BMWG – Containerized Infrastructure Benchmarking

IETF 115 Hackathon

November 05-06, 2022

Remote

Hackathon Plan

- Our draft main goal is to figure out container networking performance impacts by various resource options.
  - Draft:
    Considerations for Benchmarking Network Performance in Containerized Infrastructures
    https://tools.ietf.org/html/draft-dcn-bmwg-containerized-infra
  - Two main features
    - Discuss **various network acceleration models** consideration that affect container network performance
    - Discuss **different deployment configuration settings** consideration that affect container network performance

Hackathon Plan

Previous Hackathon

Verify different eBPF Acceleration Models performances

- ✔ Initial test with OVS-AFXDP supported vSwitch
- ✔ vhost interface between vswitch and container

In this hackathon

- ➢ VPP vSwitch with memif interface
- ➢ Cloud Native Data Plane (CNDP) with K8s AFXDP plugin
- ➢ Cilium

What got done

- VPP-AFXDP vSwitch with memif interface

NIC ↔ Userspace vSwitch

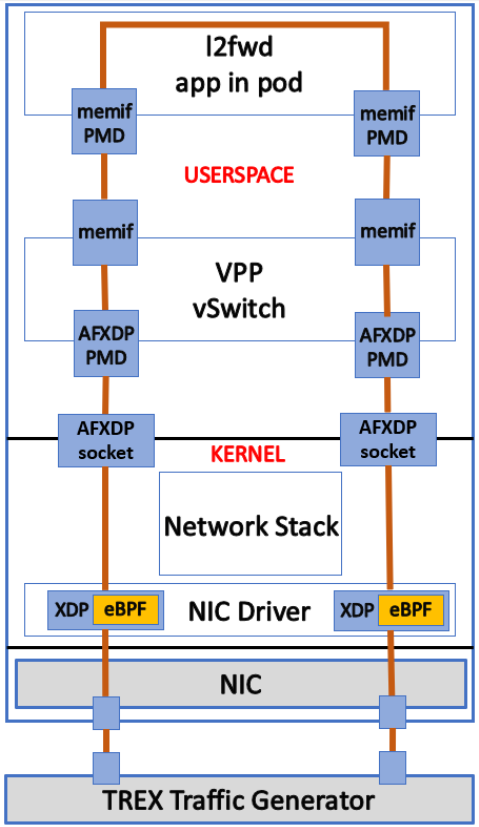
VPP-AF-XDP supported

- The AF-XDP socket receive packet directly from NIC rx queue, bypassing kernel
- AF-XDP PMD at VPP vSwitch to poll the packets from the socket

Userspace vSwitch ↔ Container

Memif interface

- Shared memory packet interface provides high performance packet transmit between user application and VPP vSwitch
- Perform better than vhost (which packets go through kernel space)



What got done

- Cloud Native Data Plane with AFXDP K8s Plugin

NIC ↔ Userspace

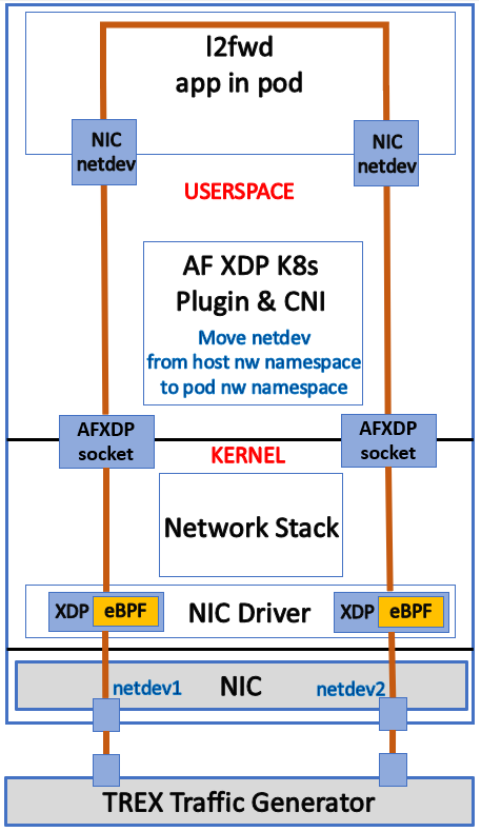
AF-XDP

- The AF-XDP socket receive packet directly from NIC rx queue, bypassing kernel

Userspace ↔ Container

CNDP AFXDP K8s CNI Plugin

- CNDP's AFXDP CNI Plugin move NIC netdev from host network namespace to pod network namespace
- Attached netdevs at Pod create AFXDP socket
- CNDP's AFXDP CNI Plugin configure busy polling on the socket to transmit packets to application



