

Performance Comparison of Virtual Network Devices

VMware® ESX Server 3.5

The networking performance of a virtual machine is greatly influenced by the choice of virtual network devices in the virtual machine and the physical devices configured on the host machine. ESX Server 3.5 supports multiple virtual network devices (vlance, e1000, vmxnet), each with its own usability advantages and performance benefits. It is clear that vlance is not the best choice for high-performance workloads. However, many users are still uncertain about the performance differences between e1000 and vmxnet virtual network devices.

This paper introduces the available virtual network devices in ESX Server 3.5 and discusses the performance differences between them. The goal of this paper is to provide performance data to help customers choose the right virtual network device for their configuration.

This study covers the following topics:

- [“Virtual Network Devices in ESX Server 3.5”](#) on page 1
- [“Benchmarking Methodology”](#) on page 2
- [“Performance Results”](#) on page 3
- [“Latency Results”](#) on page 5
- [“Conclusion”](#) on page 6
- [“References”](#) on page 6

Virtual Network Devices in ESX Server 3.5

VMware ESX Server supports the following virtual network devices:

- vlance - AMD® Lance PCNet32® ethernet adapter
- e1000 - Intel® e1000 ethernet adapter
- vmxnet - VMware high speed virtual ethernet adapter

Support for the vlance virtual network adapter is included with most operating systems, including 32-bit and 64-bit. Therefore, the vlance virtual adapter can be used without installing additional drivers. Vlance has been replaced by e1000, which is the only supported device on 32-bit Microsoft® Windows Vista®, for all 64-bit guests. The e1000 device driver is also bundled with most operating systems and provides ready-to-use network connectivity and improved performance over the vlance device.

Vmxnet, a paravirtualized device used for high I/O performance, was previously supported for most 32-bit guest operating systems. The user needs to install VMware Tools to access the vmxnet driver; in the absence of VMware tools, the device behaves like a vlance device.

ESX Server 3.5 introduces the enhanced-vmxnet device for a majority of operating systems. The enhanced vmxnet device supports 64-bit operating systems and also adds support for advanced features, such as TCP Segmentation Offload (TSO) and Jumbo Frames. In this paper, all vmxnet based experiments were done using virtual machines that were configured with an enhanced-vmxnet virtual device. The user must install VMware Tools to access the enhanced-vmxnet driver.

In ESX Server, there are differences in the way network I/O is triggered for the e1000 and vmxnet virtual network devices. ESX Server needs to emulate the complete e1000 device in the software so that guests with unmodified e1000 drivers can interact with the device. Each transmit or receive call causes access to a number of virtual device registers. This level of access is expensive because each instance causes an address space switch. Attempts to minimize this performance hit can introduce latencies in the I/O path. Also, the number of possible optimizations to e1000 is limited by the fact that the driver code cannot be modified.

The vmxnet driver, however, is a virtualization-aware driver. The device and the driver have been designed with virtualization in mind. Both the device driver and the virtual device code have been optimized to reduce the number of virtual register accesses and provide good performance through polling and direct calls into the VMkernel. As a result, vmxnet exhibits lower latencies than e1000.

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

Transmit and receive paths might exhibit different levels of performance in a virtual machine and in a native system.

Depending on configuration options, such as IP protocol type, socket buffer size, and message size, the benchmark can generate different traffic patterns. For this paper, we used the TCP protocol. We considered the following performance metrics:

- Throughput

This metric was measured using the netperf TCP_STREAM tests. We ran three parallel netperf processes and measured the aggregate throughput.

- Latency

This metric was measured using the netperf TCP Request-Response (TCP_RR) test. It reports the number of request-responses per seconds, which is inversely proportional to latency.

For more details on these tests, see the netperf documentation.

The details of the experimental setup are presented in [Table 1](#). All virtual machines in the experiments used a uniprocessor hardware abstraction layer (HAL) and were configured with one virtual CPU and 512 MB RAM.

The experimental test bed comprises the following physical hosts:

- Server machine running ESX Server 3.5
- Client machine running RHEL5 natively

The 1 Gbps and 10 Gbps physical NICs on the ESX Server host were connected through crossover cables to identical NICs on the client machine.

