VMware vSphere™ 4.1 Networking Performance

April 2011

PERFORMANCE STUDY

vmware®

Table of Contents

Introduction

The growth of the web-connected and multi-tiered enterprise applications over the last two decades has caused a steady increase in the demand for network throughput. As network-intensive applications are virtualized and moved onto public and private clouds, the underlying software platform must be able to supply them the full power of today's multi-core servers and high-speed interconnects. VMware vSphere 4.1 provides a stable, high performance virtualization platform that allows multiple virtual machines (VMs) to share hardware resources concurrently with minimal impact on performance. vSphere's highly optimized networking stack, high performance paravirtualized networking devices, and NUMA-aware scheduler, ensure the highest levels of networking performance.

This paper demonstrates that vSphere 4.1 is capable of meeting the performance demands of today's thoughput-intensive networking applications. The paper presents the results of experiments that used standard benchmarks to measure the networking performance of different operating systems in various configurations. These experiments:

- Examine the performance of VMs by looking at VMs that are communicating with external hosts and are communicating among each other.
- Demonstrate how varying the number of vCPUs and vNICs per VM influences performance.
- Show the scalability results of overcommitting the number of physical cores on a system by adding four 1-vCPU VMs for every core.

Executive Summary

The networking performance of several aspects of vSphere communications were measured using standard benchmarks. Results show that:

- Due to performance improvements in vSphere 4.1, a single VM can saturate a 10Gbps link in both transmit and receive cases.
- Using regular TCP/IP sockets, virtual machines residing on the same vSphere host can communicate at rates of up to 27Gbps. This is a 10 times increase in throughput from VMware ESX™ 3.5.
- vSphere networking performance scales well even when the system is overcommitted in terms of CPU.
- A single VM with multiple virtual NICs running on vSphere can easily saturate the bandwidth of four 10Gbps NICs (the maximum number of 10Gbps NICs the test machine supported) for a cumulative throughput of more than 35Gbps.

Performance Enhancements in vSphere 4.1

Asynchronous Transmits

In earlier vSphere releases, a portion of packet transmissions were synchronous. Synchronous transmissions are not desirable because they stall the execution of the VM for the duration of the transmission. In vSphere 4.1, all packet transmissions are completely asynchronous. This allows the VM to make progress while packets are being transmitted and also allows packet processing to be moved to idle cores on the system.

Large Receive Offloads

For improved receive performance, vSphere 4.1 now supports both hardware and software Large Receive Offload (LRO) for Linux VMs. This allows vSphere to aggregate incoming packets before forwarding them to the VMs, thus reducing the amount of packet processing that the VM needs to perform. The CPU savings resulting from

fewer interrupts and fewer packets transferred to the VM translate to improved receive throughput in the VMs. This feature enables 1-vCPU VMs running on vSphere 4.1 to receive data at line rates of over 10Gbps network cards.

VMXNET3 Virtual NIC

VMXNET3 is a paravirtualized network driver that was designed with performance in mind. An earlier study lists the performance benefits of VMXNET3 [1]. To summarize, VMXNET3 supports a larger number of transmit and receive buffers than the previous generations of VMware's virtual network devices. The large buffer sizes reduce packet losses in very bursty workloads. VMXNET3 also supports Receive-Side Scaling (RSS) for the latest Windows operating systems. RSS allows these operating systems to spread the task of packet processing across multiple cores.

Although VMXNET3 was introduced in vSphere 4, it is now bundled with SUSE Linux 11 and other Linux distributions for the best out-of-the-box performance.

Experimental Setup

The benchmarking tool netperf version 2.4.5 [2] was used for all of the tests. Netperf is a small, lightweight benchmark that can be used to measure the throughput during unidirectional bulk transfers as well as the performance of bursty TCP or UDP traffic. Netperf has a simple client-server model which consists of the following two components:

- Netperf client that acts as the data source
- Netserver process that functions as the data sink

This paper focuses on multi-session throughput during the tests. A custom framework was developed to invoke multiple sessions of netperf and the tests were synchronized across multiple VMs. In order to ensure that the measurements were representative of the test, each test case was run three times and the average was reported. The single vCPU (VM-to-VM and VM-to-native) tests used five simultaneous sessions per VM because, in the small socket–small message size cases, a single netperf session is incapable of saturating a 10GbE link. When scaling up the number of virtual NICs per VM, for consistency, the test used five sessions per virtual NIC configured in the VM. Each virtual NIC was connected to a different physical NIC.

