . . . . . .  15
    4.2.1.  CPU Isolation / NUMA Affinity . . . . . . . . . . . .  15
    4.2.2.  Hugepages . . . . . . . . . . . . . . . . . . . . . .  16
    4.2.3.  CPU Cores and Memory Allocation . . . . . . . . . . .  16
    4.2.4.  Service Function Chaining . . . . . . . . . . . . . .  17
    4.2.5.  Additional Considerations . . . . . . . . . . . . . .  17
```

enhanced

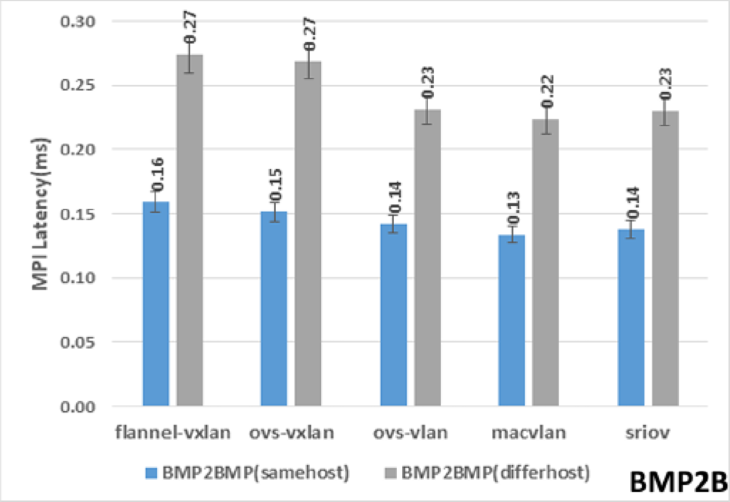
added

# Detailed Updates (1)

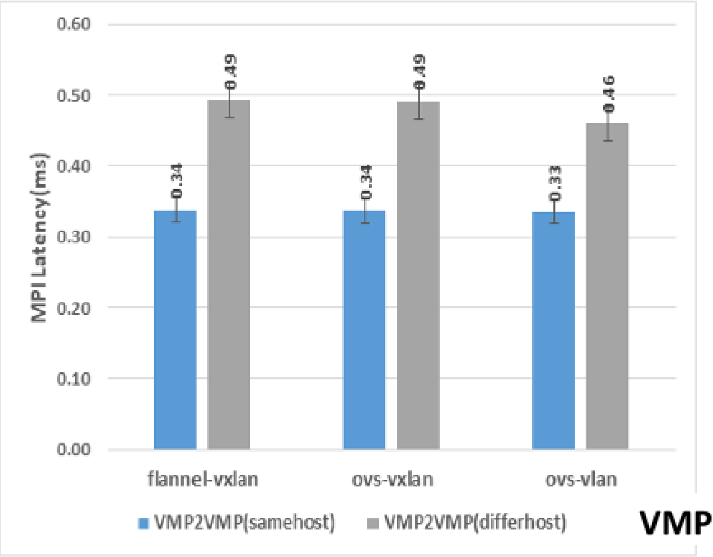
## Removed Sections

- **Additional Deployment Scenarios**
  - The considered scenarios are:
    - Pods are deployed on Bare Metal (BMP)
    - Pods are deployed on Virtual Machine (VMP)
  - Both us and VinePerf tested similar scenarios and the performance difference is negligible, differences only caused by the chosen networking technologies
- **Additional Configuration Parameters**
  - We agreed with VinePerf that this should be placed in the Resources Configuration consideration section

- VLAN technologies(ovs-vlan, macvlan, sriov) are shown better performance up to 10% than overlay network (vxlan) for all test scenarios.



