OPNFV Data Plane Performance Testing Projects Solution Brief

OVERVIEW:

- Network functions virtualization (NFV) is critical for 5G and related initiatives
- Network service performance is tightly coupled to business case
- Communications service providers (CSPs) need to expand performance validation efforts with NFV

CURRENT PROBLEM:

- Industry standard servers offer a large number of design and configuration options that directly affect performance
- VNF validation and performance characterization needs to be conducted in a real-life environment

---

SOLUTION — OPNFV PROJECTS:

- Bottlenecks
- QTIP
- NFVbench
- StorPerf
- VSPERF
- Yardstick

OVERVIEW

Network Functions Virtualization (NFV) is an important technology that helps build network services using virtual network functions (VNFs) on industry standard Intel® and ARM® servers.

NFV does not just stop with virtualization — it brings automation to cloud resource, VNF, and network service provisioning and, over time, fully automated monitoring and service assurance. NFV promises to help Communications Service Providers (CSPs) increase revenue, improve customer satisfaction, slash capital and operational expenditures (CAPEX and OPEX). In order to successfully introduce 5G, edge computing, and IOT services, NFV is rapidly becoming a mandatory technology.

However, NFV brings with it issues that CSPs did not previously have to worry about to the same degree. One such topic is performance of network services. Performance directly affects CAPEX since it determines the number and flavors of NFV hardware infrastructure required to support a given number of subscribers.

Broadly speaking, performance may be classified into three categories: data plane, control plane, and management plane. Control plane and management plane performance are indeed important and can affect overall agility and availability of the network service; however the performance that we will discuss in this document is data plane performance since it most directly affects user experience. Data plane performance measures the performance of network traffic and is captured using metrics such as throughput, packets/sec, latency, packet loss, single/multi-stream performance, and others.

PROBLEM STATEMENT

Given how interrelated performance is to the overall business case, optimizing the performance of network services is critical for CSPs.

However, when it comes to data plane performance, NFV and general purpose systems create a unique set of challenges. The first challenge is to identify and optimize NFV infrastructure (NFVI) performance related factors, an activity also known as tuning. The second relates to determining the true performance of a VNF in a real-life environment.

NFVI Performance Factors and Tuning

The following four factors affect NFVI data plane performance.

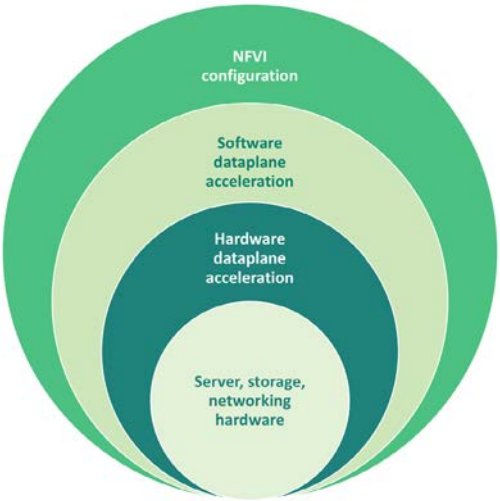

Figure 1: Performance Factors

At the basic level, server, storage and networking hardware affect performance. These might include CPU core count, memory speed, memory quantity, internal disk flavor, external storage server sizing, NIC speeds, number of physical networks, and so on. This factor also includes connectivity speeds (e.g. 1GbE vs. 10GbE vs. 25GbE) and the topology interconnecting servers and storage (e.g. leaf-spine).

Next, performance depends on data plane acceleration. This acceleration is of two types: hardware and software. Hardware acceleration might includes smart NICs, FPGAs, GPUs, custom ASICs (e.g. crypto), cache features, SR-IOV, and so on. Software acceleration includes components such as DPDK, FD.io/VPP, IOVisor, OpenDataplane, memory/CPU pinning, and real-time Linux.

Finally, NFVI configuration parameters affect performance as well. Configuration such as huge pages, jumbo frames, NIC bonding, number of PCI virtual functions, and numerous others can have a material impact on data plane performance.

These factors have to be optimized as a whole system (as opposed to piecemeal) for the network service being considered. Additionally, since it is not possible to have a different node flavor per VNF, a CSP needs to consolidate all their performance requirements and boil them down to a manageable number of physical node types.

In short, performance of a network service is complex with many trade-offs. There are numerous metrics relevant to this activity impacted by a multitude of design and deployment factors.

