The Work of Containerized NFV Infrastructure on Arm Platform

Trevor Tao <trevor.tao@arm.com>
Agenda

• Background
• Data Plane Acceleration on Arm
• Advanced CNI plugins and Use Case for Container Networking
• Installer on Arm
• High Performance Networking with VPP on Arm
• OPNFV CI/CD on Arm
• Next Steps
Background
Linux Foundation Projects
Projects with Arm

Armband
- The purpose of this project is simply to integrate and test all aspects of OPNFV releases on ARM-based servers.

Yardstick
- A test framework with test cases and test stimuli to enable NFV-I performance verification

Auto
- This project focuses on ONAP component integration and verification with OPNFV reference platforms/scenarios

Compass4nfv
- An installer project based on open source project Compass, which provides automated deployment and management of OpenStack and other distributed systems
- Ansible is used by default.
- Our main installer for OPNFV Container4NFV project

Container4NFV
- Provide a container full-stack environment where VNF can run, including data plane VNF and control plane VNF. Let the platform support container and virtualization technology. Collect requirement for containerized NFVs.
- Previously named as OpenRetriever
- What are we focusing on for building Arm’s containerized NFV infrastructure now

![Diagram of NFV Architecture](attachment:Arm-NFV-Diagram.png)
This project focuses on **ONAP component integration and verification with OPNFV reference platforms/scenarios**, through primarily a post-install process in order to avoid impact to OPNFV installer projects.

Related Project Opera: developing OPNFV-installer supported scenarios that can deploy and verify ONAP as a whole.
Validate **ONAP** (Open Network Automation Platform) as NFV Orchestrator and VNF Manager in OPNFV ecosystem; Auto project [home page](#)

Show added value of:

- Automation using closed loops (defined in CLAMP), policies (defined in Policy Framework), and DCAE (real-time monitoring, execution of closed loops and policies; also alarm correlation)

- Design-time portal-based (as well as API-based control) **streamlined VNF lifecycle management**: Onboarding (with SDC, to define VSPs with VLMs, and end-to-end Services), Deployment (with VID and MSO), and Operations (with persistent inventory data in AAI)
Data Plane Acceleration
DPDK (Data Plane Development Kit)

• A Project at The Linux Foundation
  • A set of libraries and drivers for fast packet processing
  • The first supported CPU was Intel x86 and it is now extended to Arm and IBM POWER.
• Runs mostly in Linux userland
  • Kernel bypass
  • Zero copy
  • Huge pages
  • Polling/Event driven
  • Enable accelerators
  • Batch packet handling
  • Lockless synchronization
  • NUMA awareness
DPDK on Arm

- Multiple active members on Arm platforms
- Arm platform porting & optimization
- DPDK performance on Arm platforms
  - Throughput/Latency perf test on Arm platforms
  - x86 platform for reference & cross check
- NEW features development
- Functional verification / enablement
  - DTS (DPDK Test Suite) test in bare-metal & virtualization
  - Several platforms / NICs selected
- Internal CI setup
- Use cases setup
  - DPDK in container, Nginx with DPDK + mTCP, etc.
L2FWD performance

• Hardware
  • ARMv8 64-bit, 2.4Ghz
    – 1-15 cores isolated
    – 1G Hugepage size
  • Intel 2-ports 82599ES 10Gbe
  • IXIA traffic generator

• Software
  • Debian Linaro ERP 17.08
    – 4.12 kernel
  • DPDK 17.08
Advanced CNI Plugins and Use Case
OPNFV Container4NFV Project

- Containerized VNFs on Arm NFV infrastructure
- Kubernetes as Virtual Infrastructure Manager (VIM)
- Multiple networking interfaces support in container
- VPP for switching and routing
- DPDK vhost-user/virtio_user for container networking
Container-based NFV Ecosystem on Arm

ONAP supports multiple VNF environments by integrating with multiple VIMs, VNFMs, SDN Controllers, and even legacy equipment.

