WHITE PAPER

ixia

Examining Factors of the NFV-I Impacting Performance and Portability

915-0953-01 Rev. A, May 2015

Table of Contents

Introduction

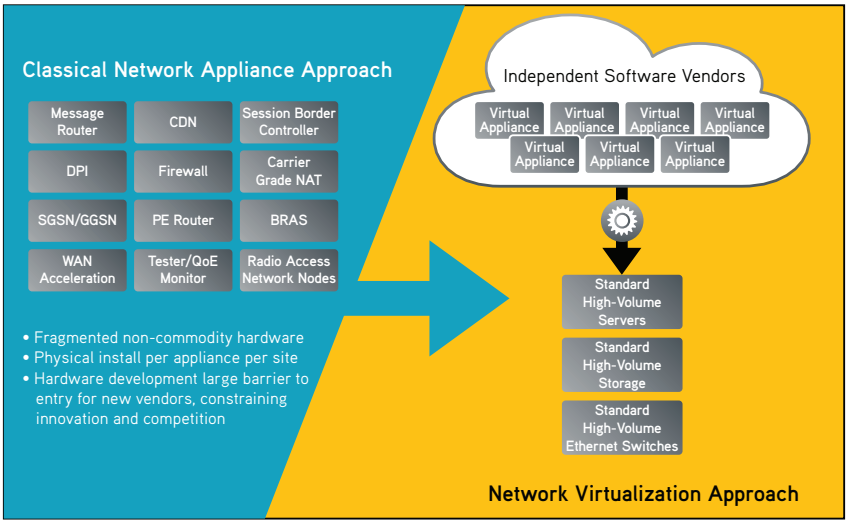
Service providers are facing significant market challenges and, as a result, are looking for every opportunity to reduce costs (CAPEX and OPEX), increase service agility and optimize resources. Network Functions Virtualization (NFV) promises to address these challenges and transform the network into a next generation platform with carrier-grade services.

The premise of NFV is to evolve from running network functions on appliances using purpose-built hardware to running in software as a virtual network function (VNF) on virtualized commercial off-the-shelf (COTS) x86-based compute platforms. The concept of running network functions in software on generalized compute platforms is not new. Software-based network functions have been available for years (e.g., Quagga open source router and the strongSwan IPsec gateway).

The concept of running network functions in software on generalized compute platforms is not new, in fact, there have been software based network functions available for many years.

Most Network Equipment Manufactures (NEMs) develop their solutions in software, then port to hardware for specific hardware-enabled functions and performance. Several changes have taken place, causing service providers to move toward COTS hardware for their network services. For example, the evolution of the multi-core CPU, which can handle 250 million packets per second and has been shown to support hundreds of Gigabits per second on a single server platform. Also, the progression of cloud computing has showcased the agility of a public or private cloud and the ability to quickly spin up virtualized services.

Service providers want to drive down the cost of network equipment by running on COTS servers and leveraging cloud technologies for agile provisioning and management of network services. Several leading service providers produced an initial white paper in 2012 and worked together in the ETSI Industry Study Group (ISG) for NFV. The ISG has outlined requirements and specifications, resulting in 17 documents addressing many areas.



The Vision of Network Function Virtualization (NFV) from the ETSI ISG for NFV

The platform enabling these services is defined by the NFV ISG as the Network Functions Virtualization Infrastructure (NFV-I). It is made up of the compute platform, Network Interface Cards (NICs), host operating system, a virtualization layer (aka hypervisor), and virtual networking technology (aka vSwitch). On top of this platform, Virtual Network Functions (VNFs) are deployed typically as Virtual Machines (VMs), which are assigned CPU, memory and storage. (An alternative technology is Linux Containers, however that will not be covered in this document.)

The second critical part of this platform is the Management and Orchestration (MANO). This document will not focus on MANO, but it is typical to use a Virtual Infrastructure Manger (VIM) to handle the instantiation of VMs and configure the required networking. Although orchestration is a critical component, it first requires that the NFV-I be verified. This document examines many factors of the NFV-I affecting performance and portability of the VNF and, in turn, the overall service.

Service providers require predictable performance of the VNFs so they are able to deliver on Service Level Agreements (SLAs). Server virtualization enables portability with features like VM migration providing the ability to move a running VM from one server to another. Service providers will need this to deliver high availability when addressing maintenance of the compute system, or when moving the VM to a compute platform with more resources. Portability also means the ability to deploy a VNF anywhere it is needed, so consistency from the NFV-I is needed to support that deployment.

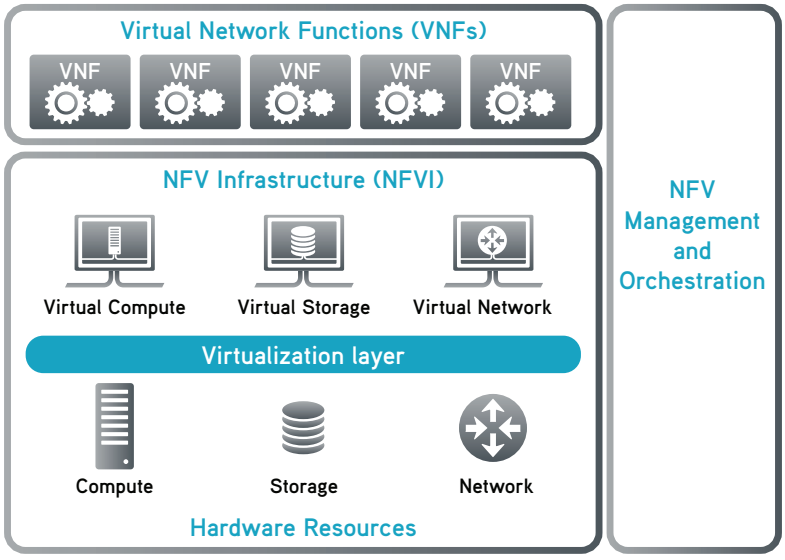
Service Providers require predictable performance of the VNFs

Virtualizing network functions and testing and certifying them on various NFV-I platforms is a major challenge for network equipment manufacturers. In many cases, they offer compute platforms—or partner either directly or through ecosystems in order to provide an integrated offering. By offering an integrated solution, they can ensure that the VNF will perform as expected since it will have been verified for interoperability and performance.

ixia Expert Tip Prior to testing a VNF, collect all details the vendor is willing to share, including the virtualization platforms the VNF has been tested on, along with supporting performance information. It is also important to know minimum and maximum resources the VNF can be allocated and how that affects performance.

Service providers must evaluate platforms for NFV—which include the NFV-I and MANO components, as well as individual VNFs—and then, eventually, services that run over one or multiple VNFs in a service chain. They move through individual component verification to Proof of Concept (PoC) test and then on to operationalizing the service for field trials and production service offerings.

When beginning a project for NFV, detailed planning and paper comparisons are necessary (since there are many interdependencies from one layer to the next). In essence, the NFV-I are puzzle pieces that may or may not fit together. To make the best choices, it is usually recommended to determine a use case to work toward so that the proper components are selected. Most operators are looking to standardize on a common platform for NFV and require it to be open source-based as well as multi-vendor.



Example depiction of the NFV-I provided by the ETSI ISG for NFV

The compute platform will have the single biggest impact on the performance of a VNF.

While the agility and flexibility of this system is great, the potential complexity is also quite great. Simply looking at the variables a vendor must test in order to verify their VNF on several platforms is a significant challenge. There are many different compute platforms, NICs, hypervisors and virtual networking options… and that's just the beginning.

Another significant challenge companies are facing is the skill gap. Many networking professionals have become experts in configuring devices via CLI. These devices are purpose built, so there are few requirements to customize or verify the internal architecture. When moving to standardized compute, it is necessary to know each component of the system, and how it is configured from the BIOS, when booting the server to the NIC drivers and everything in between. Users must have high proficiency using the hypervisor and virtualization manager in order to properly handle the instantiation, configuration and optimizations of the VNFs running as VMs on the NFV-I. This document will begin to look at those complexities so that users can effectively progress with development, testing and achieving predictable performance.

