

6WIND Virtual Accelerator Demo

Contents

1 Iperf Documentation and Use Cases	1
2 Iperf TCP configuration commands	1
2.1 Server	1
2.2 Client	1
3 Iperf UDP configuration commands	2
3.1 Server	2
3.2 Client	2
4 Options	2
5 Virtual Accelerator demonstration: Iperf test use cases	2
5.1 Network creation	3
5.2 VMs spawning	3
5.2.1 VMs on compute nodes with 6WIND Virtual Accelerator	3
5.2.2 VMs on compute nodes without 6WIND Virtual Accelerator	3
5.3 Two VMs on different compute nodes with 6WIND Virtual Accelerator	3
5.4 Two VMs on different compute nodes without 6WIND Virtual Accelerator	4

1 Iperf Documentation and Use Cases

6WIND's [Virtual Accelerator](#) product dramatically boosts the network performance of Linux machines by offloading the forwarding function from the Linux kernel. The following demonstration uses the network bandwidth measurement tool Iperf to show the performance boost. Iperf is a widely used tool that creates TCP and UDP data streams and measures the throughput of a network.

2 Iperf TCP configuration commands

2.1 Server

```
iperf3 -s
```

2.2 Client

For the client, the IP address of the iperf server must be specified with the option -c.

```
iperf3 -c serverIP
```

3 Iperf UDP configuration commands

3.1 Server

```
iperf3 -s
```

3.2 Client

```
iperf3 -c serverIP -u -b 10G
```

The -b option is used to set the target bandwidth. In this example the target bandwidth is set to 10Gbps.

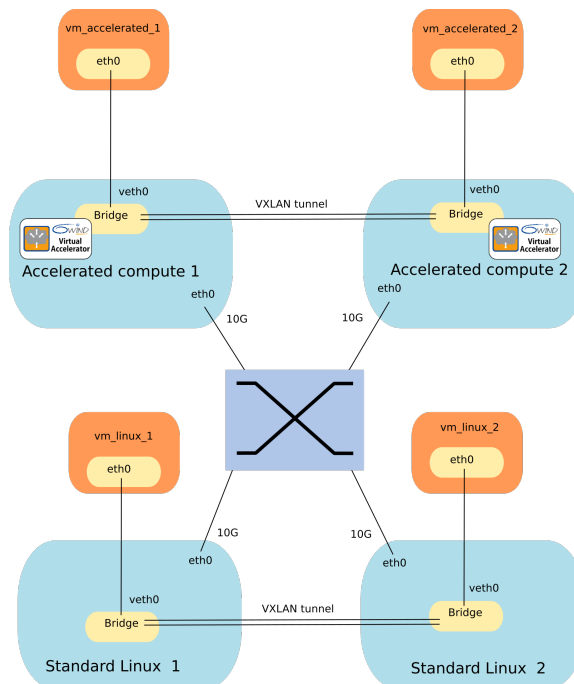
4 Options

With UDP or TCP client, several options can be used:

- -p : specify the port used (default is 5201)
- -t : specify the duration of the test (default is 10 secs)

5 Virtual Accelerator demonstration: Iperf test use cases

Using Iperf two tests are connected. Each test will use a pair of VMs; each VM resides on a different (compute) node. The diagram shows the four nodes each with a single VM. Any data transfer between the pair must traverse the VXLAN tunnels. Note that the physical NICs are connected to the switch via 10GE links. One test uses the compute nodes with [6WIND Virtual Accelerated](#) installed, the other uses the two compute nodes with standard Linux. To actually run the tests, you will create the topology shown in the diagram.



It is assumed that you have logged into the portal and retrieved the OpenStackrc file to execute the following commands.

```
source ${project}-openrc.sh
```

For information related to login and OpenStackrc file retrieval, please refer to the 'Tutorial Advantech Platform Portal' document. Review tutorial sections 1 and 2.

5.1 Network creation

For all the use cases, the network will be the same. Let's create network and save its ID in a variable:

```
neutron net-create net_test
neutron subnet-create --name subnet_test net_test 10.0.0.0/24
net_id=$(neutron net-list | grep net_test | awk '{ print $2}')
```

```
echo "$net_id"
```

5.2 VMs spawning

One VM on each compute node is spawned.