To get a fair idea of the small packet performance of vSphere 4.1 for the small packet size test cases, which had a message size of less than 1KB, netperf was run with TCP_NODELAY enabled. This allowed the performance under extremely high packet rates to be measured. The experiments used two 64-bit operating systems—SUSE Linux 11 SP1 and Windows Server 2008 R2. The latest version of VMware Tools was installed in both operating systems. Both VMs were configured with VMXNET3 virtual devices.

For the VM-native tests, the experimental setup consisted of three machines—one vSphere server and two client machines running Linux. The server machine contained two dual-port 10Gbps NICs which were connected to two NICs in each client machine. All tests ensured that the load was equally balanced across available NICs and the two client machines; for example, in the 24 VM tests, each client machine communicated with 12 VMs, and 6 VMs shared one physical NIC.

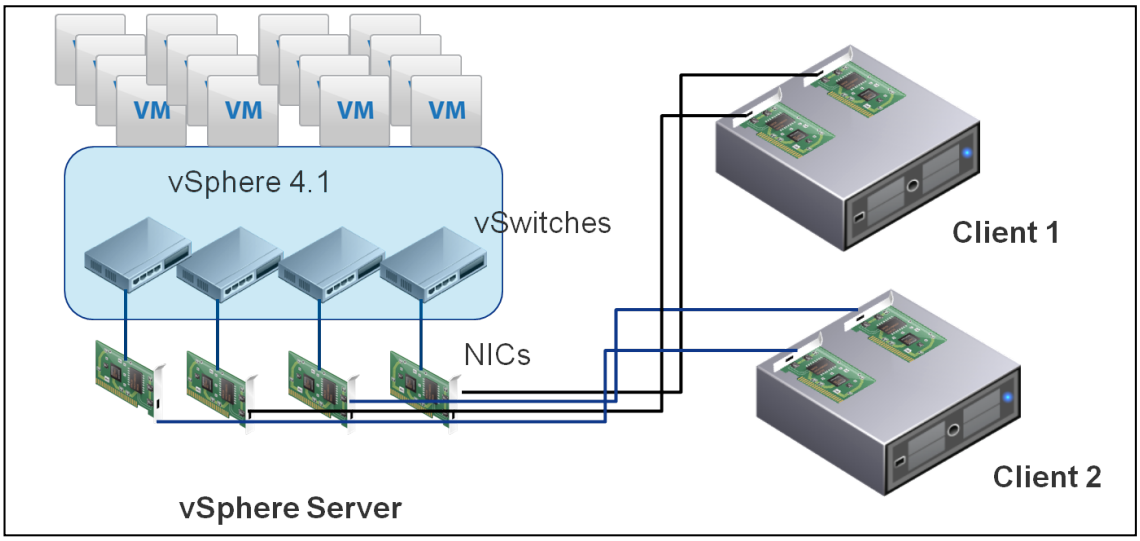

Figure 1. Hardware Setup for Multi-VM Tests

Figure 1 presents a schematic diagram of the setup. Table 1 presents the hardware specifications, while Table 2 shows the VM specifications.

	SERVER	CLIENTS
Make	HP DL 380 G6	HP DL 380 G6
CPU	2x Intel Xeon X5560 @ 2.80GHz	2x Intel Xeon E5520 @ 2.27GHz
RAM	24GB	12GB
NICs	2x Intel x520 2-port 10Gbps Adapter	2x Intel 82598EB 10Gbps Adapter
OS	vSphere 4.1	Red Hat Linux 5.3 64-bit

Table 1. Hardware Setup Details

	LINUX VM	WINDOWS VM
OS	SUSE 11 Service Pack 1	Windows Server 2008 R2 64-bit
RAM	512MB	2GB (single VM), 512MB (multi-VM)
vNIC	VMXNET3	VMXNET3

Table 2. VM Configuration

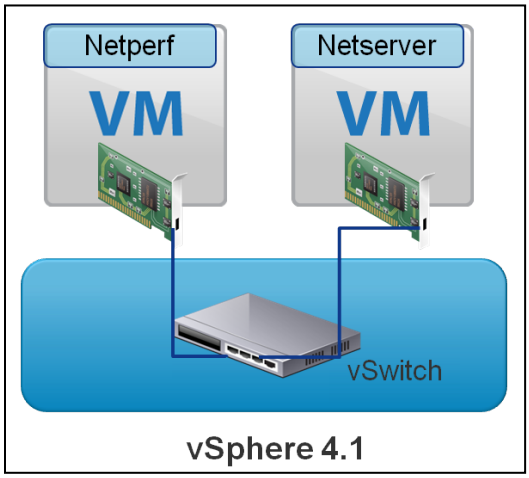

Figure 2. Hardware Setup for VM-to-VM Tests

For the VM-to-VM tests, the two VMs were connected to the same vSwitch on the same vSphere host, as shown in Figure 2.