Determining VNF Performance

The next challenge is to determine the true performance of a VNF in real-life conditions. This is critical for making decisions such as vendor selection and capacity planning. A vendor typically benchmarks their VNF on a system that brings out the best performance for their product. For this reason, the environment, tuning, and test suites used by the vendor might be markedly different from a CSP's production environment. For example, if a VNF vendor uses 10 x 10GbE NIC cards per server and consumes 100% of CPU resources to demonstrate a benchmark, this is not going to be useful in the real world.

A real-life scenario also means other important deployment considerations and constraints that can substantially impact the overall performance of the system. For example, parameters such as rack density, port density, cost of the top-of-rack switches,

power consumption, cost of various NIC cards (10GbE vs 25GbE vs. 40GbE etc.), and others all contribute to making one system better than another from an overall CAPEX standpoint. Since the management burden for different systems varies, these variations can also affect OPEX. Ultimately, these factors distinguish traditional lab performance benchmarking, where practical deployment conditions and economics may be underemphasized, from production performance benchmarking.

SOLUTION

Recognizing these challenges around performance, the OPNFV community has created a number of data plane performance testing projects.

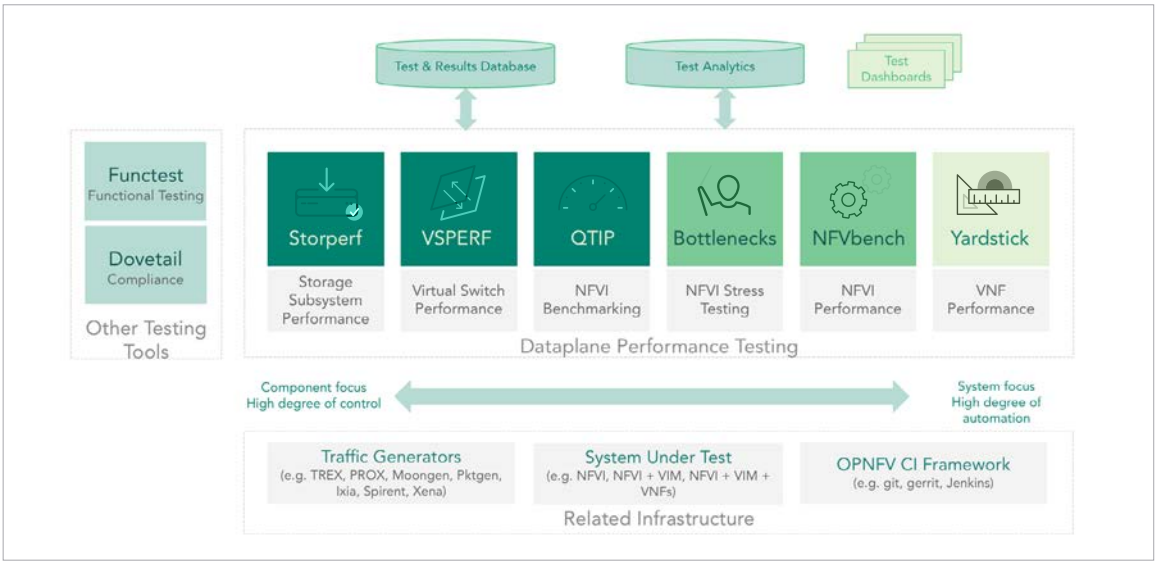

Figure 2: OPNFV Testing Projects

These projects solve both challenges discussed above by helping optimize NFVI performance and determining VNF performance. We need these different performance tools to address the the complexity of the various trade-offs discussed above.

The OPNFV data plane performance testing projects may be classified by their testing scope (or system-under-test i.e. SUT) as follows:

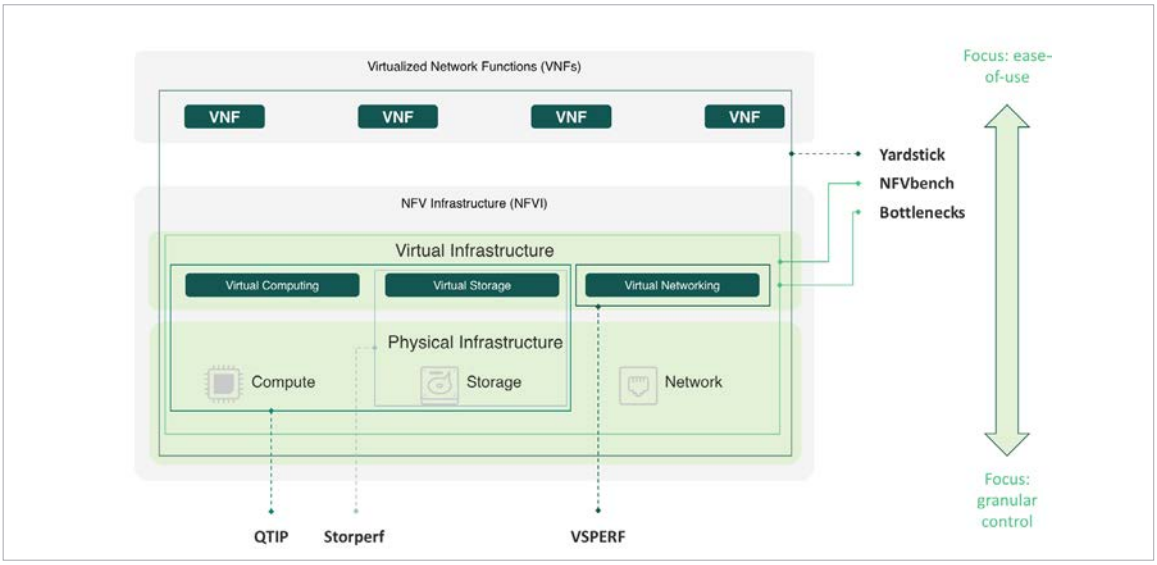

Figure 3: Scope of Different Performance OPNFV Testing Projects

NFVI Focused Projects

The NFVI layer needs to be optimized at both a component level and at a subsystem level. Comparing NFVI to a car, a car designer has to optimize each component of the car e.g. the engine, transmission system, etc. and then also the fully assembled car. Projects such as QTIP (compute/storage benchmarking), StorPerf (storage performance testing), and VSPERF (virtual switch performance testing) allow each individual component to be tested and optimized. Subsystem level projects such as Bottlenecks (stress testing), NFVbench (full NFVI path testing), and Yardstick (NFVI and network service testing) allow for NFVI layer testing and optimization.

VNF Focused Project

The OPNFV Yardstick project with the Network Service Benchmarking (NSB) initiative may be used to test VNF performance across vendors in a consistent manner. As a VNF cannot be tested independently of NFVI, testing VNF performance implicitly tests the cobmination of NFVI + VNF.

In contrast to the project scope, another way to classify the above projects is by environment focus: lab vs. production.

	Lab	Production
Projects	Bottlenecks QTIP StorPerf VSPerf Yardstick	NFVbench

Table 1: Classification of OPNFV Performance Projects by Environment

Lab vs. Production

Most of the OPNFV performance projects are intended to be used in a dev/test style lab environment. Consequently, these tools also target developers such as NFVI ingredient developers, NFVI architects, and VNF developers.

NFVbench, on the other hand, is intended to be run on production/pre-production environment and targets users such as operators, solution architects, and so on.

Next, let us review each project in more detail to see what specific problems they solve.

OPNFV DATA PLANE PERFORMANCE TESTING PROJECT DETAILS

There are six data plane performance testing related projects in OPNFV. Three of them — QTIP, StorPerf, and VSPERF — deal with component testing. The other three NFVbench, Yardstick NSB, and Bottlenecks relate to subsystem or system level testing.

Let us first look at the component testing projects.

QTIP

As the "Platform Performance Benchmarking" project in OPNFV, QTIP aims to provide users a simple indicator for NFVI compute and storage performance, supported by comprehensive testing data and a transparent calculation formula. QTIP can help users choose the right hardware and tune it appropriately.

The core values of QTIP are as follows:

- **Transparent:** Being an open source project, users can inspect all the details behind the QTIP Performance Index (QPI), e.g. formulas, metrics, and raw data.
- **Reliable:** The integrity of QPI is guaranteed by traceability in each step back to raw test results.
- **Understandable:** The QPI report is broken down into section and workload scores, to help users better understand them.
- **Extensible:** Users may create their own QPI by composing existing QTIP metrics or adding new ones.

The compute metrics include Dhrystone 2.1, Whetstone, OpenSSL Speed, RAMSpeed, DPI while the storage metrics cover IOPS, bandwidth (number of kilobytes read or written per second), and latency.

StorPerf

The purpose of StorPerf is to provide a tool to measure ephemeral and block storage performance of NFVI. This project can help users select and tune their storage hardware and software appropriately to meet their performance needs.