What got done

- Cilium CNI Plugin

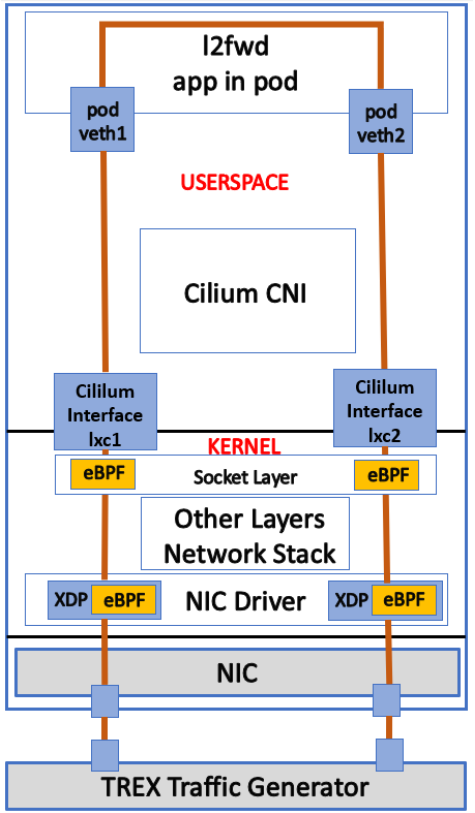
  NIC ↔ Userspace

  eBPF at driver layer and socket layer

  - eBPF at NIC driver layer handles North-South traffic
    - accelerated via XDP
  - eBPF at socket layer handles East-West traffic
    - accelerated by eliminating per-packet costs for service translation (memory allocation for packet metadata,…)

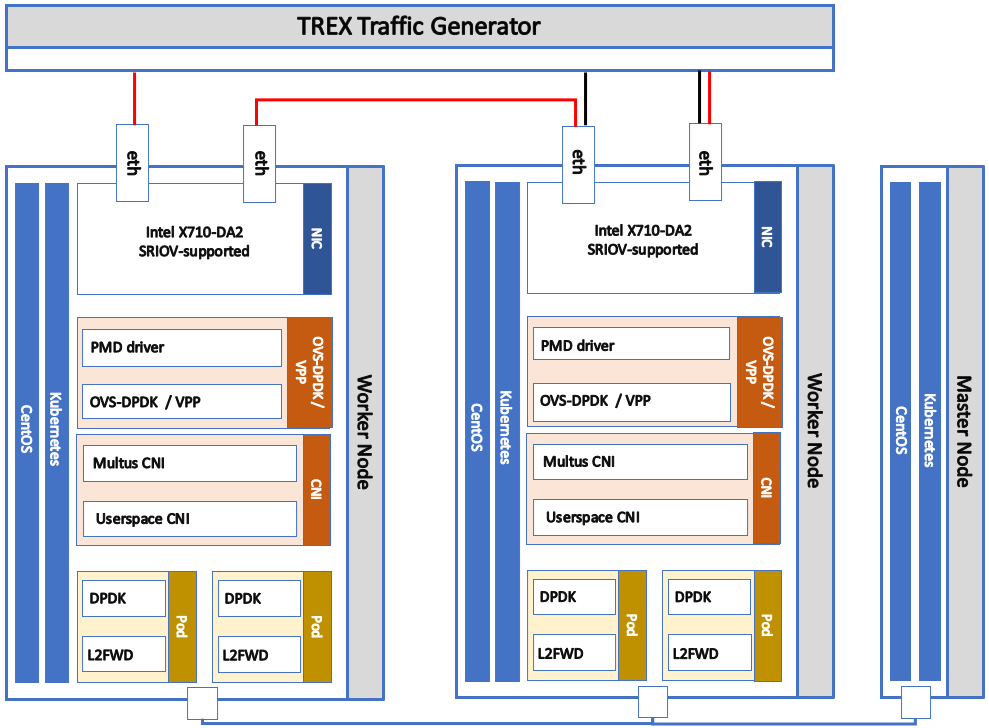
  Userspace ↔ Container

  vEth Pairs created by Cilium CNI



What got done

- Benchmarking Testbed – same with previous hackathons



Single node scenario

Multi nodes scenario

- eBPF Supported NIC: Intel X710
- AF-XDP supported kernel: Ubuntu 22.04 (kernel v5.15)
- Pod multi-interfaces: Multus
- vSwitch supported CNI: Userspace CNI
- CNDP-AFXDP k8S CNI
- Cilium CNI

What got done

- Benchmarking Configuration
- **Hardware – Worker Node**

CPU	Intel(R) Xeon(R) Gold 5220R CPU @ 2.20GHz 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> CPU C-state Disabled CPU P-state Disabled Intel(R) Hyper-Threading Tech Enabled Turbo Boost Disabled