To ensure that the PCI bus on any of the machines did not become a bottleneck, the fastest available PCIe Gen2 slots were available in the machines. The dual-port NICs in the vSphere server were inserted in PCIe-x16 slots, whereas the NICs in the client machines were installed in PCIe-x8 slots. The network card used in the server was based on PCIe Gen2 and was thus capable of sending and receiving traffic at line rate on both interfaces.

Multi-VM tests used non-persistent linked clones of the original VMs that were used in the single-VM tests. This was done only to save on storage space and should not have any bearing on the final results of the tests.

vSphere was not tuned in any way for the tests. In the multi-VM tests with over-committed CPU, the VMs were neither pinned nor were any kind of CPU reservations used. The tests were limited to four 10Gbps NICs on the server because of hardware limitations (limited number of slots). Guest VMs were scaled up to only four virtual NICs to match the number of physical NICs available on the host.

Results

VM-to-VM Performance

As the number of cores per socket increases in today's machines, the number of VMs that can be deployed on a host also goes up. In cloud environments with massively multi-core systems, VM-to-VM communication will become increasingly common. Unlike VM-to-native machine communication, VM-to-VM throughput is not limited by the speed of network cards. It solely depends on the CPU architecture and clock speeds.

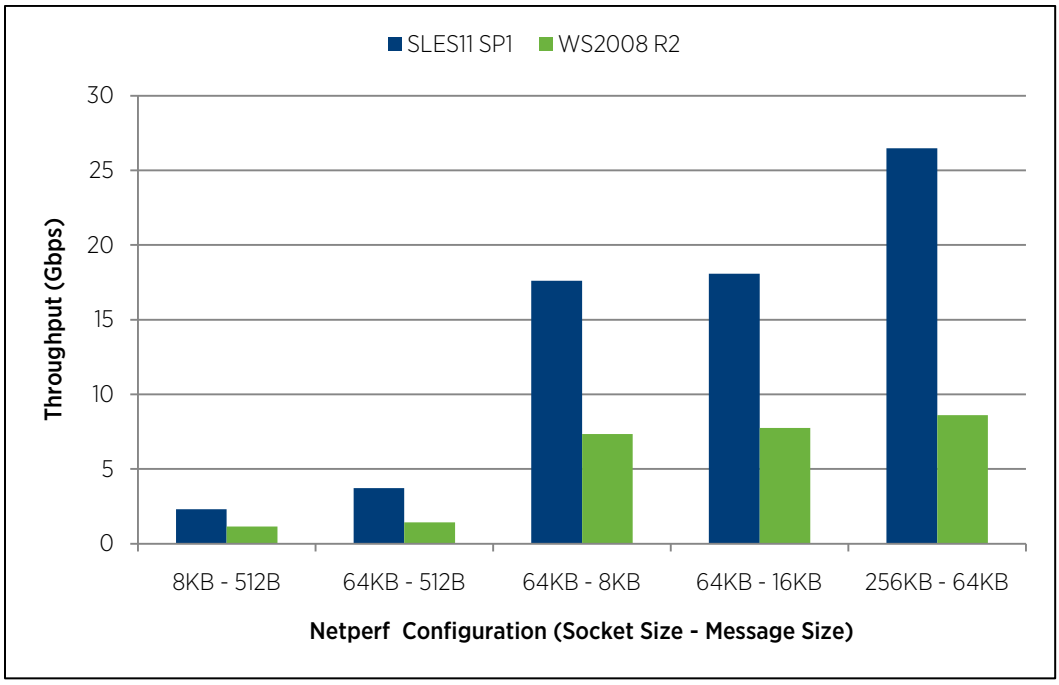

Figure 3. VM-to-VM Performance

Figure 3 presents the throughput results for both the Linux and Windows VMs. The figure shows the Linux VM-to-VM throughput approaches 27Gbps, which is nearly three times the rate supported by the 10Gbps network cards available in the market today. Not only is the VM-to-VM performance outstanding in absolute terms, it has also improved nearly 10 times over the performance in ESX 3.5 [3]. Today, the VM-to-VM throughput obtained using regular TCP/IP sockets is comparable to the throughput obtained using specialized VMCI sockets [4].

