Transparent Container Solution for DPDK Applications

TANYA BROKHMAN
SHAHAR BELKAR
Agenda

- Introduction to containers
  - Overview and background
  - Docker Containers Networking Model (CNM)
  - Existing solutions for DPDK
- Extending CNM capabilities
  - Container networking over SRIOV
  - DPDK networking for containers
- Evaluation
- Conclusions and future work
Introduction & Market Overview
**Disadvantages of the Classical approach:**

- Network functions are based on specific HW&SW
- Physical install per appliance per site

---

**Advantages of the Virtualized approach:**

- Network functions are based SW-based
- Multiple roles over same hardware
Application Evolution

- Increase in Complexity of interactions
- Information grows exponentially
- Compounded by Web-Based communications
- SOA/Web Services bring exponential increase in interactions

- Distributed applications introduces Security, Management Issues
- Balancing flexibility with control is becoming more difficult
- Latency became important
Application Shift to Microservices

- Application Design is moving to Microservices:
  - Loosely couples distributed application
  - Private datastore
  - Owned by different teams

We need a packaging unit to deploy a Microservice…
Is there an alternative?

Disadvantages of VMs:
- Additional memory and storage footprint
- Performance overheads due to additional SW layers
- Can’t be easily migrated
- Slow spinup time
- Low density
Containers are Better Match for Microservices

**Container** is a standard unit of software that packages up code and all of its dependencies.

- Low footprint
- Easy to deploy
- Isolate software from its environment
- Portable across hybrid infrastructures
- Fast spin-up
- High density

Deploy with Containers
451 Research Says Application Containers Market Will Grow to Reach $4.3bn by 2022

New York, December 5, 2018 – The latest applications container market research from 451 Research, leading technology analyst and advisory firm, indicates this market will continue to expand and be worth more than $2.1 billion in 2019 and more than $4.3 billion in 2022 – a compound annual growth rate (CAGR) of 30%, according to the latest Market Monitor Cloud-Enabling Technologies – Application Containers study.

Application Containers: Total Market Revenue ($M)

- **30.8% 2017-22 CAGR**
- **2017: $1,124**
- **2018: $1,567**
- **2019: $2,126**
- **2020: $2,755**
- **2021: $3,467**
- **2022: $4,311**

184 vendors included in analysis with individual estimates and forecasts

World Trend – Shifting to Containers

Organization’s adoption plans for containers (e.g., Docker)

- **In use**: 19%
- **In discovery/proof of concept**: 19%
- **Plan to implement in next 12 months**: 7%
- **Plan to implement in next 24 months**: 3%
- **Considering, but no current plan to implement**: 21%
- **Not in-use/not in plan**: 31%

% of respondents (n=426)

Q. Please indicate your organization’s adoption plans for containers (e.g., Docker).

Source: 451 Research, Voice of the Enterprise: Servers and Converged Infrastructure, Workloads and Key Projects 2018
Microsoft plans to rearchitect Azure Stack by making it container-based

Introducing Cloud Native Networking for Amazon ECS Containers

Microsoft Azure Catches Up With Container Networking

SANTA CLARA, Calif. -- The Microsoft Azure cloud is adding container support to its networking, combining container and virtual machine (VM) into what the company is calling a “single-click” model for software-defined networking (SDN).
Next Wave of Service Deployment

Running within the container == Running directly on host/VM

How we achieve that?

Containers Direct!
Docker & CNM
What is Docker?

Docker is platform for developers and sysadmins to develop, deploy and run applications with containers

- Flexibility
- Scalability
- Cross Platform
- Decentralized
- User Friendly
- Support
Docker Networking is a communication channel between the independent containers
**Container Networking Model (CNM)**

**Container Networking Model** formalizes the steps required to provide networking for containers, while providing an abstraction that can be used to support multiple network drivers.

Architecture of container Networking model
Container Network Model Objects

**Driver**
Provides an actual implementation of the network

**Endpoint**
Provides the connectivity for services exposed by a container network

**Network Controller**
Provides an entry point into the libnetwork

**Network**
An implementation of CNM:Network

**Sandbox**
Represents container’s network configuration
CNM Lifecycle

- Drivers register with Network controller
- Network created by the controller and bound to driver
- Driver create endpoints for the network and configures it
- The container will attach to the endpoint and a sandbox will be created for it.
CNM: Available Network Drivers

Container Network Model (libnetwork)

- Bridge
- Host
- Overlay
- MACVLAN

Native Drivers

3rd Party plugin

Remote Drivers

Docker Engine
Existing Solutions for DPDK Container Networking

- Based on virtio/vhost with virtual switch
  - vhost consumes a lot of CPU cycles
  - Redundant data copies between host and container

- All examples target connectivity between containers running on same host
Container Direct

~ZERO PERFORMANCE IMPACT WHEN RUNNING WITHIN A CONTAINER
Container Networking over SRIOV

- Leverage HW mechanisms for security and isolation
- Dedicated NIC Tx and Rx queues per container
  - No contention with other containers!
- Native access to PCIe based networking devices with no overheads
- No redundant SW layer copies!
hisriov Driver – SW Components

- **go-plugin-helpers/network**
  - Go handler to create external network extensions for Docker

- **netlink**
  - An interface to the Linux kernel for network manipulations

- **sriovnet**
  - Go library for configuring SRIOV networking devices

- **hilogger**
  - Logging framework

Legend:
- **New component added and described by this document**
- **Existing package/library**
Usage Example – Create a New Network

$ docker network create -d hisriov -subnet=40.0.0.0/24 -o netdevice=eth2 -o nvf=5 mynet

1. Enable SRIOV
2. Enable and configure 5 VFs
3. Save new network configuration
DPDK Networking for Containers

- Enables SRIOV based DPDK networking for containers
- Allocate a VF(s) for the container
- Handle DPDK setup/teardown transparently to the user
- Works on top of a hisriov managed network

`docker_run_dpdk` is a Cobra based utility to enable running DPDK applications inside a container

(*) Cobra library provides us with a simple interface for creating new applications
docker_run_dpdk Utility – SW Components

- **Cobra library**
  - Create a new CLI interface
- **dpdkmap**
  - Handles bind/unbind to/from DPDK
  - Utilizes *dpdk-devbind.py* script from the DPDK build
- **hilogger**
- **sriovnet**
Usage Example – Start a DPDK Bound Container

$ docker_run_dpdk run --net=mynet --name=dpdkC --numVf=2 -it centos-hiofed:latest bash

1. Bind allocated VF to DPDK
2. Generate a new docker run command exposing UIO char device of the VF
3. Save container details in local DB
Usage Example – Stop a DPDK Bound Container

$ docker_run_dpdk stop --name=dpdkC

1. Load container details from local DB
2. Unbind container interfaces from DPDK
3. Stop the container
Evaluation & Wrap up

LET'S SEE SOME NUMBERS, AND NICE GRAPHS…
Performance Evaluation – Container Direct

Our setup:

- IXIA as packet generator
- L2FW application from DPDK examples

Less than 2% bandwidth degradation over the container
Performance Evaluation – Container Direct

Less than 3% latency overhead added by the container
Conclusions and Future Work

- The containers world trend is here and DPDK has to adapt fast
  - DPDK needs to shift towards containerized environment

- Container Direct breaks the performance limitations of the current container networking approach, with full application transparency
  - We don’t have to compromise on performance when moving from VMs to containers

- More work is required with Telco-Cloud to adapt the solution for their needs
  - Smoother VNF -> CNF transition
Thank You!

Tanya Brokhman
Shahar Belkar