Compute Platform

The compute platform has the single biggest impact on the performance of a VNF. To determine the proper platform, users should know ahead of time what functions are to be deployed on the server, and how many. A given VNF may need as little as 2 CPU core and 2GB of memory. However, depending on what you are running, the VNF could require 16 core and 16GB of memory. This will become clearer as the impact of the compute platform is explained.

If you are looking at virtualizing a few network functions in a demonstration or test system, a one or two RU server may meet the requirements. If you are looking at chaining several VNFs and supporting hundreds or thousands of services, you may require several blade servers.

Small Platforms	Medium Platforms	Large Platforms
 Dell R620	 Dell R720	 Dell M1000e & M620Blades
 HP DL360	 HP DL380	 HP BL460

Example compute platforms offering increasing port density (NICs), CPU, memory and storage

Most NFV-I systems are made up of a controller node, compute nodes and possibly storage nodes. For high-availability, redundant controller and compute nodes are required. Controller nodes provide administrative functions as well as virtual infrastructure management (VIM) and VNF management. The compute nodes are the virtualized servers that host the VNFs and networking (physical and virtual) required by the service.

For high-availability redundant controller and compute nodes are required.

CPU/Motherboard and Storage

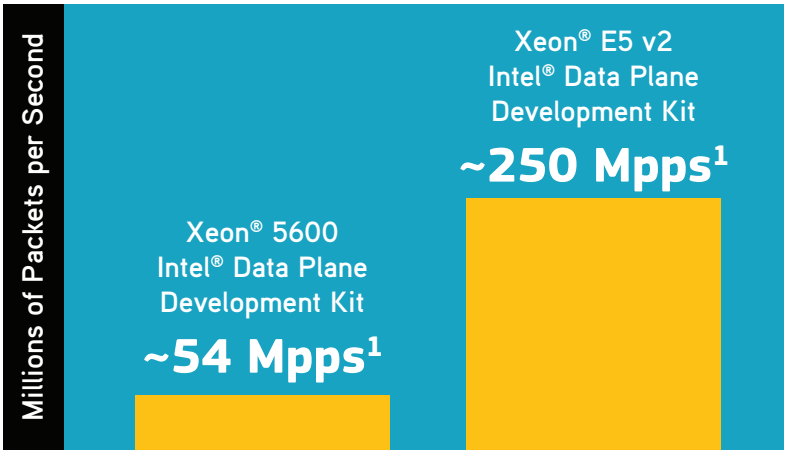
Modern CPUs, like those available from AMD and Intel, are multi-core and most server platforms offer multiple CPUs, e.g.:

- The current generation of CPU available from Intel is called Haswell (recently renamed to ARK) Processors (Xeon E5-26xx v3 series)
- Many available servers come with the Ivy Bridge processor family (Xeon E5-26xx v2 series)

When considering a server, there are several metrics that affect performance:

- Processor Speed (measured in Gigahertz; examples are 1.8GHz up to 2.9GHz; note that some offer turbo mode, which can increase the clock rate)
- L3 Cache (measured in Megabits, usually 5MB – 20MB)
- Cores (the number of physical cores on each processor)
- Processors (the number of physical processors, each having its own slot on the motherboard)
- QPI (the number of QPI connections between processors and speed)

Memory is also an important component since VNFs will require a specific allocation of memory.



Example performance improvements from Sandy Bridge to Ivy Bridge families from Intel

ixia **Expert Tip** When testing a platform, be sure to document all details of the compute system under test. This information will be required to verify or reproduce how the results were achieved. For example:

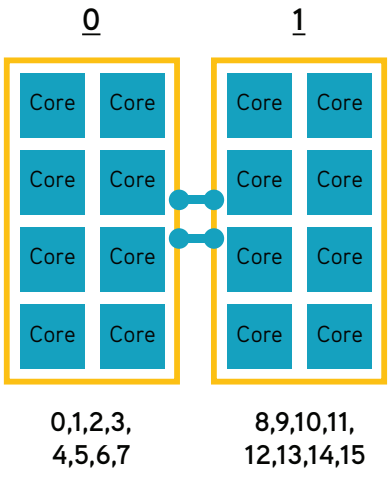
Component	Description	Quantity
Compute Platform	Dell PowerEdge R620	1
CPUs	Intel Xeon E-2650 v2 2.6GHz, 20M Cache, 8.0GT/s QPI, Turbo, HT, 8C, 95W Max Mem 1866MHz	2
Memory	16GB RDIMM, 1866MT/s, Standard Volt, Dual Rank, x4 Data Width	4
Storage	300GB 10k RPM SAS 6Gbps Hot-plug Hard Drive	2
Motherboard	R620 with 4 hard drive and 3 PCIe slots	1
Network Interface Cards	Broadcom 5720 QP 1GbE Network card (management) Intel X520-SR2 Dual 10GbE PCIe v2.0 (5.0GT/s) x8 Lane	1 2

Because VNFs require specific memory allocations, memory is also an important component. The number of DIMM slots and total memory affect how many VNFs can be supported. Also, the speed of the DIMMs matter. While most servers support mixed speeds, it is common for them to operate at the slowest DIMM's common frequency. Also, note that DIMM types (UDIMM, RDIMM or LRDIMM) typically cannot be mixed. When selecting or upgrading memory, this is an important consideration. Most servers run registered (i.e., buffered) memory RDIMMs because they offer more scalability and robustness over lower priced, unregistered UDIMMs. Load Reduced DIMM (LRDIMM) can control the amount of electrical current flowing to and from the memory chip at any time. These typically would only be used with motherboards that require them.

CPU Pinning

Provisioning VNFs requires the allocation of resources (memory and CPU). It is best practice to assign cores from the same CPU (i.e., the same socket) to a VNF for the best I/O performance and reduced latency. Assigning cores from different CPUs may result in degraded I/O performance and latency variation. This assignment of VMs to specific CPU cores is called "CPU pinning" and it is done so that Non-Uniform Memory Access (NUMA) boundaries are not crossed. (NUMA refers to the architecture between a systems processor and memory.) When a processor accesses memory that is not connected (i.e., remote memory), the data must be transferred over the NUMA connection and this impacts performance.

Assigning cores from different CPUs may result in degraded I/O performance and latency variation.



Example CPU core to processor mapping, each processor has 8 core

While hyper-threading is a feature commonly used, it can also degrade performance.

To check the NUMA configuration on the Linux host OS, use the `numactl --hardware` command, as shown below. The result shows node 0 has 8 CPU cores (0–7) and node 1 has 8 CPU cores (8–15).

```
[root@ixia dpdk-1.7.0]# numactl --hardware
available: 2 nodes (0-1)
node 0 cpus: 0 1 2 3 4 5 6 7
node 0 size: 32733 MB
node 0 free: 5990 MB
node 1 cpus: 8 9 10 11 12 13 14 15
node 1 size: 32768 MB
node 1 free: 7261 MB
```

To pin the hypervisor virtual CPU designations to the NUMA designations, the virsh command line tool can be used:

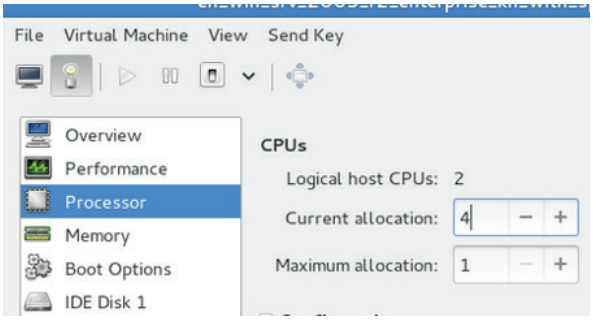
```
virsh vcpupin vnf1 0 0
```

Using a shell script to configure the required cores used by each VNF, where vnf1 is the guest virtual machine name:

[root@ixia ~]# virsh vcpupin vnf1

```
VCPU: CPU Affinity
----------------------------------
  0: 0
  1: 1
  2: 2
  3: 3
  4: 4
  5: 5
  6: 6
  7: 7
```

Note that some Virtual Machine Managers, and even Orchestration systems, now provide the ability to configure CPU pinning.