Kubernetes as COE

Multus plugin for Kubernetes as CNI

Flannel/DPDK/Vhost user CNI plugins integrated

Ref: https://wiki.opnfv.org/pages/viewpage.action?spaceKey=OpenRetriever&title=Container%27s+Architecture+for+Cloud+Native+NFV
Multus CNI with 2 Flannels

Description submitted to Container4NFV:
https://github.com/opnfv/container4nfv/blob/master/docs/arm/multi_flannel_intfs_deployment.rst
Container Networking Acceleration with DPDK

- K8s MASTER
  - API Server
  - Scheduler
  - Controller Manager

- etcd

- NODE1
- NODE2
- NODE3

- PF
- VF
- VF

- Flannel CNI
- SR-IOV CNI
- Multus CNI

- KUBELET
- KUBE-PROXY

- Kernel
  - docker0 bridge
  - Flannel0 bridge

- Containers
  - Docker
  - VNF

- VFIO/UIO
  - Docker
  - VNF
  - DPK

- arm
Use Case – Deliver Network Services with Kubernetes

Enterprise Branch Connects to Internet

vCPE: Containerized OpenWRT

Service Provider’s Data Center using standard ARM Servers
Installer on Arm
Project Compass4nfv for Kubernetes -- Installer now Arm Support

Compass4NFV on Arm (Yibo Cai, Di Xu):

What We Have Done:

- Ported Compass4NFV docker images for AArch64 and uploaded to dockerhub Linaro repo.
  - Compass-tasks-k8s
  - Compass-deck
  - Compass-mq
  - Compass-cobbler
  - Compass-db
- Supported AArch64 baremetal deployment (CentOS7, Ubuntu16.04) for Compass4NFV project.
- Supported deploying Kubernetes (1.9.x) cluster on AArch64 virtual and bare-metal nodes.

Compass4NFV repo

Our ‘F’ release scenarios for Container4NFV would be based on the work in Compass4NFV
What we have done

• Enabled containerized VNFs on ARM NFV-I
• Used Kubernetes as Virtual Infrastructure Manager (VIM)
• Build VNFs Container Image for vCPE use case
• Provide 3 k8s deployment scenarios and docs on arm in Container4nfv
  • **K8s basic deployment**
  • **K8s deployment with 2 Flannel interfaces**
  • **K8s sriov cni deployment**
  • documentation for kubernetes data plane with dpdk acceleration on Arm64 platform
  • documentation for sriov cni with pf mode
  • 2 Flannel interfaces installation document for arm platform
• Enabled 2 Kubernetes scenario CI jobs in ‘F’ release:
  • **k8-multus-nofeature-noha with Compass**
  • **k8-sriov-cni-nofeature-noha with Compass**
• Enabled Yardstick performance tests for containerized VNFs on Arm NFV-I
What we have done – cont.

- **Enabled Yardstick CI job in ‘F’ release**
  - Still under improvement

- **Presentations:**
  - [Containerized VNFs with Data Plane Acceleration On Arm platform](#)
  - [Performance Evaluation for Containerized NFV-I on Arm via OPNFV Yardstick](#)

- **Other Contributions to OSS community:**
  - [SR-IOV CNI: Assign pf directly if we don’t need vf](#)
  - Yardstick: [bug fixing](#), function enhancements 1, 2
  - [Arm Containerized NFV Infrastructure description for Container4NFV project](#)
  - [Multiple Flannel Interfaces Deployment Doc for Kubernetes Pod on Arm server](#)
High Performance Networking with VPP on Arm
FD.io/VPP (Vector Packet Processing)

- A Project at The Linux Foundation
  - Multi-party
  - Multi-project
- A Software Dataplane
  - Virtual switching/Routing
  - High throughput
  - Low Latency
  - Feature Rich
  - Resource Efficient
  - Bare Metal/VM/Container
  - Multiplatform

