# Network Performance Comparison of Multiple Virtual Machines

**Alexander Bogdanov** [1]
[1] *Institute forHigh-performance computing and the integrated systems, e-mail: bogdanov@csa.ru, Saint-Petersburg, Russia*

**Pyae Sone Ko Ko**[2], **Kyaw Zaya**[3]
[2,3]*St.Petersburg State Marine Technical University, e-mail: pyaesonekoko@gmail.com, kyawzaya4436@gmail.com, Saint-Petersburg, Russia*

Virtualization technologies have been at the forefront of the planning and implementation efforts for many IT professionals. Network Virtualization represents the most significant innovation in networking since the inventions of Ethernet and the multi-protocol router, two technologies pioneered to level the playing field for device connectivity. This emerging technology will prove to be the key to unlocking $100B in IT expenditures shifting to cloud[1].

This paper provides a brief description of network virtualization and compares the networking performance in multiple virtual machines being used today throughout enterprise networks.

## Introduction

While the benefits of cloud are commonly well represented in current IT discussions – e.g., business velocity, operational efficiency, better capital utilization, new consumption models – the networking barriers inhibiting multi-tenant cloud deployment remain. Inadequacies of earlier attempts to bring together networking and server hypervisor technologies have further exacerbated these issues, causing many enterprises either to enact fragile, static deployments models – or, even worse, bypass the benefits of cloud entirely. We believe the cloud model should extend beyond simple virtualization to include all of infrastructure. Virtualizing the network is the key to making this happen.

Network Virtualization starts with an architecture that decouples the physical from the virtual aspects of a network. Said simply, the physical network continues the work it was designed to do: forward packets, utilizing trusted and well understood routing protocols. The virtual network, however, maintains operational policies, including access control lists, configuration policies, and network services.

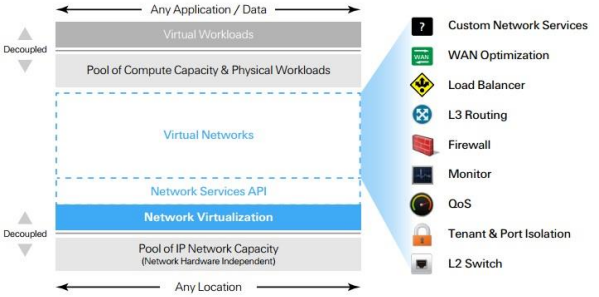



*Figure 1. Network Virtualization*

Network virtualization starts with the creation of multiple virtual networks on top of network hardware. These virtual networks are topologically independent and have the same operational model of virtual machines found in server virtualization. They can be easily set up and torn down with no changes to the physical network substrate. Network virtualization, if done correctly, should be able to run any workload that is compatible with the existing networks, over any type of hardware at any location[2].

## Seven Properties of Network Virtualization

The following list of seven properties must be in place to gain these benefits. Without them, it is not possible to unlock the true potential of cloud.

1. Independence from network hardware

In the emerging multi-tenant cloud, the old rules of vendor lock-in are rapidly changing. A network virtualization platform must be able to operate on top of any network hardware, much like x86

server hypervisors work on top of any server. This independence means the physical network can be supplied by any combination of hardware vendors. Over time, newer architectures that better support virtualization as well as commodity options are becoming available, further improving the capital efficiency of cloud[3].

2. Faithful reproduction of the physical network service model

The vast bulk of enterprise applications have not been written as web applications, and the cost/payback ratio of rewriting tens of billions of dollars of application development is neither realistic nor even possible. Therefore, a network virtualization platform must be able to support any workload that runs within a physical environment today. In order to do so, it must recreate Layer 2 and Layer 3 semantics fully, including support for broadcast and multicast. In addition it must be able to offer higher-level in-network services that are used in networks today such as ACLs, load balancing, and WAN optimization[2].

It is also important that the virtual network solution fully virtualize the network address space. Commonly, virtual networks are migrated from or integrated with physical environments where it is not possible to change the current addresses of the VMs. Therefore, it is important that a virtual network environment not dictate or limit the addresses that can be used within the virtual networks, and that it allows overlapping IP and MAC addresses between virtual networks.

3. Follow operational model of compute virtualization

A key property of compute virtualization is the ability to treat a VM as a soft state, meaning it can be moved, paused, resumed, snapshotted, and rewound to a previous configuration. In order to integrate seamlessly in a virtualized environment, a network virtualization solution must support the same control and flexibility for virtual networks.

4. Compatible with any hypervisor platform

Network virtualization platforms must also be able to work with the full range of server hypervisors, including Xen, XenServer, KVM, ESX, and HyperV, providing the ability to control virtualized network connectivity across any network substrate as well as between hypervisor environments. This “any-to-any” paradigm shift provides for:

- Ÿ More effective utilization of existing network investments,
- Ÿ Cost and management reduction of new, Layer 3 fabric innovations,
- Ÿ Workload portability from enterprise to cloud service provider environments.