Example: How to allocate CPU to a VNF with a Virtual Machine Manager

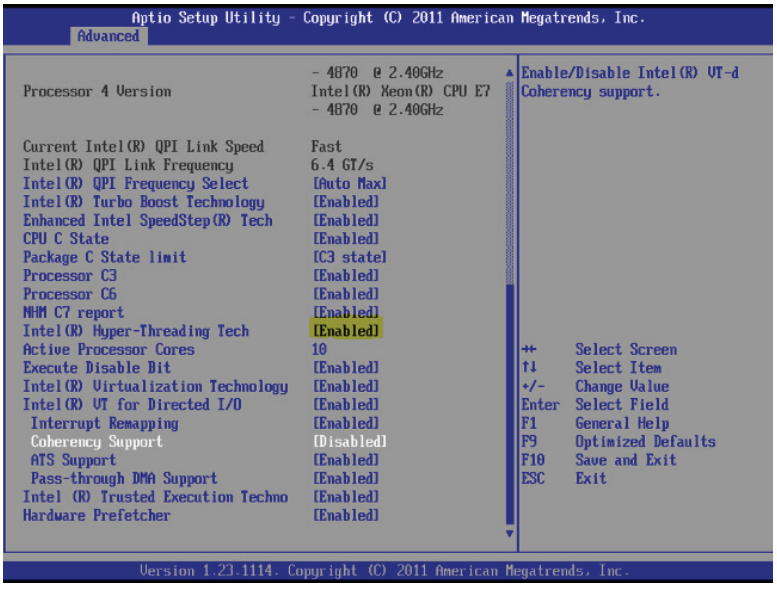
Logical Processor Support

Logical Processors—e.g., Intel's Hyper-Threading technology (HT)—were designed to use processor resources more efficiently, thus doubling the amount of physical cores of a multi-core CPU. For example, if a server has two (dual socket) Intel E5-2650 v2 processors, each with 8 physical cores, enabling HT provides 32 logical cores (2x 8 cores x2).

While hyper-threading is a commonly used feature, it can also degrade performance. Generally, if the VNF/application/workload is the same as that assigned to the logical cores, it is not beneficial to enable HT since there can be resource contention. If there are different VNFs/workloads, HT may provide more efficient use of CPU resources.

Intel's hyper-threading option must be enabled or disabled in the Processor Settings menu in the BIOS system setup. In most cases, HT is enabled by default. Any concerns about contention can be managed with proper core assignment (e.g., assigning the physical and logical core to the same VM).

Modern Intel systems use a high-speed point-to-point technology called QuickPath Interconnect (QPI).

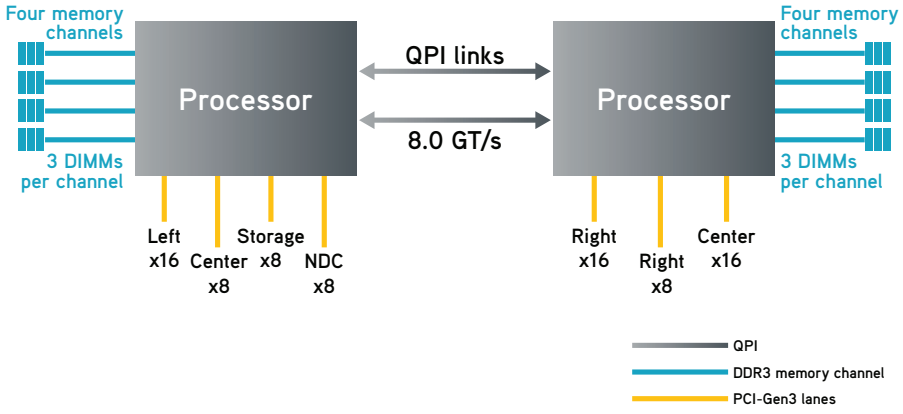


Example BIOS where Intel Hyper-Threading is enabled (highlighted)

QuickPath Interconnect (QPI)

When using a multi-processer server, it is important to understand the architecture of the system and how the processors are interconnected. Modern Intel systems use a high speed point-to-point technology called QuickPath Interconnect (QPI).

A system with two processors will have two QPI connections and provide a speed like 8.0GT/s, which translates to 32Gbps. Traffic I/O from a PCIe to another PCIe slot on the same processor will not need to traverse the QPI, while traffic to a PCIe slot on the second processor will. This can degrade performance due to speed mismatch, buffering and queuing to pass over the QPI. So it is important to understand the architecture of the system and what the PCIe slot assignment is.



Example depiction of processors interconnected with QPI Links

Riser	PCIe Slot	Processor Connection	Height	Length	Link Width	Slot Width
1	1	Processor 2	Low Profile	Half Length	x8	x16
1	2	Processor 2	Low Profile	Half Length	x16	x16
3	3	Processor 1	Low Profile	Half Length	x16	x16

Example table showing PCIe slot to Processor assignment for a specific server

Looking at the table above, provided by a server manufacturer, PCIe slots 1 and 2 are both attached to Processor 2, so traffic traversing between these slots will not be impacted by the QPI.

To get optimal I/O forwarding rates, be sure the devices with the most bandwidth needs are on the same processor. This means knowing the NICs that are in PCIe slots controlled by the same processor.

ixia Expert Tip When executing a "bottom-up" performance test using physical test ports connected to the server's NICs, the best I/O performance will be achieved by using a port pair topology (i.e., pairing ports on NICs in PCIe slots on the same processor). For example, if the system has six network interfaces and they are assigned as follows...

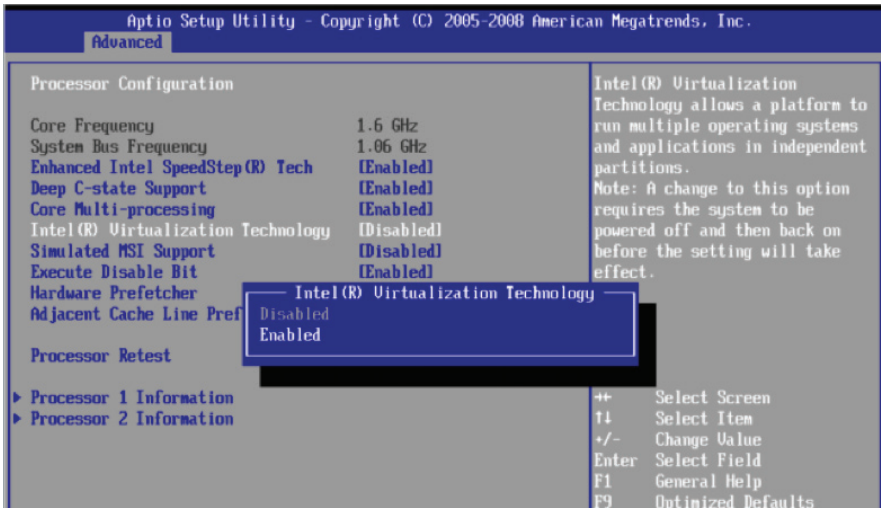
Ports 1-2	PCIe Slot 1	Processor 2
Ports 3-4	PCIe Slot 2	Processor 2
Ports 5-6	PCIe Slot 3	Processor 1

the best results will be using port pairs 1-3, 2-4 and 5-6. If traffic were to be configured with a port topology of 1-5 or 4-6, these examples would require traffic traversing the QPI and result in higher loss and latency, especially when pushing to full line-rate traffic and small packets. A full-mesh traffic pattern would also produce non-optimal results.

Packet Processing Enhancements and Acceleration

Vendors like Intel continue to address the issue of packet processing with enhancements and features. Intel Virtualization Technology (Intel VT) is an important feature that must be enabled on the server platform.