BMP2BMP

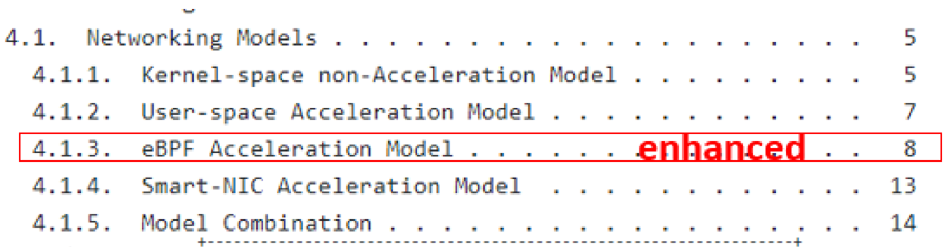


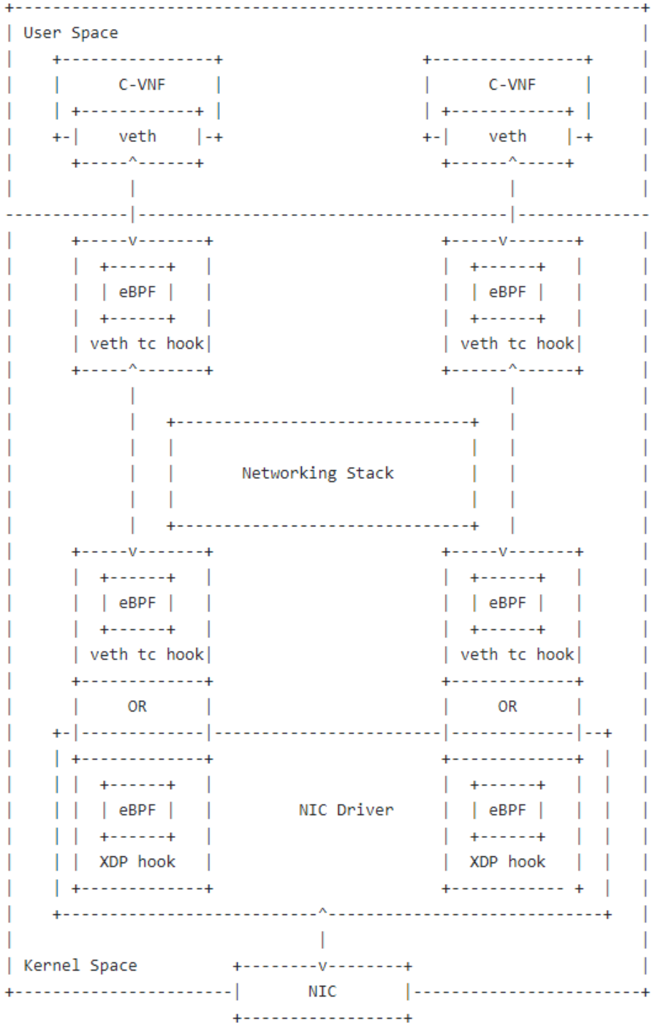
VMP2VMP

Tested Results from slides-104-bmwg-considerations-for-benchmarking-network-performance-in-containerized-infrastructures-00

# Detailed Updates (2)

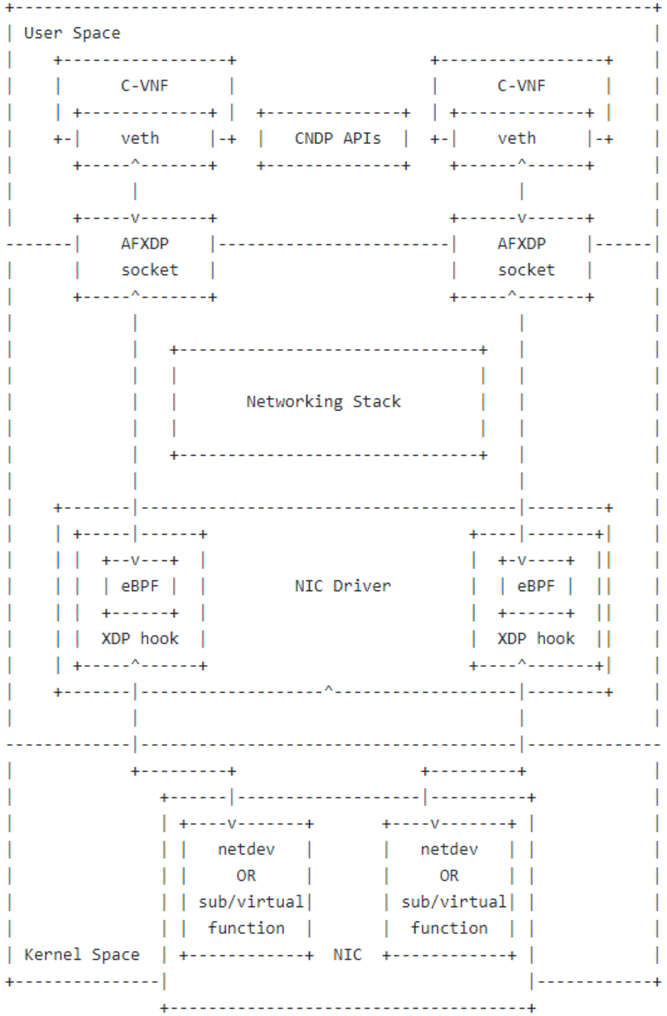
## Enhanced Section- eBPF Acceleration Model





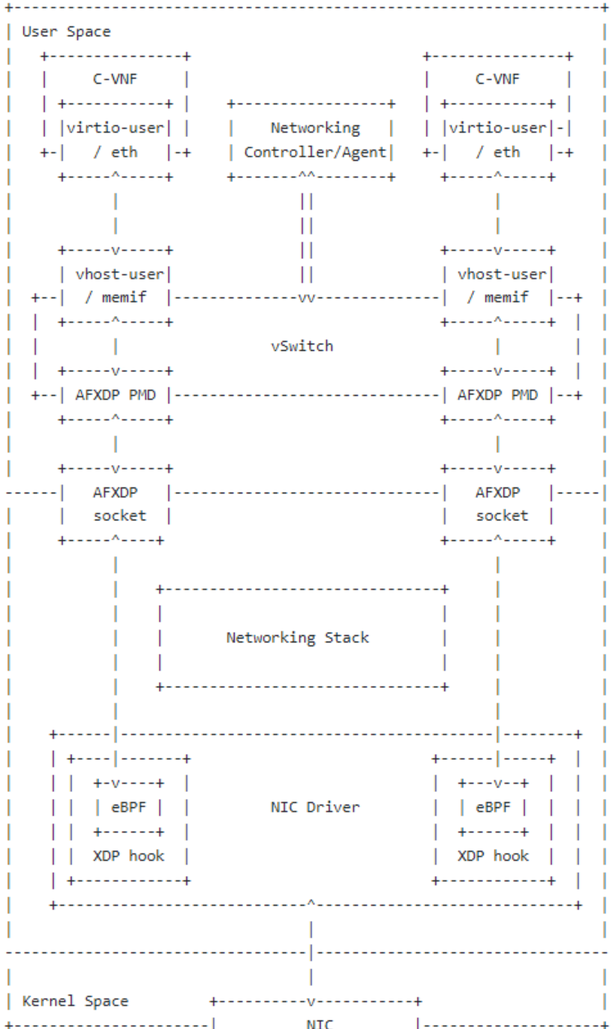
**non-AFXDP**

(Cilium, Calico-eBPF)



**Using AFXDP supported CNI**

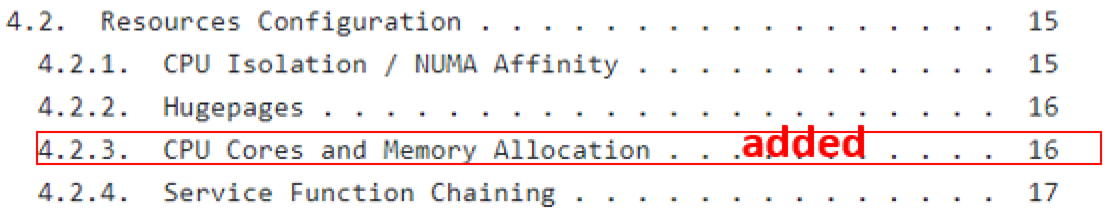
(AFXDP K8s CNI – used by Intel CNDP)



**Using AFXDP supported userspace vSwitch**

(OVS-DPDK, VPP with AFXDP enabled)

# Detailed Updates (3)



added

## New Section – CPU Cores and Memory Allocation

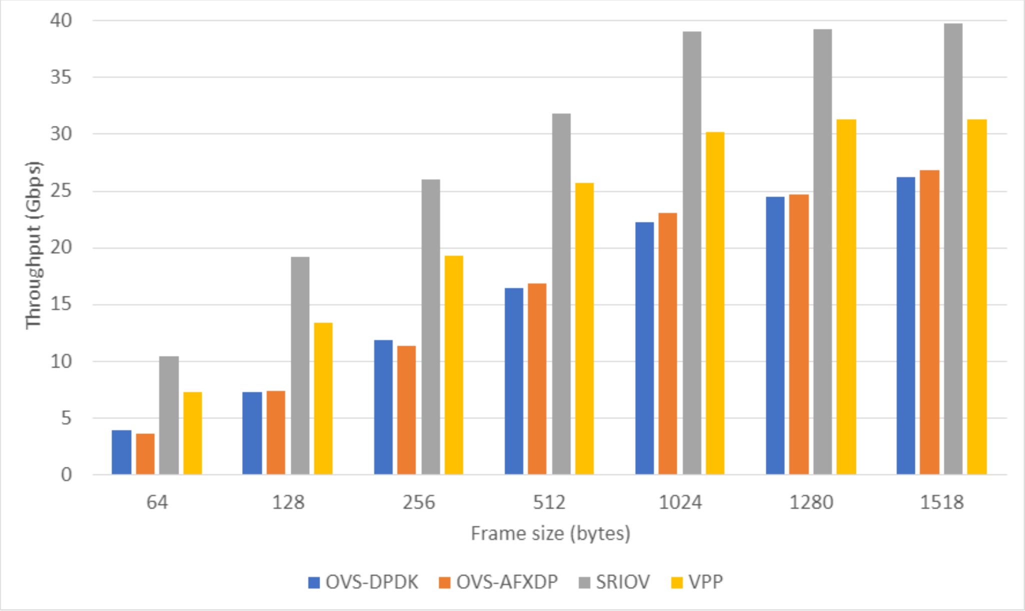
