VSPERF: Iruya and Beyond

Introduction

The colored blocks shows the current Features of VSPERF, the White-blocks shows the features we plan to add to VSPERF, in near future.

Significance of VSPERF

VSPERF Only

VSPERF can be used for both Lab and Production/Live Environments, as summarized in the figure below. Typically, in cloud deployments, Telcos define set of "Acceptance Tests" (AT) for cloud. Performance testcases can be part of this AT, and VSPERF is the ideal tool to be used to run those testcases.
VSPERF with other OPNFV Testing Projects:

Similar to NFVBench, measure and assess the L2/L3 forwarding performance of an NFV-infrastructure solution stack (i.e. OPNFV scenario) using a black-box approach. It can be agnostic of the installer used, the hardware used, the controller used (ML2/OVS, ODL...), the network stack used (OVS, OVS-DPDK, VPP...).

Five Main Focus Areas of VSPERF for Iruya and Beyond:

Continued Contribution to Standardization
ETSI TST 009 and VSPERF:

<table>
<thead>
<tr>
<th>Section</th>
<th>Compliance Comments</th>
<th>Missing Aspects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Setups</td>
<td>Fully Compliant</td>
<td></td>
</tr>
<tr>
<td>Throughput</td>
<td>Missing capacity reporting. Only throughput is reported.</td>
<td>Capacity with X % Loss</td>
</tr>
<tr>
<td>Latency</td>
<td>Missing Transfer time in percentile. Other 4 supported.</td>
<td>Transfer Time (Yth percentile, X % Thpt, Filter)</td>
</tr>
<tr>
<td>Delay Variation</td>
<td>Missing FDV and IFDV with %le and %ge throughput</td>
<td>FDV (Yth percentile, X % Thpt, Filter) and Inter-Frame Delay Variation (X % Thpt, Filter)</td>
</tr>
<tr>
<td>Loss</td>
<td>Missing loss ratio with %ge throughput and loss-free seconds.</td>
<td>Loss Ratio (X % Thpt, Filter), Loss-Free Seconds (X % Thpt, Filter)</td>
</tr>
<tr>
<td>Methods of Measurement</td>
<td>Pre-test requirements automated.</td>
<td>Binary Search with NDR/PDR is missing</td>
</tr>
<tr>
<td></td>
<td>Binary Search with LV needs minor update?</td>
<td></td>
</tr>
<tr>
<td></td>
<td>NDR/PDR support is lacking.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Setting up Cross-Numa scenarios automated.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Better configuration for Soak-Tests</td>
<td></td>
</tr>
</tbody>
</table>

IETF
GSMA

Container Networking Performance Benchmarking

Goal: **Automated Setup of DUT, Comparative Study, Generate Results.**

The setup requirements:

2. Two interfaces in Container for 'east' and 'west' traffic.
   a. CN solutions multiple interfaces support.
3. Two Physical Interfaces on the host to which Traffic-Generator is connected.
   a. CN solution to use these 2 physical interfaces.
   b. CN to bridge/route the traffic across physical-physical, physical-virtual, virtual-physical and virtual-virtual interfaces.

To realize the first step of "Automated Setup of DUT", there are different options:

1. CN without Kubernetes
2. CN With Kubernetes
   a. Manual
   b. kubeadm
   c. Kubespray + Kubeadm
   d. Multus/DanM
   e. Multus + ? Plugin
   f. Multus + Userspace CNI?

The DUT architecture would be:
Containerization of VSPERF and Integration with Larger Test-Automation Frameworks

Target Frameworks: Dovetail and X-Testing.

This task has two main parts:

Part-1: Containerizing VSPERF.

Controller Container: Separating Deployment from Testing

<table>
<thead>
<tr>
<th>No.</th>
<th>Title</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Auto-Deploy</td>
<td>This container takes the following inputs and automatically deploy VSPERF and setup the TGen if required.</td>
<td>Ideal for integrating with CI/CD based larger test automation Frameworks</td>
</tr>
</tbody>
</table>
## 2. Interactive-Deploy
This container starts listening to commands from the user to perform the deployment of VSPERF and TGen.
Ideal for integrating with other ‘interactive’ (web-based) test automation frameworks.

## 3. Auto-TestControl
This container takes the following inputs and automatically runs the tests and publishes the results.
Ideal for integrating with CI/CD based larger test automation frameworks.

## 4. Interactive-TestControl
This container starts listening to commands from the user to perform the testing.
Ideal for integrating with other ‘interactive’ (web-based) test automation frameworks.

### Part-2: Integrating VSPERF with larger Test-Automation Frameworks
We are targeting both Dovetail and X-testing.

#### Analytics

##### Basic Analytics
Reference work, which was demonstrated at ONS-NA 2019 can be accessed here: [https://github.com/opensource-tnbt/vsperf-notebooks](https://github.com/opensource-tnbt/vsperf-notebooks)

##### Advanced Analytics
Work in Progress, in collaboration with Vipin Rathi

#### Ease of use

##### Configuration Generation Tool:
A PoC can be found here: [https://github.com/opensource-tnbt/vsperfwizard](https://github.com/opensource-tnbt/vsperfwizard)

##### GUI based Test management tool.
Reuse Yardstick GUI?

#### Dashboard
Custom Dashboard to showcase the status of VSPERF Ongoing Test - Topology, Traffic, etc.
We are looking for some volunteers to develop this - Maybe we can reuse

### Other Works:
1. New vSwitches support - Tungsten vRouter
2. Kernel Fastpaths (ex: AF_XDP)