R4 Test Document of IEC Type 5: SmartNIC for Integrated Edge Cloud (IEC) Blueprint Family

Introduction

The R4 release will evaluate the conntrack performance of Open vSwitch (OVS).

A DPDK based Open vSwitch (OVS-DPDK) is used as the virtual switch, and the network traffic is virtualized with the VXLAN encapsulation.

We will compare the performance of an optimized software based OVS with the performance of the OVS running on a MLNX SmartNIC with hardware offload enabled.

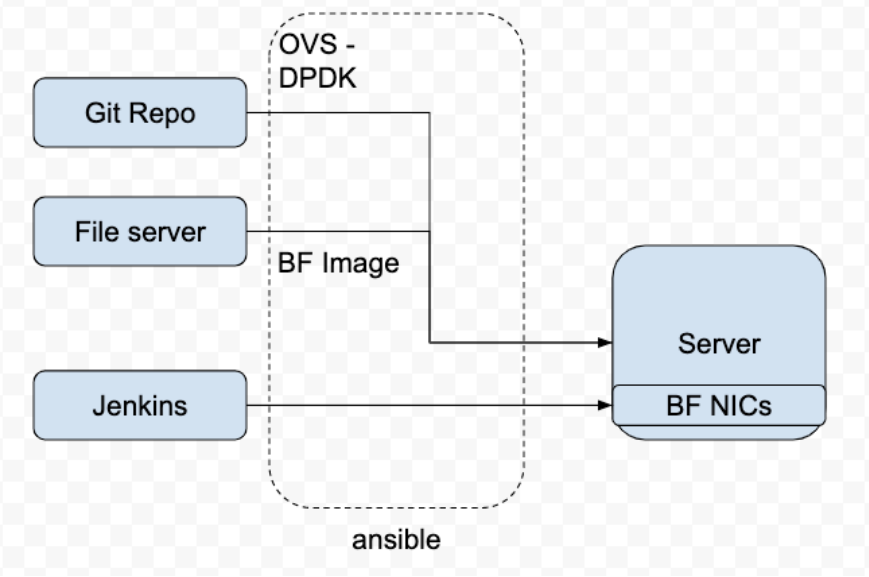
Akraino Test Group Information

SmartNIC deployed Architecture

We reuse the test architecture for smart NIC in R3 release. The below description is the same with the R3 release test documents.

To deploy the Test architecture for smartNIC, we use a private Jenkins and a server equipped with a BlueField v2 SmartNIC.

We use Ansible to automatically setup the filesystem image and install the OVS-DPDK in the SmartNICs.



The File Server is a simple Nginx based web server where stores the BF drivers, FS image.

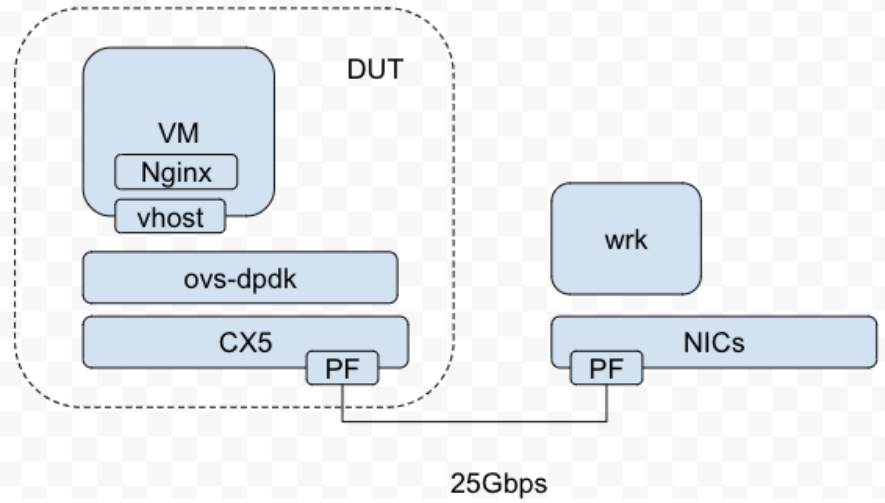
The Git repo is our own git repo where hosts OVS-DPDK and DPDK code.