- Different CPU cores and memory allocation to Pods and vSwitch Poll Mode Driver might impact the container network performance

- Both our Hackathon benchmarking test and VinePerf's test has investigated the impact of these settings. VinePerf result was described in:
  - Benchmarking Kubernetes Container-Networking for Telco Usecases (Sridhar Rao, Federica Paganelli, Al Morton - 2021 GLOBECOM)

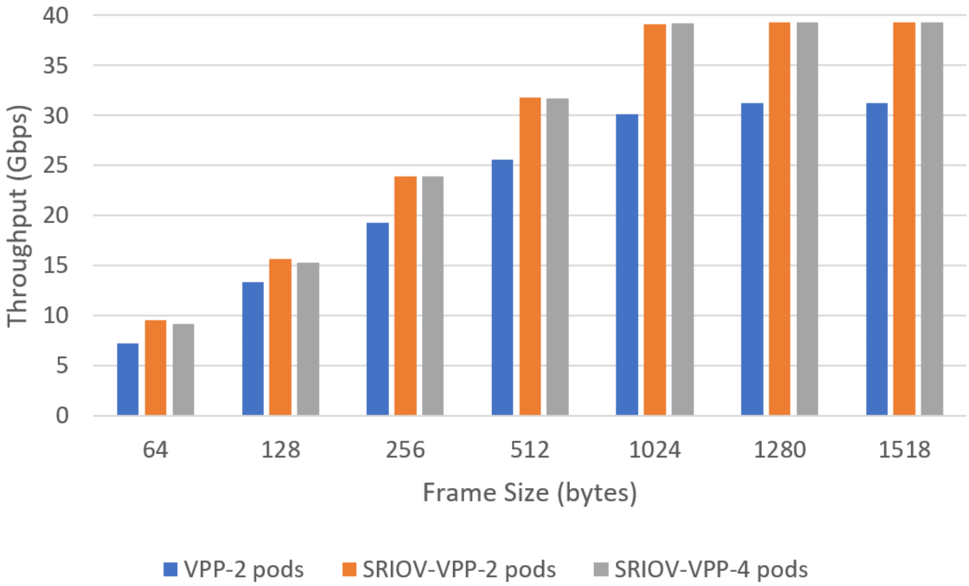
- Summary:
  - Increasing Pod's CPU significantly increases the the throughput
  - Different RAM allocation cause different, inconsistent throughput
  - Increasing CPU cores allocation to VPP vSwitch cause better latency, but not with OvS-DPDK

# From Hackathon

- Verify current networking model, and resources configuration consideration in the draft (short hackathon time, each test was run 5 times, standard deviation ≈ 0.15)