Intel VT is necessary to run virtualization technologies like the KVM hypervisor. This feature is typically enabled by default and can be configured via the BIOS in the chipset settings. Note that BIOS changes typically require restarting the server in order for the changes to take effect.



Example: Enabling Intel Virtualization Technology in the BIOS

Intel VT-d (Intel Virtualization Technology for Direct I/O) makes direct access to a PCI device possible for guest systems with the help of the Input/Output Memory Management Unit (IOMMU). This allows a NIC to be dedicated to a guest VM. The processor, as well as the BIOS, must support Intel VT-d.

NFV places unique requirements on a compute system. Most existing network functions perform well across all packet sizes, including small (64 byte) packets. In addition to

high throughput, it needs to be achieved with low latency and jitter (i.e., delay variation). The typical processing of a packet interrupts a CPU core when the packet arrives at the NIC. The CPU must then examine the packet and determine which VM will process the packet. It then interrupts the core running the target VM. That core copies the packet to the memory space of the target VM. The core interrupted to process the packet can continue with its operation until the next interrupt. As a result, most platforms have poor performance for small packets until getting over 200 bytes.

Network Infrastructure Packet Sizes

Packet Size	64 bytes
10G packets/second	14.88 Million each way
Packet arrival rate	67.2 ns
2 GHz Clock cycles	135 cycles
3 GHz Clock cycles	201cycles

Typical Server Packet Sizes

Packet Size	1024 bytes
10G packets/second	1.2 Million each way
Packet arrival rate	835 ns
2 GHz Clock cycles	1670 cycles
3 GHz Clock cycles	2505 cycles

Example data from Intel processing small packets (64 byte) vs. larger (1024 byte) packets

ixia **Expert Tip** When executing a data plane I/O test, RFC2544 methodology can be followed where a test iteration is run at each of the recommended packet sizes (in bytes): 64, 128, 256, 512, 1024, 1280 and 1518. What you are likely to see is reduced performance at the smaller packet sizes, especially 64 and 128,since they produce the highest frame per second rate. For more favorable results, uses packet sizes above 200 bytes or run a more realistic test using an Internet Mix (IMIX) packet size distribution.

Data Plane Development Kit (DPDK) maps the hardware registers into user space and provides a set of libraries and drivers for fast packet processing. It is an open source BSD licensed project. DPDK runs on Intel x86 processors and can be ported to others. The main libraries consist of:

- Multicore framework
- Huge page memory
- Ring buffers (buffer management)
- NIC poll mode drivers

The libraries are used for:

- Receiving and sending packets within the minimum number of CPU cycles (usually less than 80 cycles)
- Developing fast packet capture algorithms (tcpdump-like)
- unning third-party fast path stacks

DPDK itself is not a networking stack; instead, it is an enabling technology used by companies like 6WIND, Wind River, Calsoft Labs and others. DPDK has also been enabled by open source projects like Open vSwitch.

DPDK improves performance by leveraging the poll mode drivers (instead of interrupt-based). One side effect of this is that the CPU core assigned will run at 100%. It also maps software threads to hardware queues on dedicated CPUs. DPDK also leverages batch packet processing (handling multiple packets at a time). While this can improve throughput,

it may induce some latency, depending on the batch size used. It also uses huge memory pages, which greatly reduces TLB thrashing.

DPDK can be enabled by vSwitch and can also be enabled directly from the VM. Enabling DPDK has shown improvements of 25% or more, especially for small packets.

Commercial offerings like the 6WIND Virtual Accelerator leverage DPDK and provide acceleration for Linux-based network applications. It is transparent to the operating system on the server and can significantly improve packet processing performance for the vSwitch and other network functions. 6WIND also offers VNFs like the Turbo Router and Turbo IPsec Gateway, which are DPDK enabled.

---

ixia **Test Results** Using Ixia's Xcellon-Multis 40GE, along with IxNetwork, 6WIND was able to demonstrate an accelerated virtual switch delivering 195Gbps throughput. This system included Mellanox ConnectX®-3 Pro cards with dual 40G NICs plugged on an HP ProLiant server running the Red Hat Enterprise Linux OpenStack Platform.

http://www.6wind.com/6windgate-performance/virtual-switching/

Another commercial offering is the Titanium Server from Wind River, which provides a complete virtualization stack including an accelerated DPDK-enabled vSwitch. The Titanium Server stack includes an Accelerated vSwitch as well as optional Accelerated Virtual Port (AVP) vNIC driver for the VNFs.

Calsoft Labs offers a suite of high performance VNFs, NFV orchestration framework, and systems integration services for NFV implementation by telecom operators. VNF offerings include: Virtual CPE, Cloud VPN Gateway, Virtual WLAN Controller, Virtual B-RAS and Virtual ADC.

---

ixia **Test Results** Using Ixia's IxLoad, Calsoft was able to verify several key metrics of their Cloud VPN Gateway. This included:

- 100k IPsec + IKE VPN Tunnels
- Tunnel establishment rate of over 1,000 tunnels per second
- Upload and download traffic throughput of 20Gbps full duplex

http://sdn.calsoftlabs.com/downloads/CaseStudies/Ixia-Test-Solutions.pdf

---

When running DPDK, NICs that support DPDK are required. The majority of Intel NICs support DPDK, as do NICs from some other vendors, including Mellanox. Refer to http://dpdk.org/doc/nics for the complete list.

The OpenDataPlane (ODP) project is another option for data plane acceleration. It provides open source, cross-platform APIs which work across multiple architectures.

Storage

While most servers offer various storage and storage controller options, it is important to know the use case and whether it will frequently access the storage. This can justify choosing faster or more scalable storage options.

Typical storage requirements:

- Small/medium scale setups use the LVM/iSCSI on a controller node; dedicated storage nodes are not required. For redundancy, most controllers have a primary drive and a secondary drive (two-disk RAID is suggested). A capacity of 500GB for each drive is sufficient. The controller node storage is used for the guest (VM) images and, when running OpenStack, the configuration database, Cinder volumes and Ceilometer files. A compute node usually needs less storage since VM images and other large files are stored on the controller node. A capacity of 100GB is typically sufficient.
    - Solid State Drives (SSDs) are recommended for high performance
    - Hardware RAID array is recommended for high availability, providing transparent failover and fallback operations
- Large scale setups typically use CEPH storage on dedicated storage nodes. These systems provide high scalability.

---

ixia **Expert Tip** Some NFV-I platforms require two drives (for redundancy purposes). When RAID is used, it makes two drives look like one logical drive. For those platforms to use RAID, you would need four physical disks using RAID to create two logical disks. If four are not available, disabling RAID in the BIOS is required. It is important to understand the NFV platform requirements for storage (e.g., in this case, enabling RAID requires four physical drives in the controller node server).

Network Interface Cards (NICs)

As indicated earlier, modern servers provide a number of PCIe expansion slots required for the NICs. The selection of the NICs is a significant decision and will affect what services can be deployed on the server.

When planning the number of NICs in a server, it is important to consider that:

- One interface is required for management/orchestration
- Often, a second interface is required for OAM functions (for carrier grade systems)
- Two or more interfaces are required for network services

The NICs used for network services often must support advanced features such as SR-IOV, DPDK and others. Meanwhile, NICs used for management can often be lower speed and don't require advanced functions.

Single slot PCIe gigabit adapters come in single, dual and quad port configurations. 10 gigabit adapters come in single or dual slot. Current Intel CPUs have an integrated memory controller and an integrated PCIe controller. This has significantly improved I/O handling.