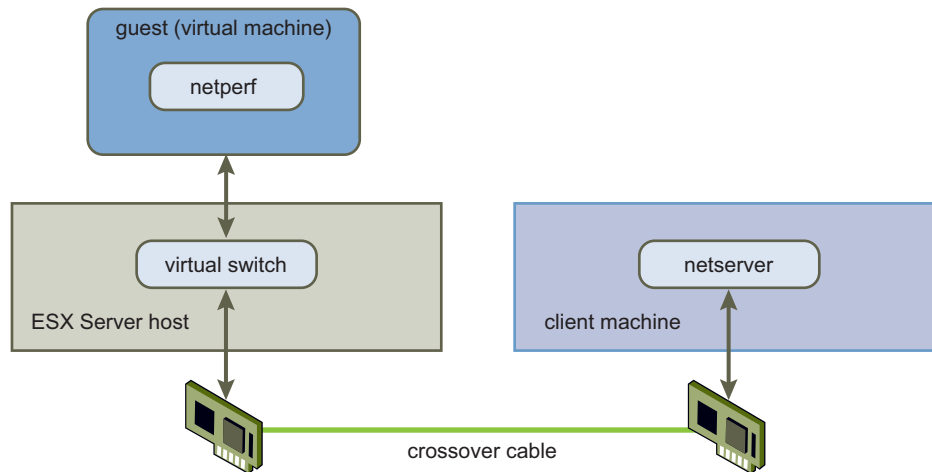
Table 1. Configuration Details

Server Machine	
CPUs	2 dual-core Intel® Xeon® 5160 CPUs
Memory	8GB

Table 1. Configuration Details (Continued)

Server Machine	
Network Cards:	
■ 1GigE Physical NIC	■ Intel® e1000 MT Server Adaptor
■ 10GigE Physical NIC	■ Neterion Xframe II® PCI-X 2.0 10 Gbps Adaptor
Virtualization software	ESX Server 3.5
Virtual Machine	
CPUs	1 virtual CPU
Memory	512MB
Operating system	Microsoft® Windows Server® 2003 SP 1
Client Machine	
CPUs	2 dual-core Intel® Xeon® 5160 CPUs
Memory	4GB
Network Cards:	
■ 1GigE Physical NIC	■ Intel® e1000 MT Server Adaptor
■ 10GigE Physical NIC	■ Neterion Xframe II® PCI-X 2.0 10 Gbps Adaptor
Operating System	RedHat Enterprise Linux 5 (64-bit)

Figure 1 shows the experimental setup for the transmit experiments where the virtual machine sends data. For the receive experiments, the netperf and netserver processes were exchanged.

Figure 1. Experimental Setup

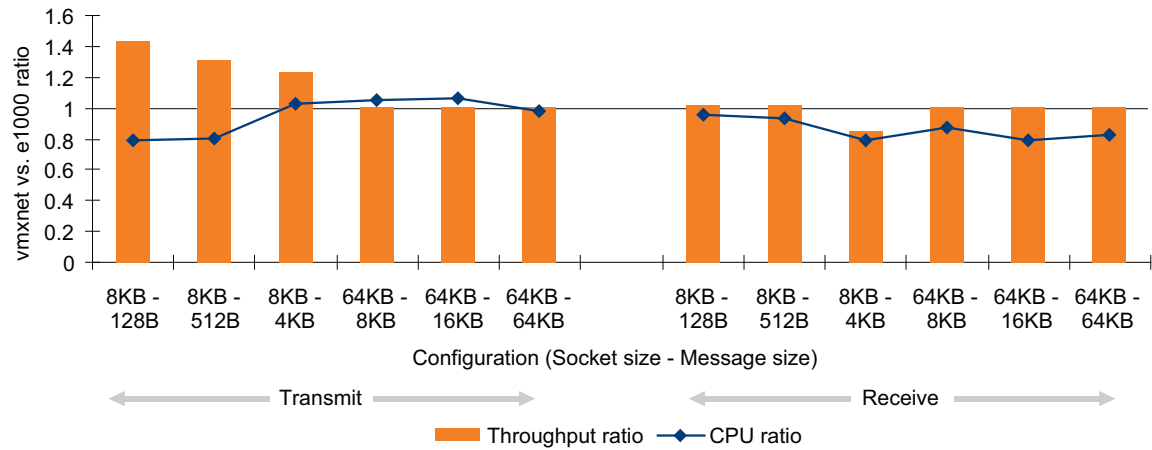
Performance Results

To compare vmxnet performance with e1000 performance, we ran experiments using 32-bit and 64-bit Windows Server 2003 virtual machines, using both 1 Gbps and 10 Gbps NICs.

Figure 2 to Figure 5 illustrate the vmxnet versus e1000 performance ratios for TCP throughput and CPU utilization for different netperf configurations. The points on the left side of the graph are for transmit workload, and the points on the right side are for the receive workload. To ensure a fair comparison, CPU utilization is normalized by throughput. For TCP throughput, a value greater than 1 indicates that vmxnet can drive a higher throughput than e1000 for the given network configuration. For CPU utilization, a value greater than 1 indicates that vmxnet uses more CPU resources than e1000 to drive the same amount of throughput. That is, vmxnet is less efficient than e1000 for the given configuration. The X-axis on the graphs represents the different network configuration parameters (socket size and message size) that are passed to netperf for the virtual machine's transmit and receive paths.