  - Different performance caused by different networking models



Comparison between
Userspace Model vs eBPF-AFXDP vs SRIOV

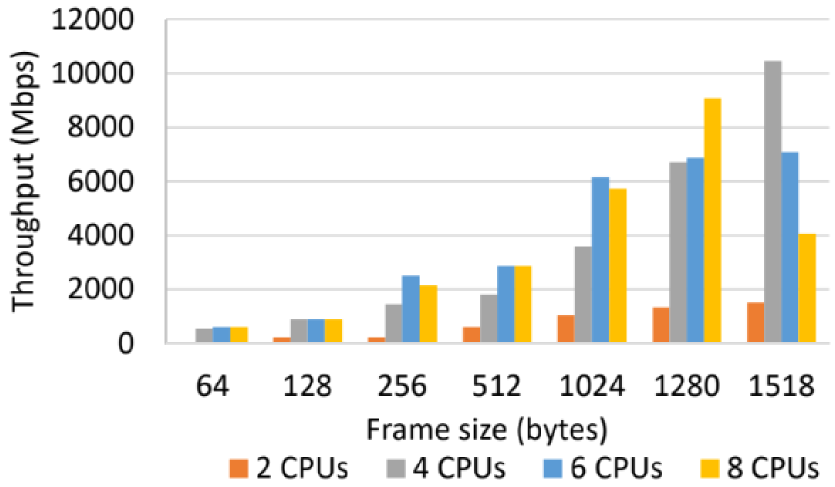


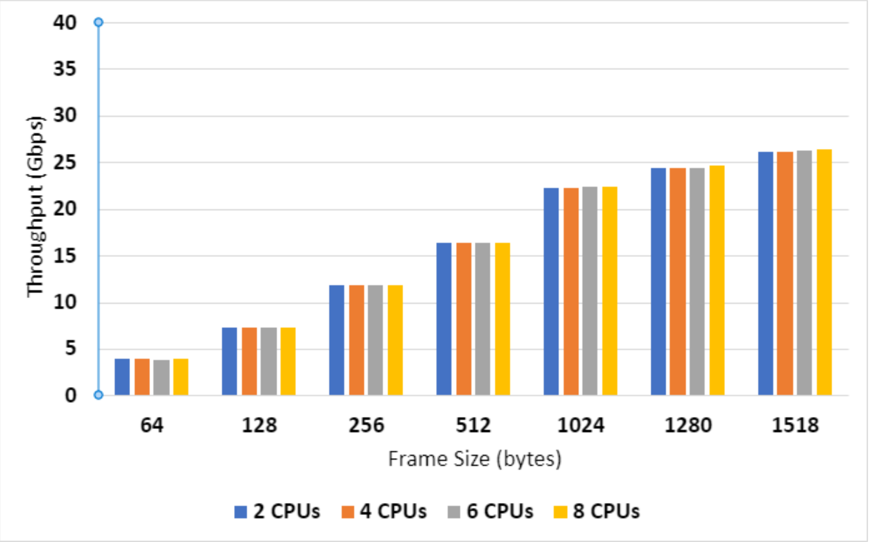
Combined model performance with impact of
different number of C-VNF

*Hackathon 116 Results tested on OVS-DPDK, 40Gb NIC*
*OvS version 3.10, DPDK version 22.10, VPP 19.04, AFXDP K8s plugin latest*

# From Hackathon

- Verify all current networking model, and resources configuration consideration in the draft (short hackathon time, each test was run 5 times, standard deviation ≈ 0.15)
  - Different performance caused by resources configuration settings
  - Our hackathon 116 results did not observe the impact of CPU and Memory allocation to pod with recent vSwitch and DPDK version (need to inform VinePerf)