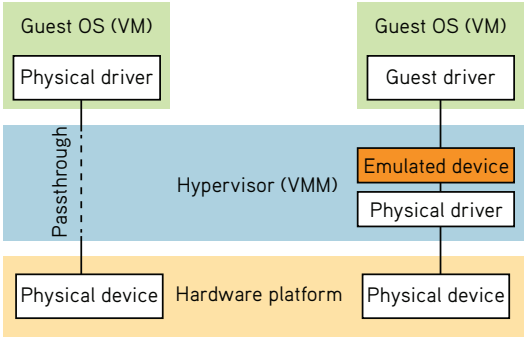
Peripheral Component Interconnect Express (PCIe) is a serial bus and is available in different formats: x1, x2, x4, x8, x12, x16 and x32. The data transmitted over PCIe is sent over lanes in full duplex mode. Each lane is capable of 500 MB/s (PCIe 2.0) scaling from 1 to 32 lanes. This means 32 lanes could support a bandwidth of up to 16Gbps in both directions. PCIe has a version associated with it, with 3.0 being the newest and 2.0 and 1.1 being older versions. It is important to know the version of the PCIe slots on the server and ensure the NICs being ordered are compatible. If 40 GE NICs are being considered, they will require PCIe 3.0.

Also, to ensure the NICs will fit in the server, determine whether the server has full height or low profile slots.

One last item not to be overlooked is ordering compatible transceivers for the NICs. Frequently, NIC vendors only support transceivers that have been tested and certified by them.

PCI Passthrough

PCI Passthrough is a configuration where the hypervisor and vSwitch are bypassed and the VM talks directly to the NIC. This may be required, or at least tested and benchmarked, for performance reasons. If it is believed that the vSwitch or hypervisor are causing significant overhead, this will provide significant I/O performance improvements.



PCI Passthrough bypasses the virtualization layer

Enabling PCI Passthrough means the CPUs provide the means to map PCI physical addresses to guest virtual addresses. When this mapping occurs, the hardware takes care of access (and protection), and the guest operating system can use the device as if it were a non-virtualized system. Both Intel and AMD provide support for device passthrough in their newer processor architectures. Intel calls its option Virtualization Technology for Directed I/O (VT-d), while AMD refers to its device passthrough support as I/O Memory Management Unit (IOMMU). VMware calls this feature Direct Path I/O.

ixia **Expert Tip** When using PCI Passthrough, the interfaces must be added to the VNF instances. This can be done, using the Add New Virtual Hardware -> PCI Host Devices menu in the Virtual Machine Manager. When using KVM, in order for the KVM libvirtd daemon to load the PCI passthrough interfaces, the NICs must be detached from the host OS kernel before the instances boot. This is required using CentOS Linux distribution and can be accomplished using the virsh command line tool with 'virsh nodedev-dettach <linux pci designation>' for each of the interfaces associated with the VNF. Direct I/O assignment will configure a 1:1 VM to physical NIC interface, preventing it from being used by other VMs on the system.

Single Root I/O Virtualization (SR-IOV)

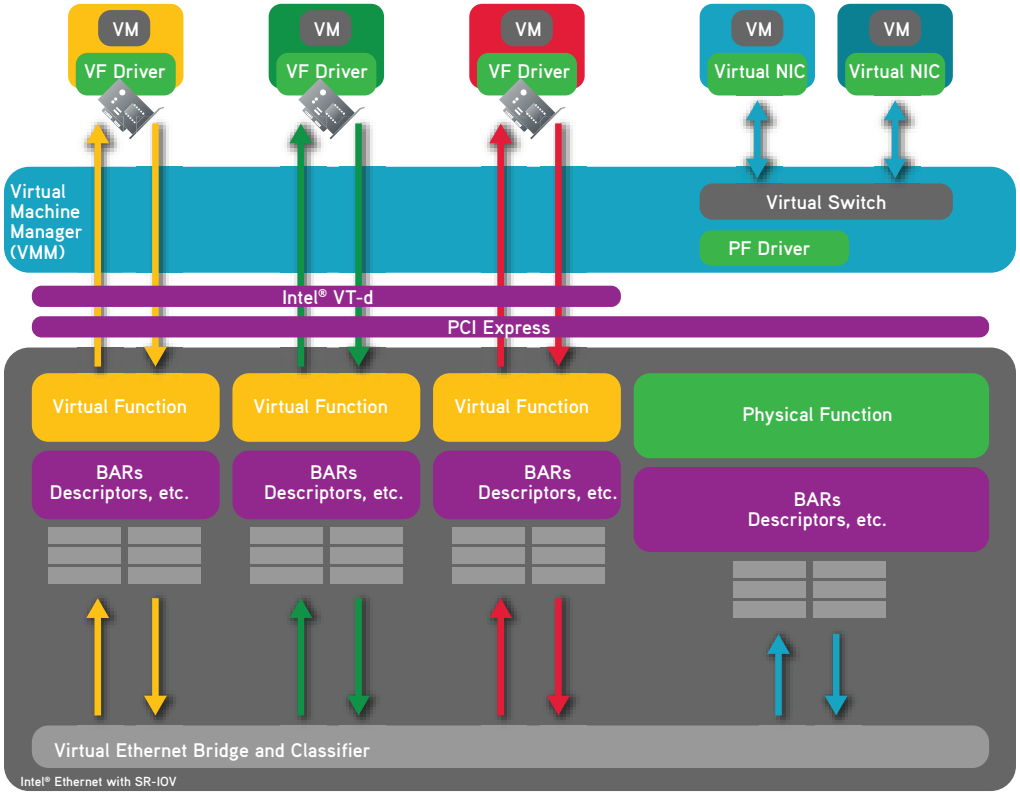
SR-IOV is an extension to the PCI Express (PCIe) specification, which allows a device such as a network adapter to separate access to its resources among various PCIe hardware functions. These functions consist of the following types:

- PCIe Physical Function (PF)
- One or more PCIe Virtual Functions (VFs)

Each PF and VF is assigned a unique PCI Express Requester ID (RID) that allows an I/O memory management unit (IOMMU) to differentiate between different traffic streams and apply memory and interrupt translations between the PF and VFs. This allows traffic streams to be delivered directly to the appropriate VM. As a result, data traffic flows from the PF to VF without affecting other VFs.

An important note is that, when a PCI device is directly assigned to a VM, migration will not be possible without first hot-unplugging the device from the guest. This will impact the portability of the VM. Although it can be achieved, it will require more sophisticated orchestration support.

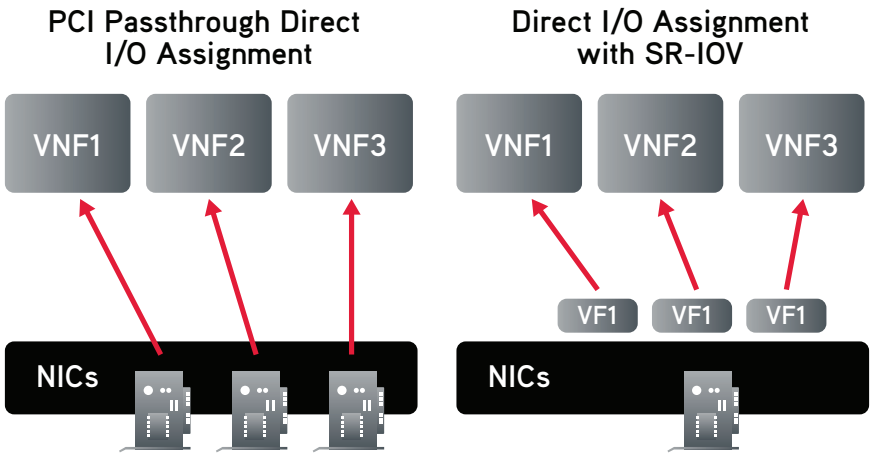
SR-IOV enables network traffic to bypass the software switch layer of the virtualization stack. Because the VF is assigned to a VM, the network traffic flows directly between the VF and VM. As a result, the I/O overhead in the software emulation layer is reduced and achieves network performance that is nearly the same as in non-virtualized environments.



Intel example comparing I/O processing of SR-IOV and traditional virtualization from VM to NIC

Each VM is assigned hardware resources by the hypervisor. The VMs must have specific driver support for SR-IOV and the NIC must support SR-IOV as well. This removes the hypervisor from the process of moving the packet from the NIC to the guest OS (VM). This significantly improves I/O performance and may be required for some VNFs to achieve peak performance.

ixia Expert Tip Using PCI Passthrough or SR-IOV will remove any bottleneck potentially caused by the virtualization layer. If possible, test cases should be run enabling these technologies to determine the best performance that can be achieved by a VNF. The down side is that this limits portability of the VM/VNF. To move it would require more complex orchestration (e.g., detaching the NICs from the host OS kernel before the instance is booted). It also requires the destination server to have these advanced features supported. Bypassing the virtualization layer also means not being able to use the vSwitch for network connectivity and impacts the network design.

