Empowering Edge Software Stack on Arm with High Performance Container Networking Technology

Trevor Tao, Song Zhu
Mar/2020
Agenda

• Introduction
• Arm Edge Software Stack Architecture
• High Performance CNIs (Calico, Cilium...) for Arm Edge Software Stack
• SRIOV and SmarNIC support for Container Networking
• Performance Tests
• Future Work (Provisional)
Introduction
Introduction

• The purpose of edge computing and MEC is to bring real-time, high-bandwidth, low-latency access to latency-dependent applications, distributed at the edge of the network. Arm CPUs are cost effective, consume low power and are customizable; making them a preferred choice for edge cloud service vendors.

• On the other side, a high performance, flexible and easy deployable container networking of edge software stack is the key to the success of using Arm platform on edge cloud since the widely used VNFs are deployed by container-based orchestration engine, such as Kubernetes, OpenShift or others.

Universal CPE example of Edge compute

Arm ecosystem leverage price, power, acceleration advantages
Arm Edge Reference Stack

- Heterogeneous Architecture
  - VM, container, bare metal
  - Servers and customized Edge platforms
  - Virtualized NFs and Physical NFs
  - Accelerator interface

- Resource constraints
  - Kubernetes
  - SDN Controller for K8s

- HW Accelerations
  - Integrated accelerators
  - PCIe/CCIX attached accelerator (Smart NICs...)

- Infra Orchestration and Installer
  - Kubernetes
  - Containerized Compass
  - Docker

- Networking Software
  - VPP, OVS
  - Linux System Networking
  - eBPF

- Controller
  - Flannel, Calico
  - Cilium
  - OVN-K8s

- Real Time Linux distribution
  - SR-IOV, DPDK

- Edge Servers or Networking Edge Platform
  - Acceleration
    - Integrated Accelerators
    - Smart NICs
    - FPGA/GPU

- Network Equipment
  - Switch/GW
Arm Edge Stack Deployment Reference Cluster

Small deployment with Marvell Mcbin DBShot Boards

Arm server samples used in edge cloud
High Performance Container Networking for Arm Edge Software Stack
High Performance CNIs available for Arm Edge Stack

Things now available in Akraino IEC Arm edge stack as a ref:

- **Calico**
  - pure IP networking fabric
  - high-level network policy management by iptables
  - Good scalability
  - Support direct(non-overlay) and overlay(IPINIP, VxLAN) network connection
  - Easy deployment
  - Calico-VPP appears

- **Cilium**
  - Linux-Native, API-Aware Networking and Security for Containers
  - Linux eBPF based network policy, load balance and security which is believed to be with incredible performance
  - Native VPP ACL/NAT based network policy and access
  - Good performance but with rather complex configuration
  - Good scalability too

- **Contiv-VPP**
  - uses FD.io VPP to provide network connectivity between PODs
  - Native DPDK interface support for phy NIC
  - Native VPP ACL/NAT based network policy and access
  - Good performance but with rather complex configuration
  - Hard to debug

- **OVN-K8s**
  - OVS/OVN-controller based K8s networking solution
  - Rather good performance with OVS inherited
  - Use OVN logical switches/routers to connect Pods and for outside access
  - No OVS-DPDK support now

- **SRIOV**
  - Direct physical interfaces(PF/VFs) support for Pods
  - High performance with direct Linux kernel eth driver or DPDK PMD driver
  - Usually co-work with other CNIs, such as Flannel, Calico by Multus or other glue CNI
  - Need resource description or annotation when do the configuration for CNI and Pod setup
  - Good scalability too
  - Uses FD.io VPP to provide network connectivity between PODs
  - Native DPDK interface support for phy NIC
  - Native VPP ACL/NAT based network policy and access
  - Good performance but with rather complex configuration
  - Hard to debug

- **Flannel**
  - Widely used and almost easiest deployment for a simple K8s networking
  - Linux network bridge for pod connection and overlay based communication for inter-hosts access
  - Easy to be integrated into other container networking solution, e.g., Cilium
  - No good network policy support

Repo: https://gerrit.akraino.org/r/admin/repos/iec
Calico on Kubernetes

1. Pure IP networking fabric
2. No encapsulation needed when simple L2 connection available for nodes
3. Easy deployment and debug
4. Supporting Kubernetes Network Policy by iptables
5. Good scalability with BGP based routing
Cilium – API Aware Networking

Kubernetes Services Implementation
- BPF-based
  - Per-CPU Hash table
- iptables kube-proxy
  - Linear List
  - All rules have to be replaced as a whole