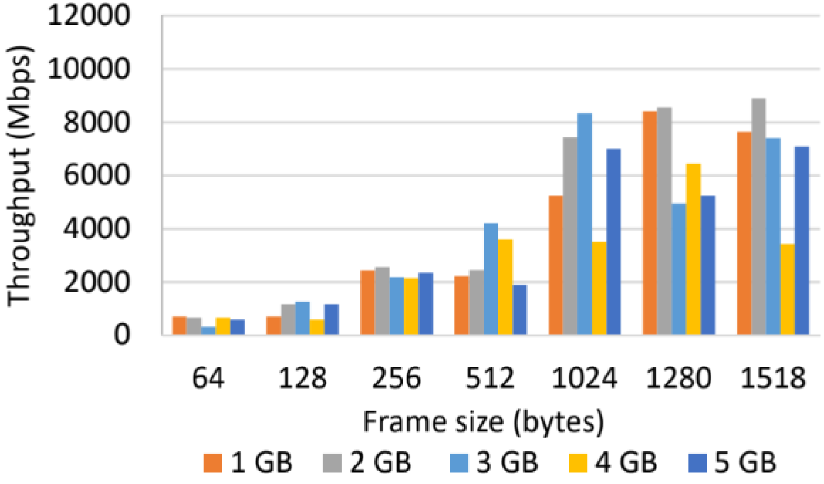
Different Pod's CPU cores allocation performances

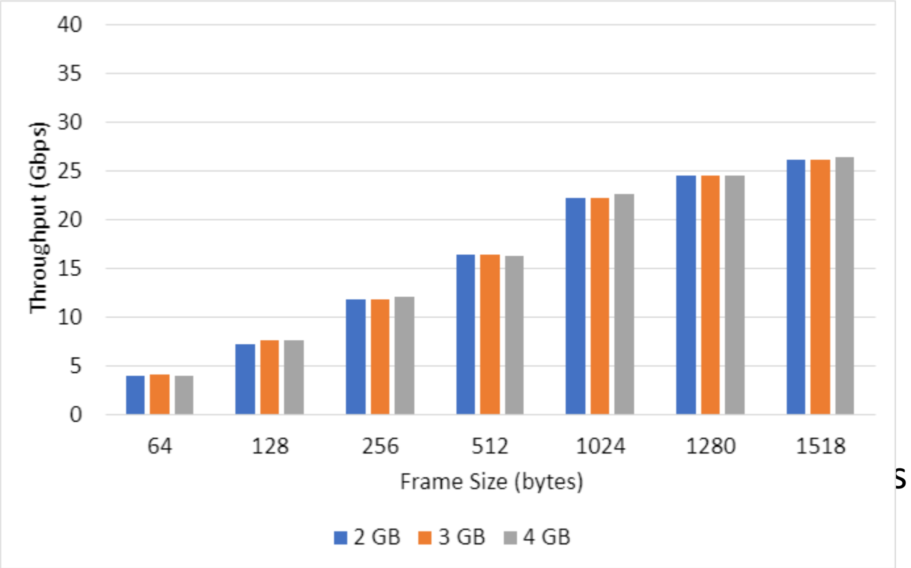
*VinePerf Results tested on OVS-DPDK, 10Gb NIC*
*OvS version 2.12, DPDK version 19.08*

*Hackathon 116 Results tested on OVS-DPDK, 40Gb NIC*
*OvS version 3.10, DPDK version 22.10*

# From Hackathon

- Verify all current networking model, and resources configuration consideration in the draft (short hackathon time, each test was run 5 times, standard deviation ≈ 0.15)
  - Different performance caused by resources configuration settings
  - Our hackathon 116 results did not observe the impact of CPU and Memory allocation to pod with recent vSwitch and DPDK version (need to inform VinePerf)





s