5. Secure isolation between virtual networks, the physical network, and the control plane

The promise of multi-tenancy requires maximum utilization of compute, storage and network assets through sharing of the physical infrastructure. It is important that a network virtualization platform maintain this consolidation while still providing the isolation needed by regulatory compliance standards such as PCI or FINRA, as well as provide the same security guarantees of compute virtualization[3].

Like compute virtualization, a network virtualization platform should provide strict address isolation between virtual networks (meaning one virtual network cannot inadvertently address another) as well as address isolation between the virtual networks and the physical network. This last property removes the physical network as an attack target unless the virtualization platform itself is undermined.

6. Cloud performance and scale

Cloud drives a significant increase in the scale of tenants, servers, and applications supported in a single data center. However, current networks are still bound by the physical limitations of networks, especially VLANs (which are limited to 4,096).

VLANS were designed during an earlier era before server virtualization dramatically increased the requirements for the numbers of virtually isolated environments[5]. Network virtualization must support considerably larger scale deployments with tens thousands, or even hundreds of thousands of virtual networks. This not only enables a larger number of tenants, but also supports critical services like disaster recovery, data center utilization, etc., which outstrip current limitations. A virtual network solution should also not introduce any chokepoints or single points of failure into the network. This roughly entails that to all components for the solution must be fully distributed, and all network paths should support multi-pathing and failover[5].

Finally, a network virtualization solution should also not significantly impact the data path performance. The number of lookups on the data path required to implemented network virtualization is similar to what data paths perform today. It is possible to implement full network virtualization in software at the edge of the network and still perform at full 10G line rates.

7. Programmatic network provisioning and control

Traditionally, networks are configured one device at a time, although this can be accelerated through the development of scripts (which emulate the individual configuration). Current approaches make the network configuration slow, error prone and open to security holes through a mistaken keystroke. In a large-scale cloud environment, this introduces a level of fragility and manual configuration costs that hurt service velocity and/or profitability[3].

A network virtualization solution should provide full control over all virtual network resources and allow for these resources to be managed programmatically. This allows the provisioning to happen at the service level versus the element level significantly simplifying provisioning logic and any disruption that might occur due to the physical network node failure. The programmatic API should provide a full access to management and configuration of a virtual network to not only support dynamic provisioning at cloud time scales, but also the ability to introduce and configure services on the fly.

**Benchmarking Methodology**

The network benchmarking tool, netperf 2.4.2, was used for all the experiments. Netperf measures unidirectional network performance for TCP and UDP traffic. It includes support to measure TCP and UDP throughput, using bulk transfers, and end-to-end latencies. Netperf has a client-server model and comprises the following:

- Netperf client, which acts as a data sender
- Netserver process, which acts as a receiver

To check for network performance under different configurations, netperf allows you to specify parameters, such as the socket size and the message size, for the tests[1].

Figure 2 shows the experimental setup for the send experiments where the virtual machine sends data. For the receive experiment, the netperf and netserver processes were exchanged.

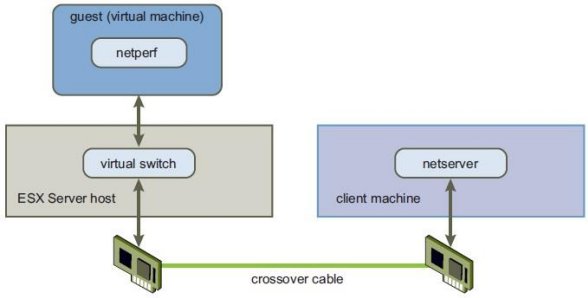



*Figure 2. Experimental Setup for Virtual Machine to Native Send Test*

Figure 3 shows the experimental setup for the virtual machine to virtual machine send experiments.

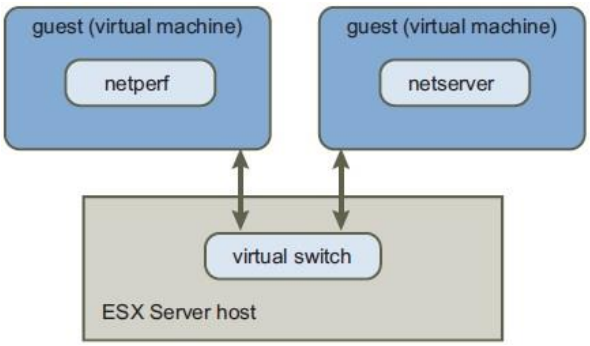



*Figure 3. Experimental Setup for Virtual Machine to Virtual Machine Send Test*

We looked at loading up a single virtual machine (VM) with multiple netperf instances, each running over its own NIC (Network Interface Controller). This effectively exposes the real virtualization overhead of high-throughput networking. While there are some real-world use cases that require this much network bandwidth in a single VM, a much more common scenario is spreading this bandwidth over many VMs running on one physical machine. This is a natural result of consolidating servers. For this paper we used the same hardware and software as in the multi-NIC paper, but performed a "scale-out" test: each of several VMs had a 1 Gbps physical NIC dedicated to it and each communicated to a similar dedicated NIC on the client machine through a netperf/netserver pair. The VMs did not share NICs. We hope this will lead to a better understanding of the performance issues involved with virtualizing networking[3].