FD.io Scope

- Bare Metal/VM/Container
- Dataplane Management Agent
- Control Plane
- Packet Processing
- Classify/Transform/Prioritize/Forward/Terminate
- Network IO
- NIC/vNIC <-> cores/threads
FD.io/VPP on Arm

• Lead/guide the Arm community
  • Collaboration between Arm partners on VPP
  • Set up VPP/AArch64 wiki page on FD.io for collaboration

• Enable VPP on Arm: 3-step strategy
  • Fix build, unit test, and packaging issues
  • Integrate Arm platforms into upstream CSIT (continuous integration)
  • Performance benchmarking and tuning

• Arm code implementation
  • CPU capability identification
  • Support for 64B and 128B cache line sizes
  • NEON enablement
  • Packet prefetch tuning
Why Use VPP for Container Networking

• Container networking requirements for NFV
  • High performance on packet processing
  • High scalability
  • High flexibility

What VPP provides

• High performance
• Abundant interfaces: ssvm, virtio/vhost, af_packet, tap, memif...
• Abundant features for control and management
How to Support DPDK APP in Container with VPP Efficiently and Dynamically

• Virtio/Vhost-user is a para-virtualization interface which could achieve quite promising performance compared with other native networking interface supported by VPP, such as af_packet

• DPDK’s Vhost user for VPP is simple and easy use,
  • set dpdk’ section as “vdev eth_vhost0, iface=/tmp/sock0.sock” statically.

• Static DPDK vhost-user interface cannot support container orchestration engine, e.g., Kubernetes, dynamically

• However, DPDK Virtio Driver Works with VPP native Vhost-user Driver which could be created on demand:
  • vpp# create vhost-user socket socket-file server
Vhost-user CNI for Kubernetes

K8S POD

- Vhost-user server socket(interface) is created in VPP
- After adding the vhost user CNI path, the virtio-user interface is used as a virtual device of DPDK
Contiv/VPP on Arm is Ongoing

- Virtio/Vhost-user Support with Multiple CNIs on arm64

Ref:
- https://ligato.io/mydoc_container_net_vpp.html
- https://schd.ws/hosted_files/fdiominisummitkubeconeu20/8c/Ligato-Kubecon-DK.pdf
OPNFV CI/CD for Arm
Container4NFV CI/CD in OPNFV Pharos Testbed

Pod src:
- Huawei
- Packet.net
- Arm Internal
  (University in US)

Yardstick

Jump Server

VPN Gateway Router

OPNFV

Internet

Firewall

VPN
OPNFV ‘F’ Release Scenarios for Arm

Fraser Scenario Status:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Installer</th>
<th>Owner</th>
<th>Jenkins Job Created</th>
<th>Intent to release 6.0 (Y/N)</th>
<th>Intent to release 6.1 (Y/N)</th>
<th>Intent to release 6.2 (Y/N)</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>k8-multus-nofeature-noha</td>
<td>Compass</td>
<td>@Trevor Tao</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>k8-sriov-cni-nofeature-noha</td>
<td>Compass</td>
<td>@Phoenix Striker</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
</tbody>
</table>

• K8-multus-nofeature-noha
  k8-multus-nofeature-noha with Compass
• K8-sriov-cni-nofeature-noha-arm
  k8-sriov-cni-nofeature-noha with Compass

Contributor: Yibo Cai, Di Xu @arm
Next Steps
Next Steps (provisional)

• Continue VPP enablement and performance tuning on Arm servers
• Performance benchmarking on Arm servers
• VPP integration (CI/CD enablement) in OPNFV Gambia release (Nov 2018)
• vCPE Use Case and more scenarios into Project Container4NFV
• Enable Contiv/VPP based container networking solutions
• Containerized NFVi for MEC(Multi-access Edge Computing)
• Integrate VPP-based NFV solutions with orchestration software (ONAP)
Thank You!
Danke!
Merci!
谢谢!
ありがとう!
Gracias!
Kiitos!
감사합니다
धन्यवाद