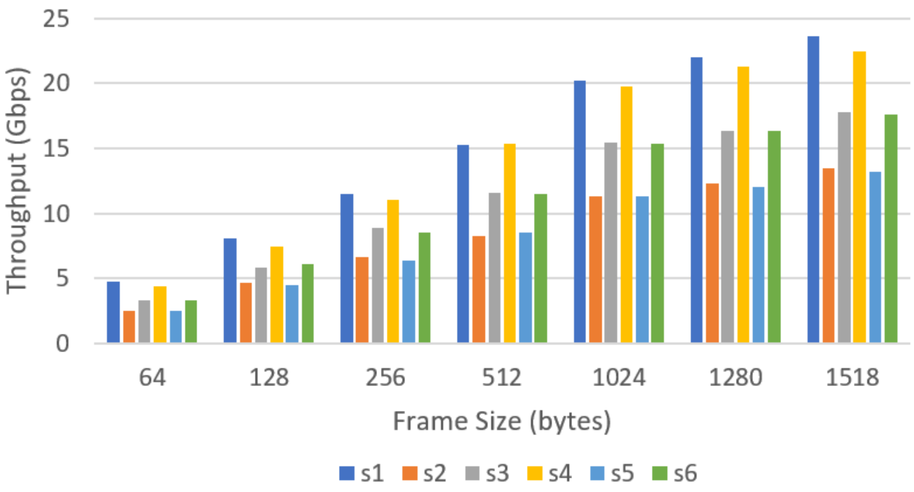
Different Pod's memory allocation performances

*VinePerf Results tested on OVS-DPDK, 10Gb NIC*
*OvS version 2.12, DPDK version 19.08*

*Hackathon 116 Results tested on OVS-DPDK, 40Gb NIC*
*OvS version 3.10, DPDK version 22.10*

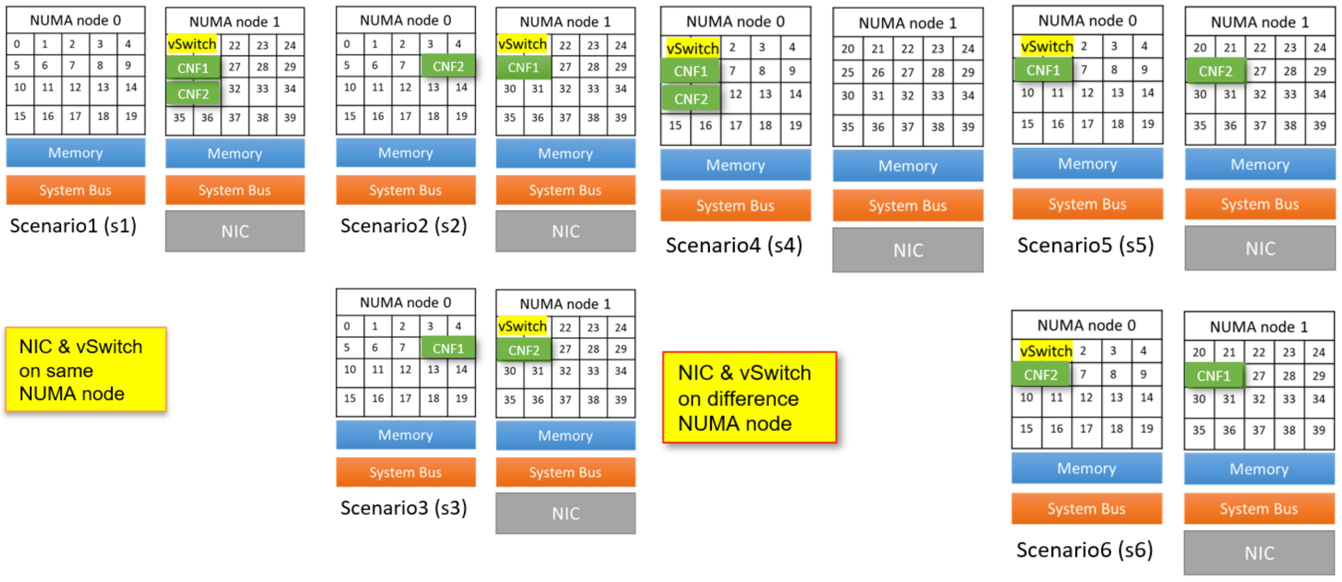
# From Hackathon

- Verify all current networking model, and resources configuration consideration in the draft
  - Different performance caused by resources configuration settings
  - Aligning pod, vSwitch and NIC in the same NUMA for highest performance



Different NUMA alignment performances

*Hackathon 112 Results tested on VPP 19.04, 40Gb NIC*



NUMA alignment scenarios

# Conclusion

- We would like ask adoption of this draft as a working group draft