Let's create the VMs on the Accelerated compute nodes:

5.2.1 VMs on computes nodes with 6WIND Virtual Accelerator

```
nova boot --flavor nodisk.tiny --image "Ubuntu 14.04 template" \
--nic net-id=${net_id} --security-group default \
--availability-zone "Accelerated compute 1" vm_accelerated_1

nova boot --flavor nodisk.tiny --image "Ubuntu 14.04 template" \
--nic net-id=${net_id} --security-group default \
--availability-zone "Accelerated compute 2" vm_accelerated_2
```

and now, the VMs on standard Linux:

5.2.2 VMs on compute nodes without 6WIND Virtual Accelerator

```
nova boot --flavor nodisk.tiny --image "Ubuntu 14.04 template" \
--nic net-id=${net_id} --security-group default \
--availability-zone "Standard Linux 1" vm_linux_1

nova boot --flavor nodisk.tiny --image "Ubuntu 14.04 template" \
--nic net-id=${net_id} --security-group default \
--availability-zone "Standard Linux 2" vm_linux_2
```

Note: The username/password for the template are root/vbreizh

5.3 Two VMs on different compute nodes with 6WIND Virtual Accelerator

For this scenario, vm_accelerated_1 and vm_accelerated_2 will be used.

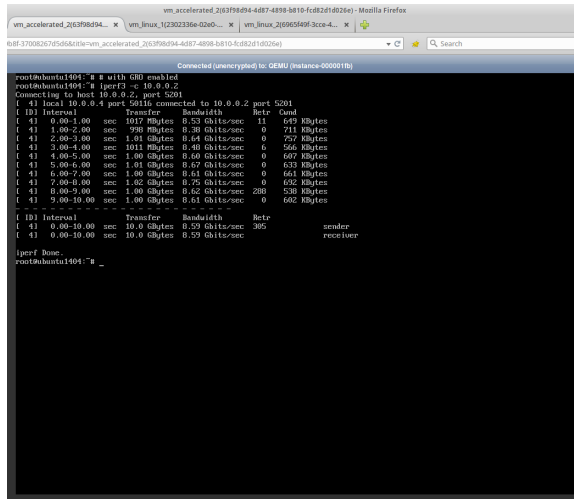
After the VMs are active and running, open a console window to each VM by right-clicking the VM and selecting "open link in new tab." Select console and then select "Click here to show only console". Login (using root/vbreizh), and launch the iperf server on vm_accelerated_1:

```
vm_accelerated_1# iperf3 -s
```

Launch the iperf client on vm_accelerated_2 and connect to the server:

```
vm_accelerated_2# iperf3 -c <IP address of server @ vm_accelerated_1>
```

The throughput should be around 8.6 Gbits/s.



5.4 Two VMs on different computes without 6WIND Virtual Accelerator

For this scenario, vm_linux_1 and vm_linux_2 will be used.

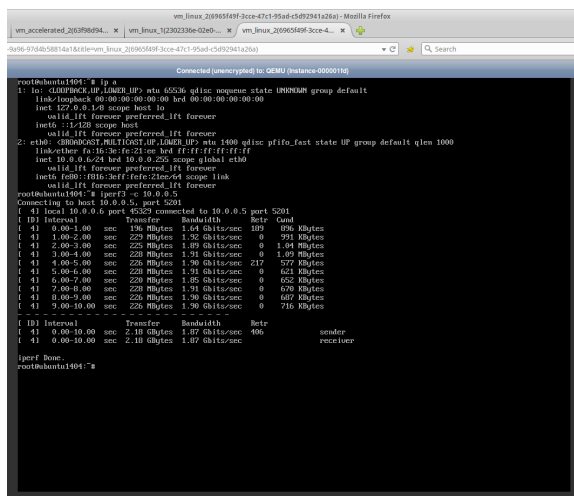
After the VMs are active and running, open a console window to each VM by right-clicking the VM and selecting "open link in new tab." Select console and then select "Click here to show only console". Login (using root/vbreizh), and launch the iperf server on vm_linux_1:

```
vm_linux_1# iperf3 -s
```

Launch the iperf client on vm_linux_2 and connect to the server:

```
vm_linux_2# iperf3 -c <IP address of server @ vm_linux_1>
```

The throughput should be around 1.9 Gbits/s.



Note the 5x performance boost in the 'accelerated zone.'
This concludes the Virtual Accelerator Iperf demonstration.