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

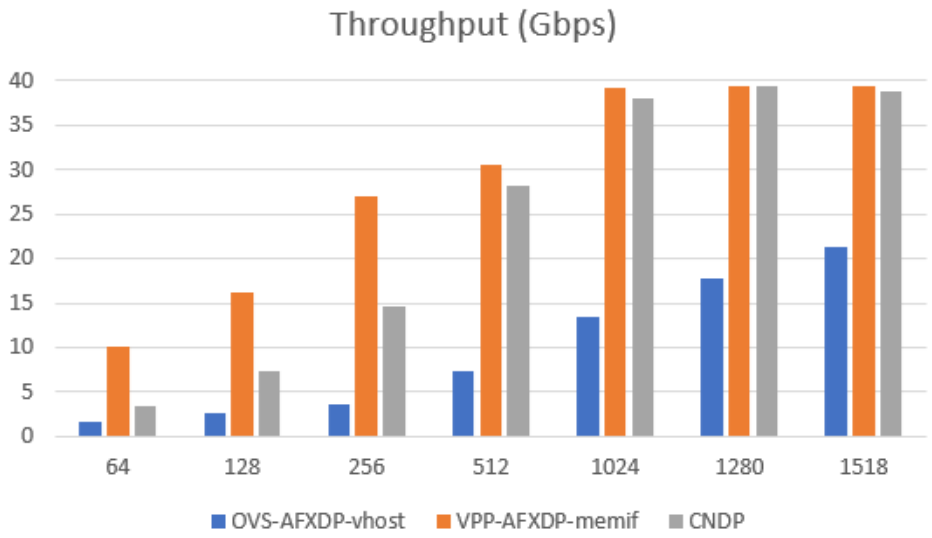
Name	T-Rex
Version	2.92
Benchmark method	T-Rex Non Drop Rate application (accepted percentage of drop rate is less than 0.1%)

- **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	21.11.1
Hugepages	1Gi

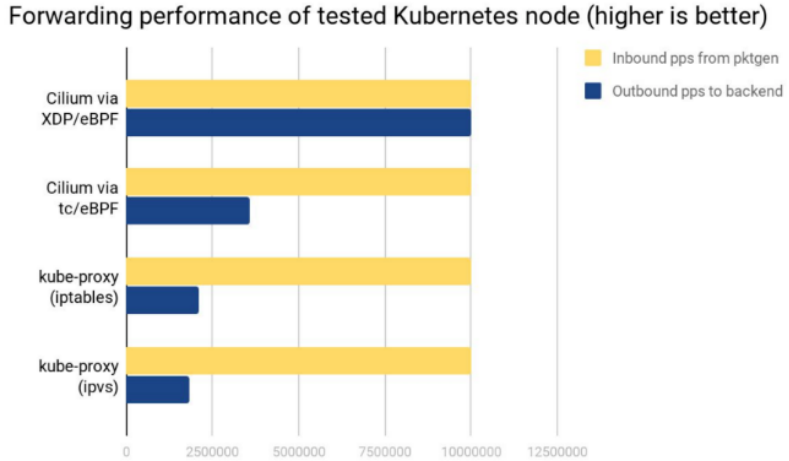
What we learned

- eBPF/AFXDP Benchmarking Performance Results (OVS-vhost vs VPP-memif vs CNDP)
  - VPP-memif outperforms OVS-vhost as expected
  - CNDP catchs up similar performance as VPP with higher size packets (>512)
  - The reason of OVS-vhost < VPP-memif is at the limitation of vhostuser-virtioPMD path between container and vSwitch
  - VPP uses memif PMD (shared memory packet interface) which is a better performance method
  - CNDP attachs NIC network devices to pod and poll packets directly from the AFXDP socket (no vSwitch) can also achieve high packet transmission performance as memif

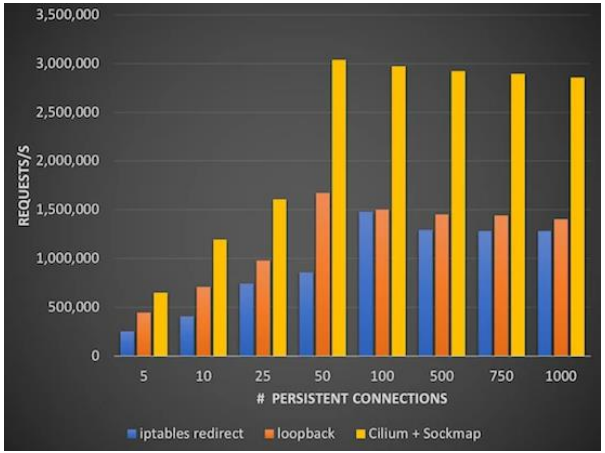


What we learned

- Cilium
  - Cilium acceleration performance has already been benchmarked by Cilium, and can be referred to
  - Cilium 1.8 Release Blog (https://cilium.io/blog/2020/06/22/cilium-18/)
    - (XDP acceleration performance NorthSouth traffic)
  - Istio 1.0: How Cilium enhances Istio with socket-aware BPF programs (https://cilium.io/blog/2018/08/07/istio-10-cilium/)
    - (eBPF Socket Layer acceleration performance – EastWest traffic)



North-South XDP acceleration vs kernel kube-proxy



East-West eBPF socket layer acceleration vs kernel kube-proxy

Future Works

- Finalize the draft and ask for reviews
- We would like to welcome any questions, comments and contributions to the draft to start the WG adoption process

Wrap Up

Team members:

Younghan Kim (SSU)

Minh Ngoc Tran(SSU)

Thanh Nguyen Nguyen (SSU)

Jangwon Lee (SSU)

Hokeun Lim (SSU)

Git repo:

https://github.com/SSU-DCN/bmwg-container-networking

Remote Hackathon from Seoul

Internet Infra System Technology Research Center – Soongsil University (IISTRC- SSU)