- Feedbacks and reviews are welcome

# Backup Slides

# From Hackathon 116

- Benchmarking Configuration

- Hardware – Worker Node

| | |
|---|---|
| CPU | Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz<br>48 CPU cores * 2 NUMA nodes |
| Memory | 256GB: 32GB x 4DIMMs x 2 NUMA nodes @ 2400MHz |
| NIC | Intel Corporation Ethernet Network Adapter X71-40Gbps |
| Microcode | 0x5003102 |
| Intel NIC Device ID | 0x1572 |
| Intel NIC Firmware version | 6.01 0x800035cf 1.1747.0 |
| BIOS setting | CPU Power and Performance Policy <Performance><br>CPU C-state Disabled<br>CPU P-state Disabled<br>Intel(R) Hyper-Threading Tech Enabled<br>Turbo Boost Disabled |

- Traffic Generator : T-Rex (v2.92)

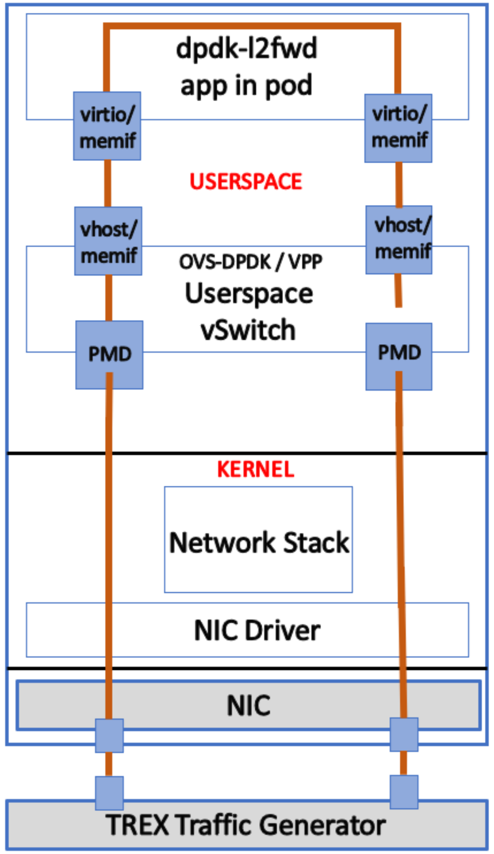
| | |
|---|---|
| Name | T-Rex |
| Version | 2.92 |