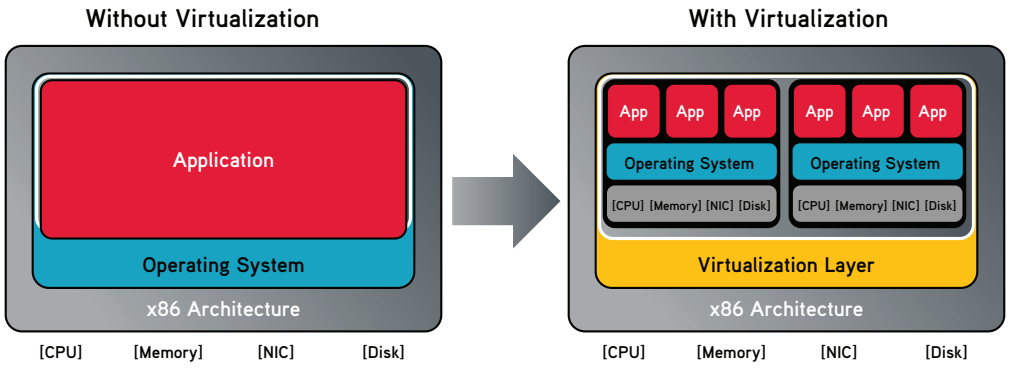


Direct I/O assigns a single NIC to a single VM while using SR-IOV allows direct access from a VM to a virtualized NIC. Both bypass the virtual switch.

Host OS and Hypervisor

When evaluating a platform for NFV, one of the first steps is to select (or at least short list) the virtualization technology that will be used. The selection of the hypervisor will largely dictate the Host OS for the server. NFV platforms are built on server virtualization (Type 1), meaning that the operating system/virtualization layer is installed directly on a bare metal server. It will not run on top of another operating system like Type 2 (desktop virtualization). Service providers driving the requirements for NFV are looking to leverage open source projects and want to avoid vendor lock-in.

A lot of industry effort is focused around the KVM hypervisor, which is part of standard Linux distributions. However, VMware is the leader in this space and, with the most feature rich product offering, VMware is typically what all others are compared to.



Example: Enabling Type 1 server virtualization

Example hypervisors:

- VMWare ESXi
- KVM/QEMU is part of Linux; many distributions are available:
    - Ubuntu
    - CentOS
    - Redhat (RHEL)
    - Fedora
    - Oracle Linux
    - Suse Linux Enterprise
    - Wind River
- Citrix Zen
- Microsoft Hyper-V

Not all hypervisors are created equal. Here are some available features to look for:

- Provides abstraction of the underlying server resources and enables the creation of virtual machines (guests)
- Each VM runs its own OS and is allocated resources such as CPU, memory and storage
- VMs operate independently and can be shut down, restarted and modified without affecting other VMs
- Once a VM is defined, it can be saved using a "snapshot" that can be used for backup and simple redeployment
- VM migration, allowing movement of a VM from one server to another without stopping it
- Changing resource allocation (CPU, memory) on the fly without restarting the VM

Since there are a variety of hypervisors and distributions, there are many variables that can affect the performance of a VNF (and resulting service). Some hypervisors are preconfigured with many optimization features—or mandate that they be used. An example of this is a commercial virtualization stack, which mandates that DPDK be run in the vSwitch.

For KVM, specifically, there are several features that can be enabled to achieve optimized performance. Input/Output Memory Management Unit (IOMMU), mentioned earlier, is one of these features which needs to be enabled in the host OS.

Huge page support is another important feature. This feature refers to changing the amount of memory "pages" that the CPU can mark for use by a process. The typical default page size is 4k bytes. When a process uses 1GB of memory, this causes the CPU and operating system to look up 262144 (1GB/4k) entries. Current CPU architectures support bigger pages (called Huge page support in Linux), so the CPU/OS has fewer entries to look up. By using huge page allocations, performance is increased since fewer pages and Translation Lookaside Buffers (TLBs, high speed translation caches), are needed. This reduces the time it takes to translate a virtual page address to a physical page address. Without huge pages, high TLB miss rates would occur with the standard 4k page size, slowing performance. Huge page support must be enabled on the Host OS.

Enable these features via Linux grub command line arguments during boot. Using CentOS, edit the /boot/grub/grub.conf file for the kernel version being used. An example of the relevant grub line is shown below, with the command line argument additions shown in **bold**.

```
kernel /vmlinuz-3.10.0-123.6.3.el7.x86_64 ro root=LABEL=DATA
nomodeset rd_NO_LUKS KEYBOARDTYPE=pc KEYTABLE=us LANG=en_
US.UTF-8 rd_NO_MD SYSFONT=latarcyrheb-sun16 crashkernel=auto
rd_NO_LVM rd_NO_DM quiet intel_iommu=on default_hugepagesz=1G
hugepagesz=1G hugepages=16
```

This example provisions 16GB of compute system memory to huge pages with a huge page size of 1GB and enables IOMMU. (Note, the host OS must be rebooted for the configuration to take effect.)

If IOMMU has been correctly enabled in the host OS in the grub command line, the output of the Linux dmesg command should include 'Intel-IOMMU: enabled'.

```
[root@ixia ~]# dmesg | grep IOMMU
[0.000000] Intel-IOMMU: enabled
```

Open Source NFV-I Platforms

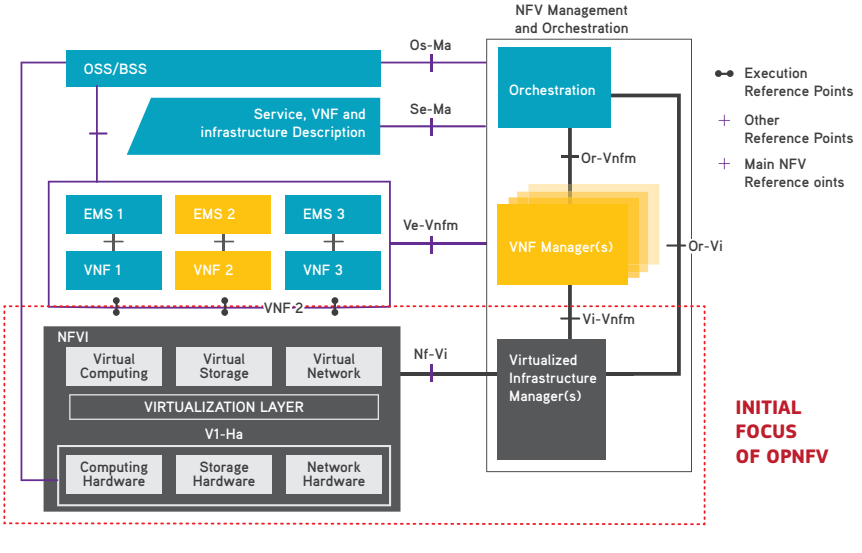
With the complexity of the NFV-I, and all the potential choices of components, it is sometimes difficult to get started or get an initial system working. The NFV ISG is producing specifications, but it is not producing usable software. In an effort to bring together companies interested in providing software integration, the Open Platform for NFV (OPNFV) was created. The OPNFV is first focusing on the NFV-I and is working toward a Spring 2015 release called "Arno." It is working on a stack that is based on open source components. The first "reference stack" they are producing includes:

Host OS	Centos Linux
Hypervisor	KVM
vSwitch	Open vSwitch
Network Controller	Open Daylight
Virtual Infrastructure Manager (VIM)	OpenStack

OPNFV is not specifying these components as the only "Stack." There is work underway to include other hypervisors, network controllers and other components. OPNFV is providing useful integration of these components, which will allow consumers the ability to download a stack that has gone through some verification. More importantly, this community is identifying gaps and issues and communicating them to various upstream communities.