The Jenkins will use ansible plugin to download BF drivers and FS image in the test server and setup the environment

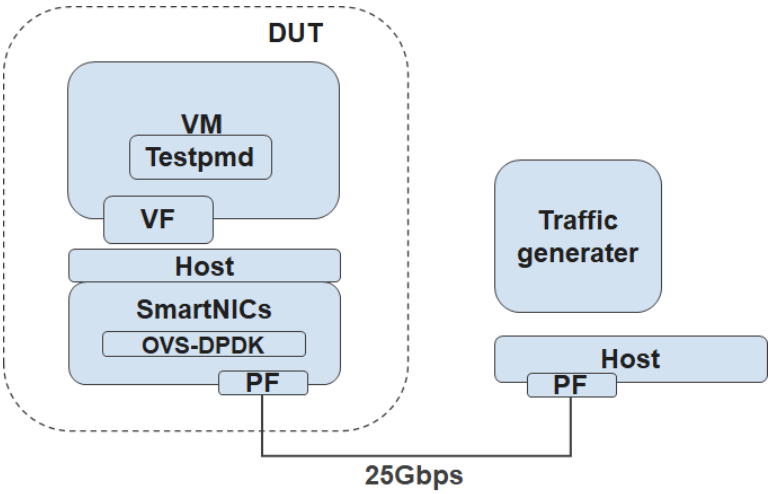
according to the ansible-playbook.

Image	download link
BlueField-3.5.0.11563.tar.xz	http://www.mellanox.com/downloads/BlueField/BlueField-3.5.0.11563/BlueField-3.5.0.11563.tar.xz
core-image-full-BlueField-3.5.0.11563.2.7.4.tar.xz	http://www.mellanox.com/downloads/BlueField/BlueField-3.5.0.11563/core-image-full-BlueField-3.5.0.11563.2.7.4.tar.xz
mft-4.16.0-105-x86_64-rpm.tgz	https://www.mellanox.com/downloads/MFT/mft-4.16.0-105-x86_64-rpm.tgz
MLNX_OFED_LINUX-5.2-1.0.4.0-rhel7.6-x86_64.tgz	http://www.mellanox.com/page/mlnx_ofed_eula?mtag=linux_sw_drivers&mrequest=downloads&mtype=ofed&mver=MLNX_OFED-5.2-1.0.4.0&mname=MLNX_OFED_LINUX-5.2-1.0.4.0-rhel7.6-x86_64.tgz

OVS-DPDK Test Architecture



OVS-DPDK on BlueField Test Architecture



The testbed setup is shown in the above diagram. DUT stands for Device Under Test

Test Framework

The software used and the OVS-DPDK setup is shown below.

Type	Description
SmartNICs	BlueField v2, 25Gbps
DPDK	version 20.11
vSwitch	OVS-DPDK 2.12 with VXLAN DECAP/ENCAP offload enabled. https://github.com/bytedance/ovs-dpdk

```
        Bridge br-phy0
        fail_mode: standalone
        datapath_type: netdev
        Port br-phy0
            Interface br-phy0
                type: internal
        Port p0
            Interface p0
                type: dpdk
                options: {dpdk-devargs="0000:03:00.0"}
    Bridge br-int0
        fail_mode: standalone
        datapath_type: netdev
        Port br-int0
            Interface br-int0
                type: internal
        Port vxlan0
            Interface vxlan0
                type: vxlan
                options: {flags="0", key="101", local_ip="1.1.1.1", remote_ip="1.1.1.3"}
        Port pf0vf0
            Interface pf0vf0
                type: dpdk
                options: {dpdk-devargs="0000:03:00.0,representor=[0]"}
    ovs_version: "2.14.1"
```

```
root:/home/ovs-dpdk# ovs-vsctl --format=csv --data=bare --no-headings --column=other_config list open_vswitch
"dpdk-extra=-w [PCIE]  -l 70 dpdk-init=true dpdk-socket-mem=2048,2048 emc-insert-inv-prob=0 n-handler-threads=1
n-revalidator-threads=4 neigh-notifier-enable=true pmd-cpu-mask=0xc00000000000c00000 pmd-pause=false pmd-rxq-
assign=roundrobin smc-enable=true tx-flush-interval=0 userspace-tso-enable=true"
```

```
root:/home/ovs-dpdk# ovs-vsctl --format=csv --data=bare --no-headings --column=other_config list open_vswitch
dpdk-extra="-w 0000:03:00.0,representor=[0-1],dv_xmeta_en=1,sys_mem_en=1", dpdk-init="true", dpdk-socket-mem="
4096", hw-offload="true", max-idle="120000"
```