| Benchmark method | T-Rex Non Drop Rate application (accepted percentage of drop rate is less than 0.1%) |

- Software

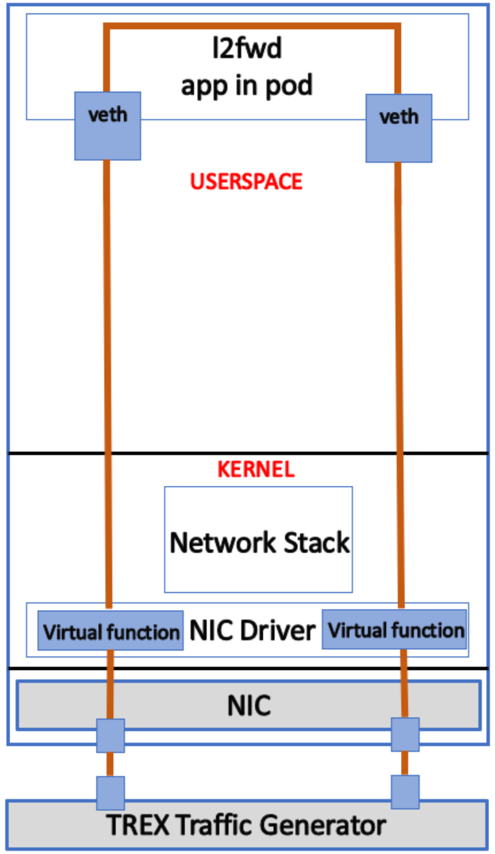
| | |
|---|---|
| Operating System | **Ubuntu 22.04** |
| Linux Kernel Version | 5.15 |
| GCC version | gcc version 4.8.5 20150623 (Red Hat 4.8.5-44) |
| DPDK version | **22.11.1** |
| Hugepages | 1Gi |

# What got done

- Different networking models based on packet acceleration techniques



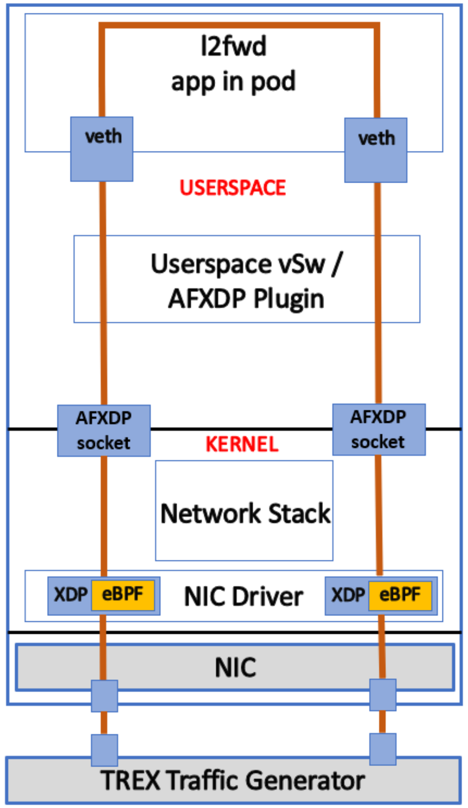
**Userspace Acceleration**
**(OVS-DPDK, VPP vSwitch)**



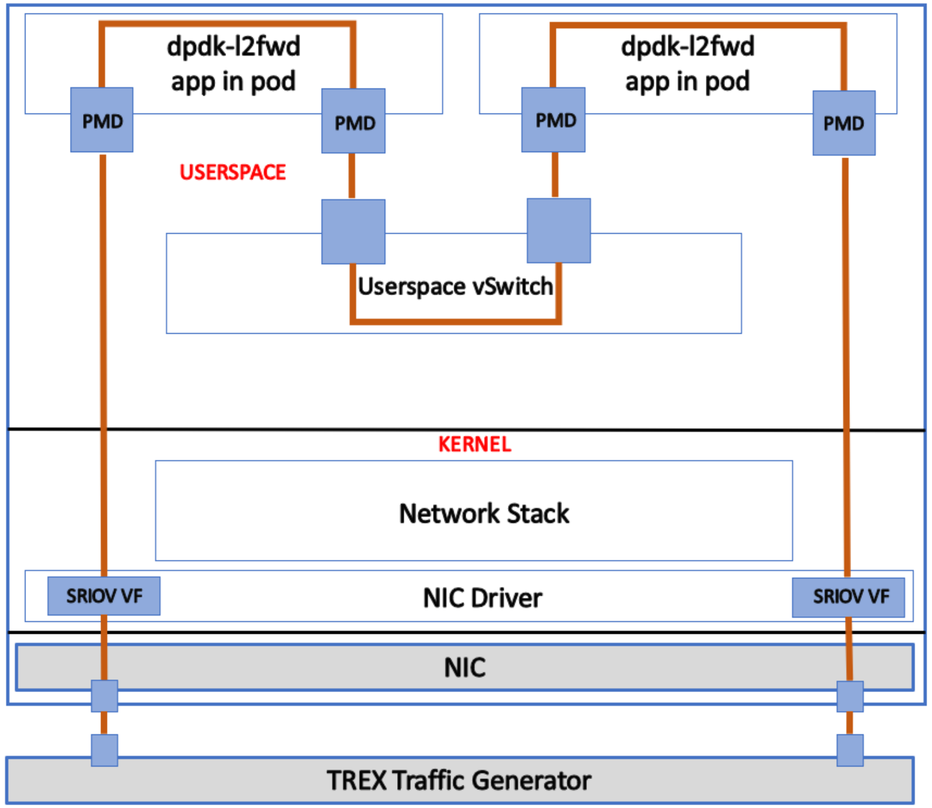
**SRIOV Acceleration**

# What got done

- Different networking models based on packet acceleration techniques



**eBPF Acceleration**
**(using AFXDP)**



**Combined Model Acceleration**
**(SR-IOV + OVS-DPDK/VPP)**