Upstream projects OPNFV is working with:

- Virtual Networking: Open vSwitch
- SDN Controllers: Open Daylight, Open Contrail, ONOS
- Virtual Infrastructure Manager: OpenStack
- Orchestration: OpenMANO



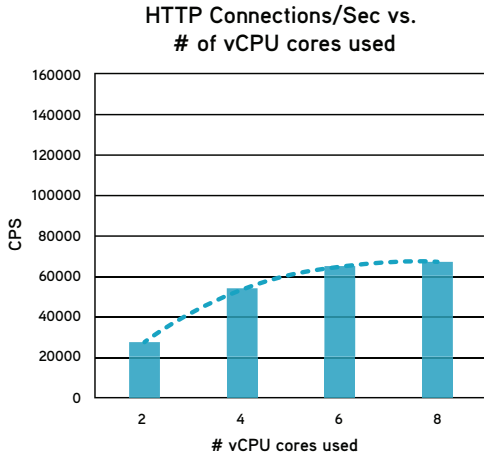
OPNFV is developing a reference stack for the NFV-I

Another open source-based reference platform is Intel's Open Networking Platform (ONP). The Intel ONP project has been operating for a year longer than OPNFV and the first reference stack is already available for download. According to Intel, "It is not a commercial product, but a pre-production reference that drives development and showcase SDN/NFV solution capabilities."

Know the Performance Curve

In the world of networking there is an old adage which says: "If you have a network problem, throw bandwidth at it." This type of statement is usually true in that more bandwidth will make a network problem go away. A similar saying happens in compute environments: increase CPU (processing power) if there are application issues. When applying this to VNF performance, it is important to apply the right number of resources for the desired performance. Most VNF vendors are providing guidelines on performance from a minimum to maximum resource allocation.

ixia Test Results



Using Ixia's IxLoad to test HTTP connections/sec increasing from 2 vCPU to 6 vCPU achieves significant performance improvements. When moving to 8 or beyond, the performance flattens or can even decrease.

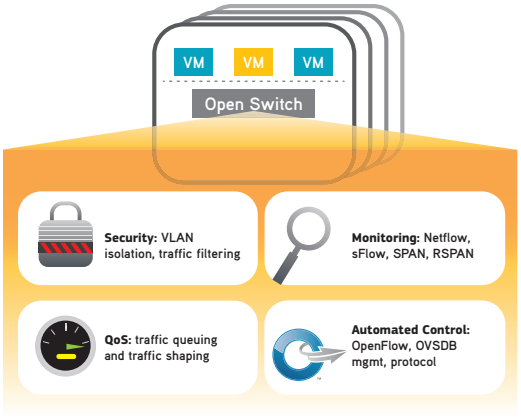
CPU resources will be at a premium and it will be important to allocate the proper amount to each VNF. Over-provisioning a VNF with too much CPU may be wasteful or even degrade the performance.

ixia Expert Tip Instead of scaling up the resources allocated to a VNF another option may be to scale out. An example of this is a vBRAS Ixia tested which required 3 vCPU core and had a capacity of 64k PPP sessions. The test goal was to get to 256K PPP sessions so four vBRAS instances were used and the goal was achieved.

Virtual Networking

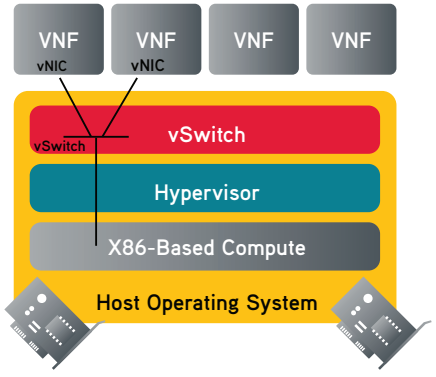
The last section of this paper will address the networking required to enable NFV. Within a Linux OS, there is a default networking facility known as the Linux bridge. This default bridge provides Layer 2 network connectivity and does it relatively fast and reliably. When attempting to use the Linux bridge in more complicated environments, like those enabling NFV, the shortcomings of the Linux bridge quickly appear. One of the primary limitations is that the Linux bridge is not good in the dynamic conditions of a virtualized network where hosts (VMs) are started, stopped and moved around. There is also a significant lack of features for security (like ACLs) or priority (like QoS). The unique needs of networking in a NFV environment have created opportunities for startups as well as new innovations from large networking companies.

One of the products to come out of a networking startup is called Open vSwitch (OVS). OVS is an open source production quality multi-layer virtual switch. It can operate both as a soft switch running within the hypervisor or as the control stack for switching silicon. It has been ported to multiple virtualization platforms and switching chipsets. It provides features for security, QoS, monitoring and automated control.



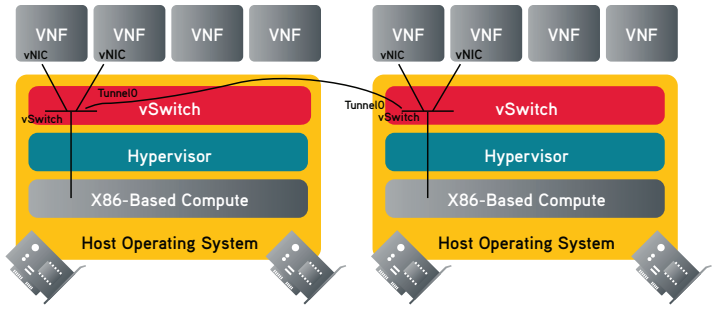
Features of Open vSwitch

OVS is supported in many of the virtualization stacks, including Xen, KVM, VirtualBox and others. Due to its rich feature set and wide availability, it is being used in many NFV PoCs and early deployments.



Virtual networking provides VM-to-VM connectivity and connectivity to the physical NIC

Looking at an example of the most basic use case of virtual networking on a single server, it is used to provide logical connectivity VM to VM (through the definition of a vSwitch using the vNIC interfaces of the VM) as well as connectivity to a physical network adaptor.



VM-to-VM connectivity on separate servers; overlay (tunneling) technologies are often used

Configuring a single server or even two servers can be done manually; however, the goal of using this technology is to enable scale. For that reason, most networking options provide automated control. Two frequently implemented methods are (1) the OpenStack plugin for configuring the network called Neutron or (2) using an SDN technology like OpenFlow and using a SDN controller to configure the required connectivity. The virtual networking technology used will also have a significant impact on the I/O performance of the VNF and service. (Virtual networking is an entire topic all on its own.)

In addition to open source solutions like OVS, there are a number of commercial solutions, including:

- Cisco Nexus 1000V
- Brocade Vayatta vRouter and Vayatta Controller

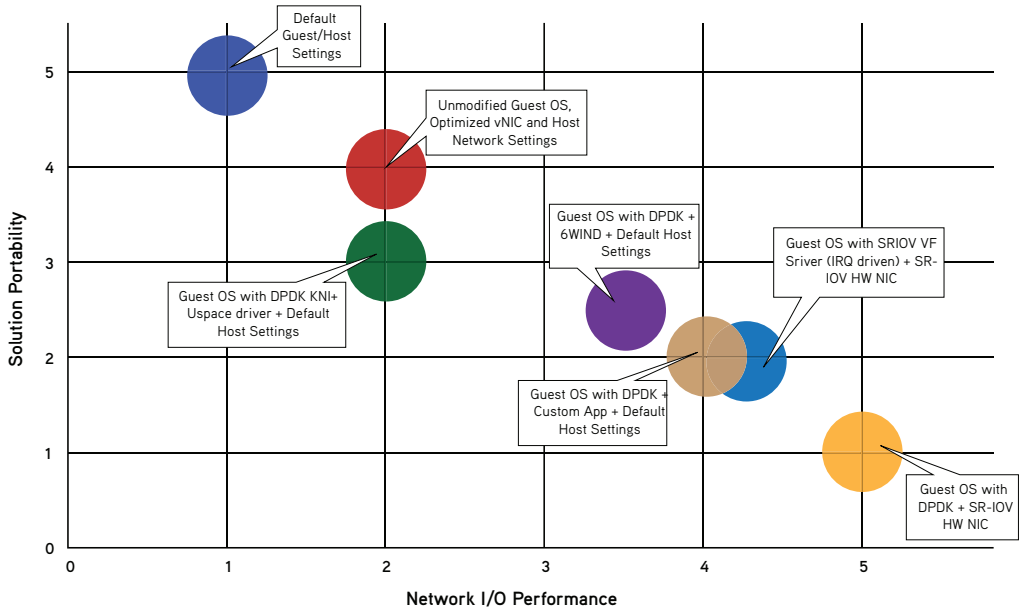
Example overlay-based solutions:

- VMWare NSX
- PLUMgrid Open Networking Suite (ONS)
- Midokura MidoNet

The virtual networking technology is a fundamental component of the NFV-I. This is an expanding area with more established vendors and startups moving into this space. While there are various tradeoffs in performance, functionality and cost, most offer integration with virtual infrastructure managers using an SDN controller or a cloud operating system like OpenStack. The details of virtual networking will be provided in a separate paper.