Traffic Generator

We will use DPDK pktgen as the Traffic Generator.

CT Ruleset

```
Br-ex rules
table=0 priority=300 in_port=eth2,tcp,tp_dst=32768/0x8000 actions=output:br-int-patch
table=0 priority=300 in_port=eth2,udp,tp_dst=32768/0x8000 actions=output:br-int-patch
table=0 priority=200 in_port=eth2 actions=output:LOCAL
table=0 priority=200 in_port=LOCAL actions=eth2
table=0 priority=200 in_port=br-int-patch actions=eth2
```

```
Br-int rules
table=0,arp,action=normal
table=1,priority=1,ip,ct_state=+trk+new,action=ct(commit),normal
table=0,ip,ct_state=-trk,action=ct(table=1)
table=1,priority=1,ip,ct_state=+trk+est,action=normal
```

Nginx configuration

```
user  nginx;
worker_processes  auto;
error_log  /var/log/nginx/error.log warn;
pid        /var/run/nginx.pid;
events {
    worker_connections  2000000;
}
http {
    include       /etc/nginx/mime.types;
    default_type  application/octet-stream;
    log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
                  '$status $body_bytes_sent "$http_referer" '
                  '"$http_user_agent" "$http_x_forwarded_for"';
    access_log off;
    sendfile        on;
    keepalive_timeout  65;
    include /etc/nginx/conf.d/*.conf;
}
```

```
server {
listen 80 backlog=8192 reuseport;
server_name localhost;

location / {
return 200 "hello";
}

error_page 500 502 503 504 /50x.html;
location = /50x.html {
root /usr/share/nginx/html;
}
}
```

WRK configuration

connection setup rates:

```
./wrk -t 32 -c 640 -d30s http://10.0.1.127/ -H "Connection: Close"
```

Performance Results

For the optimized software-based OVS, we tested on a 48C24q VM, running NGINX as a server, CT is enabled on OVS-DPDK.(upstream version / our version)

We also run 50 2c2q VMs as clients, generating HTTP GET traffic to evaluate the performance.

pps (not closing connection after each query)	pps (closing connection after each query)	connection initial rates (closing connection after each query)	QPS (not closing connection after each query)
1.66Mpps/2Mpps	1.66Mpps/2Mpps	140Kcps/200Kcps	889Kqps/1.14Mqps

For the OVS-DPDK running on SmartNIC with CT function enabled, we tested on 17C8G VM running testpmd (where 1 core for lcore and 16 cores for nb cores) as the traffic forwarder and reached the performance below.

Note: the result also depends on the traffic generator side.

Frame size	114 bytes
Packets per second	~23Mpps

The test is to evaluate the performance of SmartNIC offloading.
Test API description

Thus we currently don't have any Test APIs provided.

Test Dashboards

Functional Tests

Open vSwitch itself contains a test suite for functional test, the link is http://docs.openvswitch.org/en/latest/topics/testing/

We have run the basic test suite according to the link.

By running,

```
make check TESTSUITEFLAGS=-j8
```

We got the below results.

Total Tests	Test Executed	Pass	Fail	In Progress
2225	2200	2198	2	0

The two failed cases are about sFlow sampling. We are investigating the internal reason.

25 cases are skipped due to the configuration.

Additional Testing

n/a

Bottlenecks/Errata

n/a