Note that in the VM-to-VM case, the throughput between the VMs is limited by the CPU available to the VM that is receiving the packets because it is more CPU-intensive to receive packets than to send them. Also, there is a substantial difference between Windows performance and Linux performance because SUSE Linux supports LRO, whereas Windows does not. In the Windows VM-to-VM tests, up to 1500-byte (standard Ethernet MTU size) packets are transferred from one VM to the other. In contrast, LRO allows packets of up to 64KB in size to be transferred from one Linux VM to the other. This reduces the number of packets that need to be processed by the receiving VM, thus improving networking performance. This disparity in Windows and Linux receive performance is evident in all the test results presented in the rest of this paper.

VM-to-Native Performance

Real world applications are not expected to be running only one VM on a given vSphere server. However, single VM performance is of interest because it demonstrates the following:

- Minimal impact of virtualization on the throughput of applications
- Packet processing capabilities of vSphere 4.1

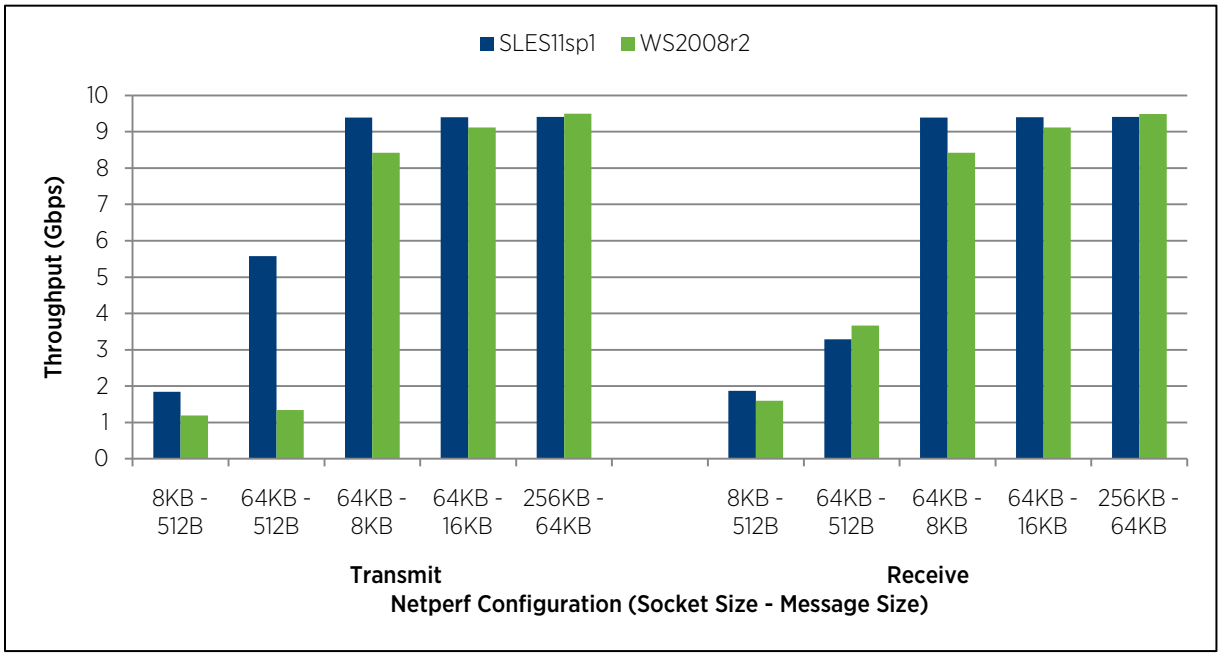

Figure 4. 1-vCPU, Single-VM Performance

Figure 4 shows the transmit and receive throughput for Linux and Windows VMs for various netperf configurations. In configurations with large socket and message sizes, both Linux and Windows VMs are able to saturate a 10Gbps link in either direction. The throughput in the small message size cases (that is, where message size is 512 bytes) is lower than those in the large message size cases because these tests were run with Nagle's algorithm disabled. Nagle's algorithm, which is on by default in netperf, improves the performance of TCP/IP applications by reducing the number of packets that are sent over the network. Nagle's algorithm was disabled for the small message size tests to get a fair idea of the packet processing power of vSphere 4.1. On this host, both Linux and Windows VMs can process upwards of 800,000 512-byte packets per second.