A key challenge to measuring disk performance is to know when the disk (or, for OpenStack, the virtual disk or volume) is performing at a consistent and repeatable level of performance. Initial writes to a volume can perform poorly due to block allocation, and reads can appear instantaneous when reading empty blocks. It is a challenge to determine what performance data is valid. The Storage Network Industry Association (SNIA) has developed methods which enable manufacturers to set, and customers to compare the performance specifications of Solid State Storage (SSD) devices .

Once launched, StorPerf presents a user with a REST interface. Issuing an HTTP POST to the configurations API causes StorPerf to talk to OpenStack's Heat service to create a new stack with as many agent VMs and attached Cinder volumes as specified. After the stack is created, a user can trigger one or more jobs. The job is the smallest unit of work that StorPerf can use to measure the disk's performance.

While the job is running, StorPerf collects the performance metrics from each of the disks under test every minute. Once the trend of metrics match the criteria specified in the SNIA methodology, the job automatically terminates and the valid set of metrics are available for querying.

StorPerf provides the following metrics:

- IOPS
- Bandwidth (number of kilobytes read or written per second)
- Latency

VSPERF

VSPERF provides an automated test-framework and comprehensive test suite based on industry test specifications (IETF RFCs 2544, 2889) for measuring NFVI data plane performance. The datapath includes switching technologies with physical and virtual network interfaces. The VSPERF architecture is switch and traffic generator agnostic and test cases can be easily customized. Software versions and configurations including the vSwitch (OVS or FD.io VPP) as well as the network topology are controlled by VSPERF independent of OpenStack. VSPERF is used as a development tool for optimizing switching technologies, qualification of packet processing components and for pre-deployment evaluation of the NFV platform datapath.

VSPERF provides a framework where the entire NFV industry can learn about NFVI data plane performance and try-out new techniques together. VSPERF has contributed to the development of a new IETF benchmarking specification (RFC8204) and is also contributing to the development of ETSI NFV test specifications. VSPERF supports a number of commercial and open source traffic generators. Some of the newer features include additional test cases, flexibility in customizing test-cases, new results display options, improved tool resiliency, additional traffic generator support, and VPP support.

VSPERF supports designing and implementing custom tests through its 'integration-tests' feature. Users can add custom 'steps', (for example, include encap-decap mechanisms to any test) and create a custom testcase. Since the focus of the project is at a component level, one example of how VSPERF provides users with granular control is enabling the user to utilize VSPERF to set up the SUT and traffic generator, just the traffic generator (i.e. user sets up the SUT), or just the SUT (i.e. user sets up the traffic generator). VSPERF sets up the environment, starts the test, collects/publishes the results. Again, there is flexibility in allowing a user to set up the environment and/or collect/publish their own results. A few other configuration examples of VSPERF's fine-grained control philosophy are virtual switch version, virtual switch configuration, configuration of tests, VNF configuration, neighboring VNF configuration (e.g. noisy-neighbor), and traffic generator configuration. VSPERF also provides very detailed information about test results including items such as the entire runtime environment, resources available vs. allocated, timestamps, infrastructure metrics from OPNFV Barometer project (e.g. cache usage, process details), and others. This level of detail is important for an developer that wants to look at correlation and causality of events.

Next, let us review the three subsystem or system level projects.

NFVbench

The NFVbench tool provides an automated way to measure the network performance for the most common data plane packet flows on any OpenStack system. It measures the data plane performance of NFVI as a black box that runs behind top of rack switches. It is designed to be easy to install and easy to use by non-experts (i.e. there is no need to be an expert in traffic generators and data plane performance testing). The tool is built around the open source TRex traffic generator and is useful for testing a full NFVI subsystem that includes ToR switches. The key areas of strength for NFVbench are in its automation of the traffic generator, ability to test a full subsystem, and to perform this testing on a production cloud.