Throughput results for send and receive are shown in the two figures below. "Number of NICs/VMs" refers both to the total number of NICs used in the physical machine and to the number of virtual machines for the hypervisor cases. In the native send case when using all four NICs the client started to become a bottleneck instead of the server. For just this case the fourth NIC was moved to a second client. The performance improvement with this change was about 2%. Use of a single client did not cause a bottleneck in the virtualized cases or in the native receive case.

The native tests achieved full wire speed using up to four NICs. The ESX301 tests stayed close to native, falling off to 88-90% of wire speed for four VMs. The XE320 tests achieved maximum throughput at three VMs and then slowed down at four VMs to 61% of wire speed for the send case and 57% for the receive case. One reason for this was CPU saturation in dom0 (Xen's privileged domain, where the backend drivers live), which is only uniprocessor. ESX Server is designed to avoid any CPU bottlenecks in the virtualization layer, and, thus, it scales much better[5].

*Figure 4. Netperf Send Results*

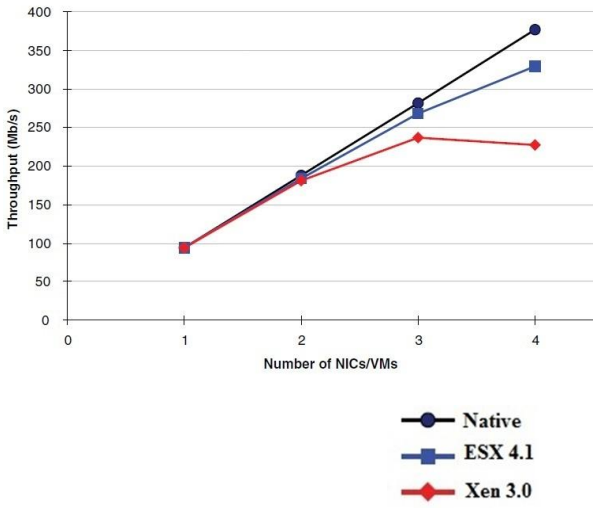
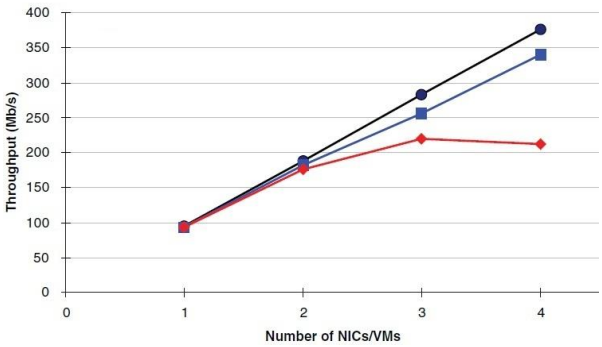



*Figure 5. Netserver Receive Results*




## Conclusion

The spread of virtualization technologies across an enterprise infrastructure has brought about exciting times. IT professionals are now able to maximize computing resources and provide a higher level of flexibility and manageability for desktops, applications, and devices that function as true "network" services. However, the

introduction of these new virtualization technologies onto an existing network infrastructure requires those same IT professionals to understand (1) how well their existing network handles increased traffic demands, (2) any impacts to the network from the new virtualization services, and (3) what types of solutions are available to ensure these new services run effectively and efficiently. The good news is there are a variety of high-performance networking and network virtualization solutions available today that can help.

**References**


1) J. Liu, W. Huang, B. Abali, and D. K. Panda, "High Performance VMM-Bypass I/O in Virtual Machines," in Proceedings of USENIX '06, 2006.
2) Qumranet, "White Paper: KVM Kernel-based Virtualization Driver," Qumranet, Tech. Rep., 2006.
3) P.-H. Kamp and R. N. M. Watson, "Jails: Confining the Omnipotent Root," in Proc. 2nd Int. SANE Conference, 2000.
4) John W. Rittinghouse and James F. Ransome. Cloud Computing: Implementation, Management, and Security. Page – 81.
5) VMWare, "VMWare Server," 2006, http://www.vmware.com.