VSPERF User Guide

Release brahmaputra.1.0 (a481bc5)

OPNFV

February 24, 2016

1 vSwitchPerf test suites userguide **1**
1.1 General . . . 1
1.2 VSPERF Installation . . . 1
1.3 Traffic Generator Setup . . . 1
1.4 Cloning and building src dependencies . . . 1
1.5 Configure the `./conf/10_custom.conf` file . . . 2
1.6 Using a custom settings file . . . 2
1.7 vloop_vnf . . . 2
1.8 l2fwd Kernel Module . . . 2
1.9 Executing tests . . . 3
1.10 Executing Vanilla OVS tests . . . 3
1.11 Executing PVP and PVVP tests . . . 4
1.12 Executing PVP tests using Vanilla OVS . . . 4
1.13 Selection of loopback application for PVP and PVVP tests . . . 5
1.14 Code change verification by pylint . . . 5
1.15 GOTCHAs: . . . 5
1.16 More information . . . 6

CHAPTER

ONE

VSWITCHPERF TEST SUITES USERGUIDE

1.1 General

VSPERF requires a traffic generators to run tests, automated traffic gen support in VSPERF includes:

- IXIA traffic generator (IxNetwork hardware) and a machine that runs the IXIA client software.
- Spirent traffic generator (TestCenter hardware chassis or TestCenter virtual in a VM) and a VM to run the Spirent Virtual Deployment Service image, formerly known as "Spirent LabServer".

If you want to use another traffic generator, please select the Dummy generator option as shown in Traffic generator instructions

1.2 VSPERF Installation

To see the supported Operating Systems, vSwitches and system requirements, please follow the installation instructions to install.

1.3 Traffic Generator Setup

Follow the Traffic generator instructions to install and configure a suitable traffic generator.

1.4 Cloning and building src dependencies

In order to run VSPERF, you will need to download DPDK and OVS. You can do this manually and build them in a preferred location, OR you could use vswitchperf/src. The vswitchperf/src directory contains makefiles that will allow you to clone and build the libraries that VSPERF depends on, such as DPDK and OVS. To clone and build simply:

```
$ cd src
$ make
```

VSPERF can be used with stock OVS (without DPDK support). When build is finished, the libraries are stored in src_vanilla directory.

The 'make' builds all options in src:

- Vanilla OVS
- OVS with vhost_user as the guest access method (with DPDK support)

- OVS with vhost_cuse s the guest access method (with DPDK support)

The vhost_user build will reside in src/ovs/ The vhost_cuse build will reside in vswitchperf/src_cuse The Vanilla OVS build will reside in vswitchperf/src_vanilla

To delete a src subdirectory and its contents to allow you to re-clone simply use:

```
$ make clobber
```

1.5 Configure the `./conf/10_custom.conf` file

The `10_custom.conf` file is the configuration file that overrides default configurations in all the other configuration files in `./conf` The supplied `10_custom.conf` file **MUST** be modified, as it contains configuration items for which there are no reasonable default values.

The configuration items that can be added is not limited to the initial contents. Any configuration item mentioned in any .conf file in `./conf` directory can be added and that item will be overridden by the custom configuration value.

1.6 Using a custom settings file

If your `10_custom.conf` doesn't reside in the `./conf` directory of if you want to use an alternative configuration file, the file can be passed to `vsperf` via the `--conf-file` argument.

```
$ ./vsperf --conf-file <path_to_custom_conf> ...
```

Note that configuration passed in via the environment (`--load-env`) or via another command line argument will override both the default and your custom configuration files. This "priority hierarchy" can be described like so (1 = max priority):

1. Command line arguments
2. Environment variables
3. Configuration file(s)

1.7 vloop_vnf

vsperf uses a VM called vloop_vnf for looping traffic in the PVP and PVVP deployment scenarios. The image can be downloaded from http://artifacts.opnfv.org/.

```
$ wget http://artifacts.opnfv.org/vswitchperf/vloop-vnf-ubuntu-14.04_20151216.qcow2
```

vloop_vnf forwards traffic through a VM using one of: * DPDK testpmd * Linux Bridge * l2fwd kernel Module.

Alternatively you can use your own QEMU image.

1.8 l2fwd Kernel Module

A Kernel Module that provides OSI Layer 2 Ipv4 termination or forwarding with support for Destination Network Address Translation (DNAT) for both the MAC and IP addresses. l2fwd can be found in <vswitchperf_dir>/src/l2fwd

1.9 Executing tests

Before running any tests make sure you have root permissions by adding the following line to /etc/sudoers:

```
username ALL=(ALL)       NOPASSWD: ALL
```

username in the example above should be replaced with a real username.

To list the available tests:

```
$ ./vsperf --list
```

To run a single test:

```
$ ./vsperf $TESTNAME
```

Where $TESTNAME is the name of the vsperf test you would like to run.