Some of the data plane performance measurement features include:

- Support for two measurement modes — fixed rate mode to generate traffic at a fixed rate for a fixed duration and NDR (No Drop Rate) and PDR (Partial Drop Rate) measurement mode
- Configurable frame sizes (any list of fixed sizes or 'IMIX')
- Built-in packet paths with OpenStack support (PVP: physical interface→VM→physical interface, PVVP: physical interface→VM→VM→physical interface, N*PVP: N concurrent PVP paths, and N*PVVP: N concurrent PVVP paths)
- Built-in loopback VNFs based on fast L2 or L3 forwarders running in VMs
- Configurable number of flows and service chains
- Configurable traffic direction (single or bi-directional)

Results of each run include the following data:

- Aggregated achieved throughput in bps
- Aggregated achieved packet rate in pps (or fps)
- Actual drop rate in PDR %
- Latency in usec (min, max, average in the current version)

Yardstick

Yardstick's original goal was to verify infrastructure compliance, mainly around performance using ETSI reference test suites from the perspective of a Virtual Network Function (VNF). After surveying a large number of NFV workloads, the Yardstick project has broken down overall requirements into a set of performance vectors to quantify compute, network, and storage aspects of NFVI. The project develops the test framework and test cases for each performance vector, and some test cases can get quite complex where Yardstick needs to run tests in parallel, inject faults, and test multiple topologies.

Since the OPNFV Euphrates release in 2017, the project has included an additional initiative called Network Service Benchmarking (NSB). The goal of NSB is to extend Yardstick to perform real world VNF and NFVI characterization and benchmarking with repeatable and deterministic methods. NSB extends the Yardstick framework to perform VNF characterization and benchmarking in three different execution environments — bare metal (i.e. native Linux environment), standalone virtual environment, and managed virtualized environment (e.g. OpenStack). It also brings in the capability to interact with external traffic generators, both hardware and software based, for triggering and validating the traffic according to user defined profiles.

NSB can test combinations such as:

- Different NFV infrastructure connecting nodes (MPLS, Vxlan, GRE, no-tunneling)
- Different ways of connecting VNFs to NFV infrastructure (virtio, vhost-dpdk, Single Root I/O Virtualization or SR-IOV)
- Different ways to provide connectivity inside NFVI (OVS, OVS-DPDK, Tungsten Fabric vRouter, physical switches)
- Different traffic generators (HW generators, SW generators like TRex, pkt-gen, prox)
- Compute, network, and storage solutions from different vendors
- VNF implementations from different vendors performing the same network function

NSB can be used for a variety of use cases ranging from performance characterization of VNFs, comparison of different VNFs, and capacity planning.

To fully demonstrate NSB, a related OPNFV project — samplevnf — provides VNFs that approximate the data plane functionality of commercial VNFs. The samplevnf project includes virtual CG-NAPT (Network Address and Port Translation), virtual ACL (Access Control List), virtual FW (Stateful Firewall), virtual PE (Provider Edge Router), virtual

EPC (Enhanced Packet Core). These are not production VNFs; instead useful only for performance testing and characterization purposes.

Bottlenecks

This project aims to find system bottlenecks by testing and verifying NFVI in a staging environment before committing it to a production environment. It finds bottlenecks by applying stress to the NFVI and now includes long-duration testing as well.

Bottlenecks is not just a data plane performance testing project, rather it tests both control-plane and data plane. However for purposes of this discussion, we will focus on data plane aspects. Bottlenecks performs stress testing on the network aspect of NFVI to measure it against benchmarks such as throughput, number of connections, packet delay, etc. The project calls other projects such as Yardstick and StorPerf iteratively and acts as "load manager" by varying the load to provide the required amount of stress to the NFVI. Bottlenecks can run multiple Yardstick tests in parallel to accomplish this. By running these stress tests over a long period of time, Bottlenecks also performs long-duration testing. These tests include additional monitoring via Prometheus, Collectd and Node, and visualization through Grafana. Bottlenecks works on a VM based NFVI orchestrated by OpenStack and is expanding its coverage to a container based NFVI orchestrated by Kubernetes (k8s).

SUMMARY

Network service performance directly affects a CSP's business case. With NFV, two new problems emerge. First, an industry standard server allows for endless permutations on component choices and configurations. Second, determining the true performance of a VNF in a real-life production environment is non-trivial. Solving these problems used to be more of an art form than a science. With the advent of OPNFV data plane performance projects: Bottlenecks, QTIP, NFVbench, StorPerf, VSPERF, and Yardstick CSPs can now jumpstart their NFVI and VNF testing under their unique NFV deployment strategies, timelines, and network configurations to accelerate their path to deployment.

We encourage you to download and try out these test projects!

RESOURCES

OPNFV project
OPNFV downloads
OPNFV Testing Working Group
Bottlenecks
QTIP
NFVbench
samplevnf
SNIA
StorPerf
Test-WG mailing list
VSPERF
Yardstick