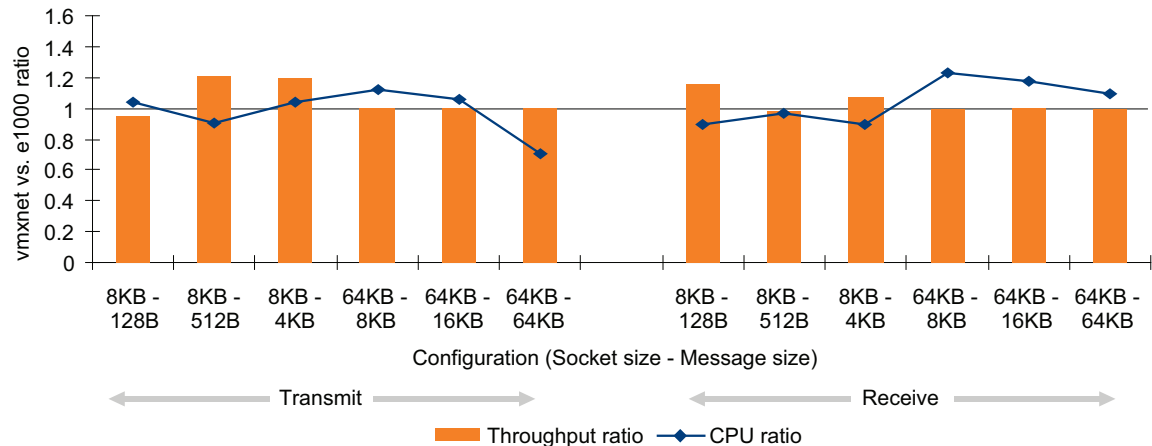
For a 32-bit Windows virtual machine on a 1 Gbps network, vmxnet performed better than e1000 with respect to throughput for small socket sizes as shown in Figure 2. For large socket buffer sizes, vmxnet and e1000 throughputs are similar.

Figure 2. 32-bit Windows 1GigE vmxnet vs. e1000 Performance Ratio



For small socket buffer sizes, TCP throughput is sensitive to latency because the TCP window size is small. The effect of small buffer sizes is partly compensated for by the lower latencies of 10 Gbps networks. Consequently, the performance gain for small socket buffer sizes is smaller for 10 Gbps networks compared to 1 Gbps networks. Figure 3 depicts the results obtained using a 32-bit virtual machine on a 10 Gbps network.

Figure 3. 32-bit Windows 10GigE vmxnet vs. e1000 Performance Ratio

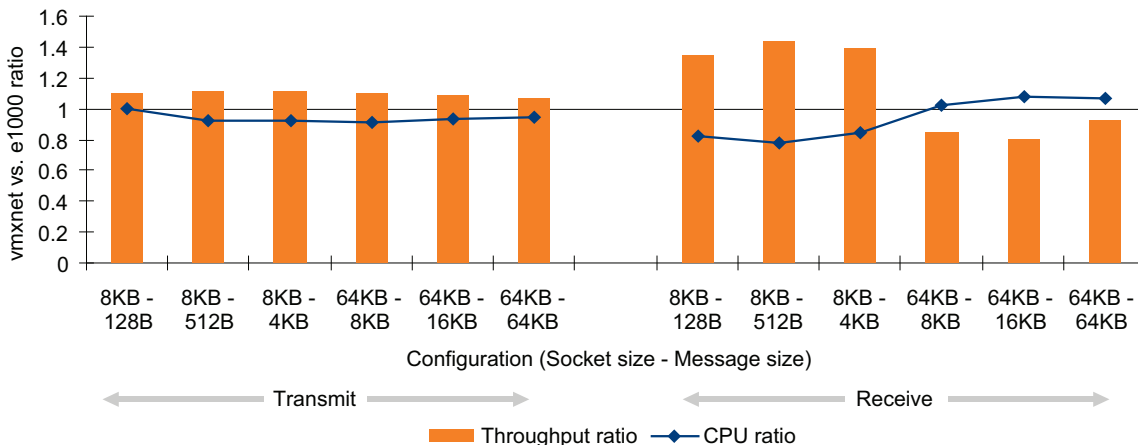


As shown in Figure 2 and Figure 3, on a 32-bit virtual machine, the CPU utilization results are mixed, with vmxnet performing better in some cases and e1000 better in others. However, on 64-bit virtual machines, shown in Figure 4 and Figure 5, vmxnet is the winner.