Multi-vCPU and Multi-vNIC Scalability

Depending on application requirements, VMs may be deployed with multiple vCPUs and (or) multiple virtual NICs. vSphere 4.1 supports up to four virtual NICs per VM and up to four 10GbE network cards per host. Most environments will not have a single VM connected to four 10GbE interfaces, however, these experiments help readers better understand the limits of single VM networking performance on a vSphere host.

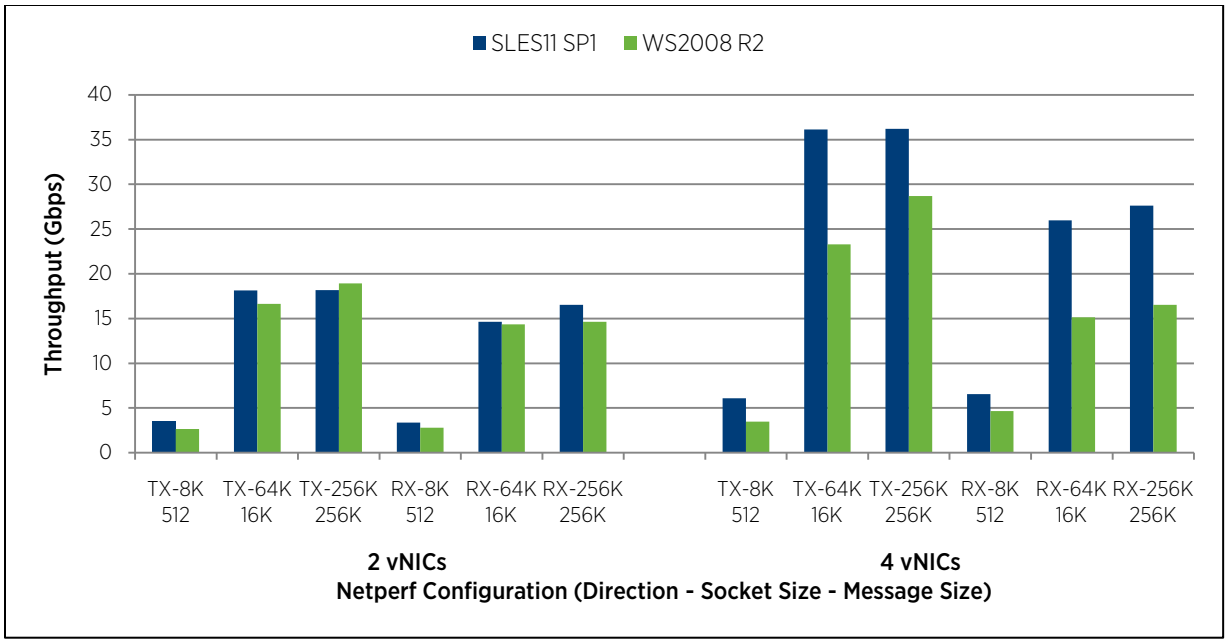

Figure 5. Single VM - Multi vNIC Performance

To better understand how much traffic a single VM could handle, experiments were run with a single 4-vCPU VM that was configured with multiple virtual NICs. In these experiments, each virtual NIC was connected to a different physical NIC. For this particular experiment, Receive-Side Scaling (RSS) was enabled for each VMXNET3 NIC in the Windows VM. RSS is a feature in Windows Server 2008 that allows the receive-side processing of packets to be spread across multiple processors [5]. Without RSS enabled, the receive performance of a multi-vCPU Windows Server 2008 VM would be similar to that of a 1-vCPU VM. However, enabling RSS in Windows has small performance overheads that result in increased CPU consumption. Since RSS benefits Windows only when multiple CPUs are available, RSS was not enabled for all other single-VM and multi-VM tests discussed in this paper.

Figure 5 presents the results of the single VM scaling experiments. The figure shows that a single VM running on vSphere 4.1 can drive enough traffic to saturate four 10Gbps NICs. This particular experiment was limited by the number of PCIe-x8 slots (and hence the number of 10Gbps NICs) available on the system.