Conclusion

In this article, we have introduced the components of the NFV-I and many of the configuration options affecting performance and portability of the VNFs. It is clearly a significant challenge for VNF vendors to test and verify their software functions on the various platforms. Service providers face an even more significant challenge when it comes to standardizing an NFV-I platform that can deliver predictable performance for carrier-grade services and SLAs.



Summary chart comparing relative performance to portability

Each hardware, software and configuration option can affect performance. Over tuning for high performance will put more demands on the NFV-I and affect portability. Service providers are starting with applications they can benefit from within a year or two, including vCPE, service chaining (vCPE, vFW, vDPI, vLB). One of the most targeted areas is in the mobile core with vEPC and vIMS. Providers are aggressively looking to move from PoCs to field trials to full service deployments. Functional validations of these technologies have progressed well.

The ETSI ISG is a great proof point, with 36 PoCs, most of which have been successfully demonstrated. The challenges going forward are operationalizing these services and ensuring they can be managed and orchestrated to achieve the expected efficiency.

Whether for an individual function or complex service chain, Key Performance Indicators (KPIs) must be defined so future orchestration of services at scale can give the platform the resources it needs to deliver the required performance. KPIs are also important for service monitoring if dynamic scale up/down is required.

NFV enables dynamic changes from service activation. With this new paradigm, NFV requires a more comprehensive approach to tests.

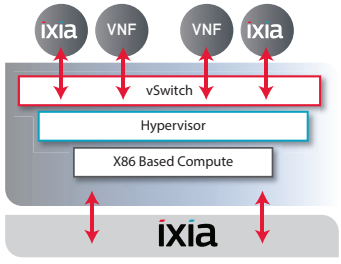
NFV requires an evolved testing cycle to enable dynamic change and service monitoring

Ixia has become a trusted partner for equipment manufacturers and service providers around the world.

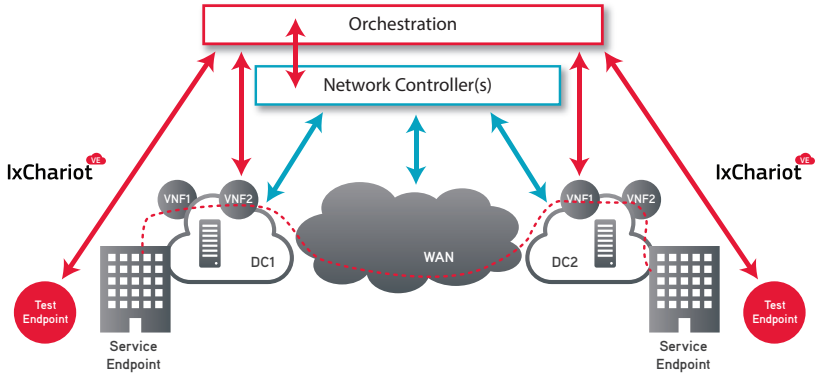
Ixia's traditional physical-based test systems provide "bottom-up" testing of the NFV-I and VNFs. This is required for wire-rate testing as servers move from 10GE to 40GE to 100GE NICs.

Ixia has also virtualized flagship products, providing "Virtual Editions" of IxNetwork (L2-3), IxLoad (L4-7) and BreakingPoint (Security and Application). These can be deployed as VNFs and enable "top-down" testing.

Moving into the monitoring space, Ixia offers a virtual tapping technology called Phantom Virtual Tap.



For service activation and management, Ixia provides test endpoints for service validation (see diagram below). Ixia's IxChariot is a lightweight software endpoint used for validating application performance end-to-end. It supports RESTful APIs for automated testing. An IxChariot endpoint can be preinstalled on almost any OS and can also be deployed as a VM.



Orchestration of service activation and integrated testing is a requirement for service providers

Ixia provides application performance and security resilience solutions so that organizations can validate, secure and optimize their physical and virtual networks. Enterprises, service providers, network equipment manufacturers and governments worldwide rely on Ixia's solutions to deploy new technologies and achieve efficient, secure operation of their networks. Ixia's powerful and versatile solutions, global support and professional services staff allow its customers to focus on exceeding their customers' expectations and achieving better business outcomes.

Learn more at www.ixiacom.com.

About the Author

Michael Haugh has worked in computer networking for nearly twenty years, with roles in network engineering, test engineering, design, system engineering, product management and marketing. He has been with Ixia for eight years and is a Product Marketing Director. He has been working on SDN technologies since 2011 and serves as the chair of the Testing and Interop Working Group for the ONF. He has also led Ixia's outbound efforts on NFV working on several ETSI PoCs, working within the OPNFV and several vendor NFV ecosystems.

References and Acknowledgements

Eddie Arrage, Nitron Labs

Eddie has executed NFV performance tests with Ixia on behalf of SDx Central and has provided valuable input on his experiences with factors that impact performance and portability of a VNF.

Learn more at www.nitronlabs.com.

6WIND www.6wind.com/products/6wind-virtual-accelerator

Calsoft Labs sdn.calsoftlabs.com

Cisco Nexus 1000V www.cisco.com/c/en/us/products/switches/virtual-networking

DPDK www.dpdk.org and www.intel.com/go/DPDK

Dell NFV www.dell.com/learn/us/en/04/large-business/network-functions-virtualization

ETSI ISG for NFV www.etsi.org/nfv

HP Servers http://www8.hp.com/us/en/products/servers/index.html

Intel Network Builders https://networkbuilders.intel.com

Intel Open Networking Platform (ONP) https://01.org/packet-processing/intel%C2%AE-onp-servers

Intel Processors ark.intel.com

Linux KVM www.linux-kvm.org

Midokura MidoNet www.midokura.com

Open Daylight www.opendaylight.org

Open DataPlane Project www.opendataplane.org

Open Platform for NFV www.opnfv.org

Open vSwitch (OVS) www.openvswitch.org

OpenStack www.openvswitch.org

PLUMGrid www.plumgrid.com

Quick Path Interconnect www.intel.com/content/www/us/en/io/quickpath-technology/quick-path-interconnect-introduction-paper.html

VMware NSX www.vmware.com/products/nsx

Wind River www.windriver.com/products/titanium-server

Ixia Worldwide Headquarters
26601 Agoura Rd.
Calabasas, CA 91302

(Toll Free North America)
1.877.367.4942

(Outside North America)
+1.818.871.1800
(Fax) 818.871.1805
www.ixiacom.com

Ixia European Headquarters
Ixia Technologies Europe Ltd
Clarion House, Norreys Drive
Maidenhead SL6 4FL
United Kingdom

Sales +44 1628 408750
(Fax) +44 1628 639916

Ixia Asia Pacific Headquarters
21 Serangoon North Avenue 5
#04-01
Singapore 554864

Sales +65.6332.0125
Fax +65.6332.0127

915-0953-01 Rev. A, May 2015