API Aware Security
- Allow GET /jobs/* from identity foo
- GET /jobs/331
- GET /healthz
- GET /apiextensions/pods
- POST /jobs

Cilium: API Aware Networking & Network Security for Microservices using BPF & XDP
SRIOV and SmarNIC support for Container Networking
High Performance Networking with SmartNIC SRIOV Interfaces by SRIOV CNI

Device Resource Description:

apiVersion: v1
kind: ConfigMap
metadata:
  name: srivodp-config
  namespace: kube-system

data:
  config.json: |
  |
  "resourceList": [{
    "resourceName": "ps225_sriov_netdevice",
    "selectors": {
      "vendors": ["14e4"],
      "devices": ["d800"],
      "drivers": ["bnxt_en"],
      "pfNames": ["enp8s0f0np0"]
    },
    "pfNames": ["enp9s0np0"]
  }], ...
"

Uses only 1 of them

For DPDK device driver(another Intel x710 nic as a sample):

apiVersion: v1
kind: ConfigMap
metadata:
  name: sriovdp-config
  namespace: kube-system

data:
  config.json: |
  |
  "resourceList": [{
    "resourceName": "intel_sriov_dpdk",
    "selectors": {
      "vendors": ["8086"],
      "devices": ["154c"],
      "drivers": ["vfio-pci"],
      "pfNames": ["enp12s0f1"]
    }
  }]
"

VFs need to be created beforehand
Networking Model

Sample: 2 Pods Connected by Stingray PS225 VFs in a single host

Hos
t

Containers

Pod

optional

eth1

SR-IOV Plugin
VF0

enp9s0

SR-IOV Plugin
VF1

tenP8p1s0f2np0

Flannel/Calico bridge

Flannel/Calico bridge

NIC

SmartNIC PS225

PF0

PF0

NIC

A72 CPUs

© 2020 Arm Limited (or its affiliates)
A Very Raw Thinking on the Possible Model for Container Networking with SmartNIC OVS offload support

Model 1:
- Container
  - docker
  - veth
  - veth_p
- OVS-Bridge
- Partial OVS Flow Offload
- User Space
  - Kernel
- OVS/OVN Control
  - veth
  - Ovn-k8s-agent
  - Ovsdb-server
  - Ovs-vswitchd
- vSwitch Offload
- NIC

Model 2:
- Container
  - docker
  - vNIC
  - veth
- Kubernetes
  - Kubelet
  - Kube-ctrl-manager
  - ...
- OVN-Kube
  - Ovn-kube-controller
  - Ovn-k8s-agent-master
- User Space
  - Kernel
- OVS Flow Offload
- NIC

© 2020 Arm Limited (or its affiliates)
Performance of Calico and OVN-Kubernetes

(tested by iperf3)

In a single Arm server

- Calico Pod-to-Pod: 13.3 Gbps
- Calico Node-to-Pod: 15.8 Gbps
- OVN-K8s Pod-to-Pod: 11.4 Gbps
Performance Test with SRIOV CNI

Container Networking by SRIOV Interface for Intra Host Communication

Perf Test1: 2 Pods Connection with Stingray PS225 in the same host
Performance Test with SRIOV CNI

Container Networking by SRIOV Interface for Inter Host Communication

Perf Test2: 1 Pod Connected with another host via SmartNIC VF

Host1
Thunderx2-04

Host2
Thunderx2-02

SmartNIC Stingray PS225

A72 CPUs

© 2020 Arm Limited (or its affiliates)
Performance Test Result

Initial Test Result (To be tuned)

Note: Here we choose iperf instead of iperf3 due to its multi-threaded implementation

Pod Communication Performance with SRIOV CNI in the same host via PS225 VFs
Pod Communication Performance with SRIOV CNI in 2 hosts via PS225 VF (10G Connection)
Future Work (Provisional)
Future Work (Provisional)

- Performance evaluation and tuning for Calico, Cilium CNI with Pod-2-Pod, Node-2-Pod cases on arm platform, mainly for cross-node. The possible tuning parameters include:
  - RPS/RFS
  - Linux system netdev parameters
  - RX/TX queue
  - MTU
  - eBPF/XDP
  - Overlay/N-Overlay

- Service mesh integration with high performance CNIs, such as Cilium/Proxy
- Further DPDK incorporated container networking usage model and performance evaluation
- Further CI for high performance container networking solutions on arm with up-to-date processes
- More thinking and possible actual work on SmartNIC support for container networking with OVS offload