A single Linux VM can process incoming networking traffic at a sustained rate of more than 27Gbps, whereas a single Windows VM can sustain traffic rates of approximately 16Gbps in the receive direction. As discussed earlier, the networking performance of Windows is limited by the absence of LRO.

Note: The results presented in the graphs are the medians of three runs. Enabling RSS on a 4-vCPU Windows Server 2008 VM introduces some variance in performance and the best throughput numbers in the tests were typically much higher than the median numbers. For example, in the 4-vNIC tests, throughput rates of up to 20Gbps in the receive direction were observed (as compared to 16Gbps in Figure 5).

Multi-VM Performance

The multi-VM case presents a common deployment scenario in which multiple server VMs are deployed on the same host and serve clients connecting from different machines. The networking performance of up to 32 single vCPU VMs simultaneously running on an 8-core system was measured.

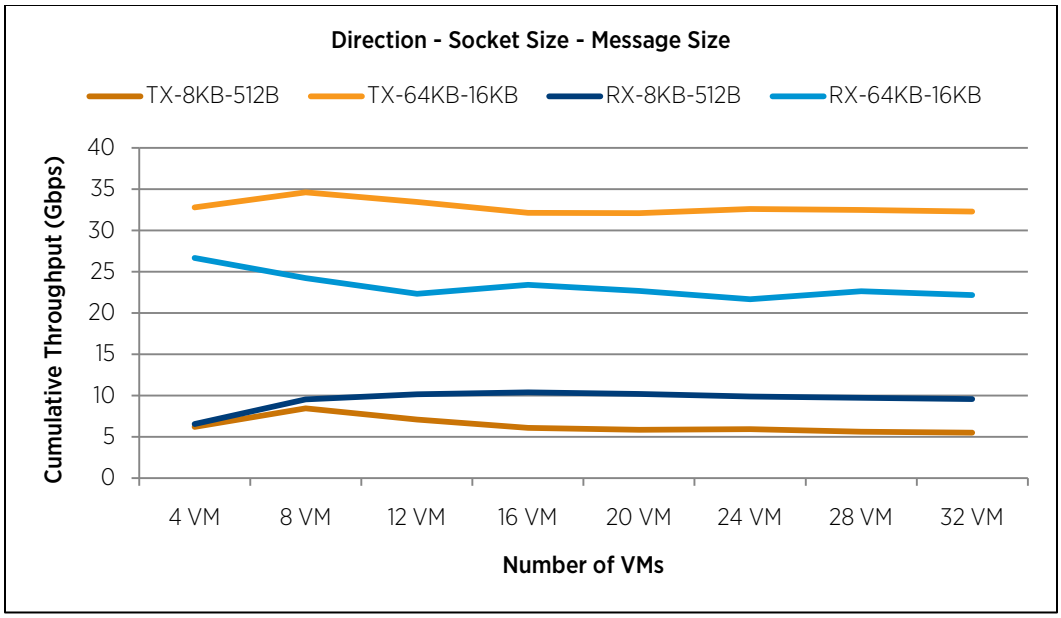

Figure 6. SLES 11 SP1 Multi-VM Performance

Figure 6 shows the aggregate throughput when multiple Linux VMs share four 10Gbps NICs on a vSphere host. The first thing to notice is that four 1-vCPU VMs are sufficient to saturate four 10Gbps NICs simultaneously. As discussed in the previous sections, a single VM with 4 vCPUs can also push data at 36Gbps. Thus, networking performance scales well with both an increase in vCPUs and an increase in VMs on a given system.

Figure 6 also shows that, in most cases, performance peaks when there are eight 1-vCPU VMs booted up on the host—that is, when the CPU-consolidation ratio is 1. This is expected because, beyond this point, there are more VMs on the host than there are available CPUs, which causes some contention for CPU resources. After the 12-VM data point, the network performance curves flatten out and there is no perceivable change in performance when the number of VMs is doubled from 16 to 32.