To run a group of tests, for example all tests with a name containing 'RFC2544':

```
$ ./vsperf --conf-file=<path_to_custom_conf>/10_custom.conf --tests="RFC2544"
```

To run all tests:

```
$ ./vsperf --conf-file=<path_to_custom_conf>/10_custom.conf
```

Some tests allow for configurable parameters, including test duration (in seconds) as well as packet sizes (in bytes).

```
$ ./vsperf --conf-file user_settings.py
    --tests RFC2544Tput
    --test-param "duration=10;pkt_sizes=128"
```

For all available options, check out the help dialog:

```
$ ./vsperf --help
```

1.10 Executing Vanilla OVS tests

1. If needed, recompile src for all OVS variants

```
$ cd src
$ make distclean
$ make
```

2. Update your "10_custom.conf" file to use the appropriate variables for Vanilla OVS:

```
VSWITCH = 'OvsVanilla'
VSWITCH_VANILLA_PHY_PORT_NAMES = ['$PORT1', '$PORT1']
```

Where $PORT1 and $PORT2 are the Linux interfaces you'd like to bind to the vswitch.

3. Run test:

```
$ ./vsperf --conf-file=<path_to_custom_conf>
```

Please note if you don't want to configure Vanilla OVS through the configuration file, you can pass it as a CLI argument; BUT you must set the ports.

```
$ ./vsperf --vswitch OvsVanilla
```

1.11 Executing PVP and PVVP tests

To run tests using vhost-user as guest access method:

1. Set VHOST_METHOD and VNF of your settings file to:

```
VHOST_METHOD='user'
VNF = 'QemuDpdkVhost'
```

2. If needed, recompile src for all OVS variants

```
$ cd src
$ make distclean
$ make
```

3. Run test:

```
$ ./vsperf --conf-file=<path_to_custom_conf>/10_custom.conf
```

To run tests using vhost-cuse as guest access method:

1. Set VHOST_METHOD and VNF of your settings file to:

```
VHOST_METHOD='cuse'
VNF = 'QemuDpdkVhostCuse'
```

2. If needed, recompile src for all OVS variants

```
$ cd src
$ make distclean
$ make
```

3. Run test:

```
$ ./vsperf --conf-file=<path_to_custom_conf>/10_custom.conf
```

1.12 Executing PVP tests using Vanilla OVS

To run tests using Vanilla OVS:

1. Set the following variables:

```
VSWITCH = 'OvsVanilla'
VNF = 'QemuVirtioNet'

VANILLA_TGEN_PORT1_IP = n.n.n.n
VANILLA_TGEN_PORT1_MAC = nn:nn:nn:nn:nn:nn

VANILLA_TGEN_PORT2_IP = n.n.n.n
VANILLA_TGEN_PORT2_MAC = nn:nn:nn:nn:nn:nn

VANILLA_BRIDGE_IP = n.n.n.n

or use --test-param
```

```
$ ./vsperf --conf-file=<path_to_custom_conf>/10_custom.conf
           --test-param "vanilla_tgen_tx_ip=n.n.n.n;
                         vanilla_tgen_tx_mac=nn:nn:nn:nn:nn:nn"
```

2. If needed, recompile src for all OVS variants

```
$ cd src
$ make distclean
$ make
```

3. Run test:

```
$ ./vsperf --conf-file<path_to_custom_conf>/10_custom.conf
```

1.13 Selection of loopback application for PVP and PVVP tests

To select loopback application, which will perform traffic forwarding inside VM, following configuration parameter should be configured:

```
GUEST_LOOPBACK = ['testpmd', 'testpmd']
```

or use –test-param

```
$ ./vsperf --conf-file=<path_to_custom_conf>/10_custom.conf
      --test-param "guest_loopback=testpmd"
```

Supported loopback applications are:

```
'testpmd'       - testpmd from dpdk will be built and used
'l2fwd'         - l2fwd module provided by Huawei will be built and used
'linux_bridge'  - linux bridge will be configured
'buildin'       - nothing will be configured by vsperf; VM image must
                  ensure traffic forwarding between its interfaces
```

Guest loopback application must be configured, otherwise traffic will not be forwarded by VM and testcases with PVP and PVVP deployments will fail. Guest loopback application is set to ‘testpmd’ by default.

1.14 Code change verification by pylint

Every developer participating in VSPERF project should run pylint before his python code is submitted for review. Project specific configuration for pylint is available at ‘pylint.rc’.

Example of manual pylint invocation:

```
$ pylint --rcfile ./pylintrc ./vsperf
```

1.15 GOTCHAs:

1.15.1 OVS with DPDK and QEMU

If you encounter the following error: “before (last 100 chars): ‘-path=/dev/hugepages,share=on: unable to map backing store for hugepages: Cannot allocate memoryrnrn” with the PVP or PVVP deployment scenario, check the amount of

hugepages on your system:

```
$ cat /proc/meminfo | grep HugePages
```

By default the vswitchd is launched with 1Gb of memory, to change this, modify –socket-mem parameter in conf/02_vswitch.conf to allocate an appropriate amount of memory:

```
VSWITCHD_DPDK_ARGS = ['-c', '0x4', '-n', '4', '--socket-mem 1024,0']
```

1.16 More information

For more information and details refer to the vSwitchPerf user guide at: http://artifacts.opnfv.org/vswitchperf/brahmaputra/userguide/index.html