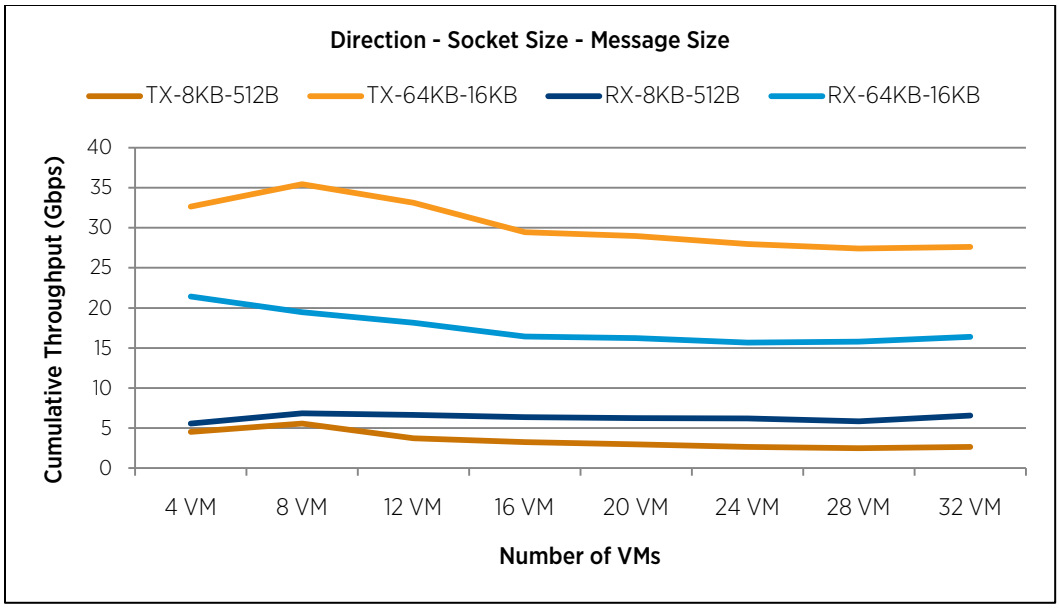

Figure 7. Windows 2008 R2 Multi-VM Performance

As in the case of the Linux multi-VM tests, the performance of multi-VM Windows tests (Figure 7) also peaks when eight VMs are booted up on the 8-core machine. This is followed by a dip in performance and the throughput in a majority of the cases stabilizes after the 16-VM case.

Note that the multi-VM experiments are trying to analyze an unrealistic, worst case scenario in which all VMs are trying to push (or are receiving) large amounts of traffic at the same time. Under these extreme artificial loads, the flat lines after a certain point in the graph indicate that even when an increasing number of VMs are competing for CPU and networking resources on the same machine, there is minimal impact on the performance of the system. Networking performance on vSphere 4.1 thus scales gracefully with increasing load on the system. VMware performance engineers expect that real-life low bandwidth workloads on vSphere 4.1 will scale much better than the benchmarks that stress their system.

Conclusion

The results in the previous sections prove that vSphere 4.1 is a high-performance platform with superior networking performance. Two VMs running on the same host can communicate with each other at rates of up to 27Gbps using only one virtual NIC. A single 4-vCPU VM running on vSphere 4.1 can push the limits of commodity x86 hardware (with only enough slots for four 10GbE NICs) by sending data at a rate of 36Gbps. The networking performance of vSphere 4.1 not only scales with increasing vCPUs, but also with increasing numbers of VMs. Networking performance is so consistent on highly loaded systems that, in most cases, there was no drop in performance observed when the number of VMs per physical CPU were doubled from two to four. vSphere 4.1 is the perfect platform for virtualizing even those applications which place substantial demands on the networking infrastructure.

References

[1] *Performance Evaluation of vmxnet3 Virtual Network Device*. VMware Inc., 2009. http://www.vmware.com/resources/techresources/10065.

[2] Jones, Rick. *Care and Feeding of Netperf 2.4.x*. Netperf. http://www.netperf.org/svn/netperf2/tags/netperf-2.4.5/doc/netperf.html.

[3] *Networking Performance in VMware ESX Server 3.5*. VMware Inc., 2008. http://www.vmware.com/resources/techresources/1041.

[4] Walha, Bhavjit. *VMCI Socket Performance*. VMware Inc., 2009. http://www.vmware.com/resources/techresources/10075.

[5] "Receive-Side Scaling," *Networking and Access Technologies*. Microsoft. http://technet.microsoft.com/en-us/network/dd277646.aspx.

[6] *10Gbps Networking Performance on ESX 3.5 Update 1*. VMware, Inc., 2008. http://www.vmware.com/resources/techresources/1078.

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com