On Intel machines, 64-bit virtual machines run with hardware virtualization support (Intel-VT), but 32-bit virtual machines run with software virtualization. When virtual machines run with hardware virtualization support, the virtual device access results in expensive virtual machine exits that consume a lot of CPU cycles. By design, the vmxnet driver accesses these registers fewer times than the e1000 driver; hence, vmxnet displays better performance than e1000 in 64-bit virtual machines.

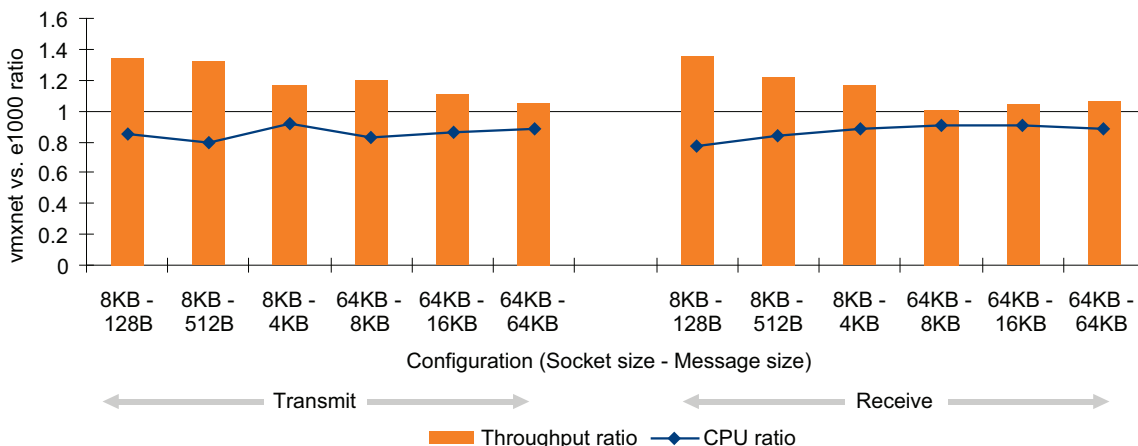
As shown in Figure 4, for a 64-bit virtual machine connected to a 1 Gbps network vmxnet performed better than e1000 for all cases with the exception of one case. For large socket sizes, the receive throughput for vmxnet was lower than that of e1000 even though vmxnet was more CPU efficient than e1000. Significant variance was observed during the throughput measurements for the 8KB socket – 4KB message case on the 1 Gbps setup. This discrepancy could be due to the guest operating system behavior.

Figure 4. 64-bit Windows 1GigE vmxnet vs. e1000 Performance Ratio



In [Figure 5](#), for 64-bit experiments on a 10 Gbps network, vmxnet showed higher throughput for all cases with lower CPU usage than e1000.

Figure 5. 64bit Windows 10GigE vmxnet vs. e1000 Performance Ratio

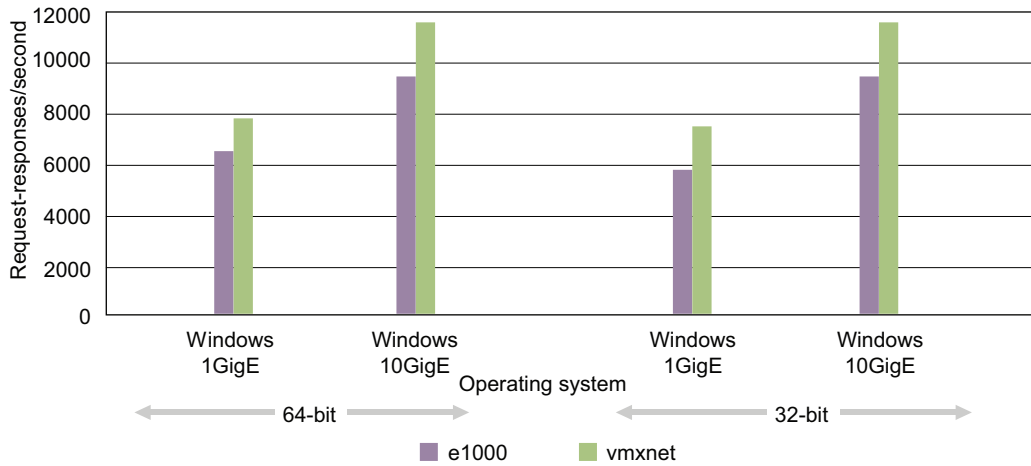


Latency Results

In addition to throughput, end-to-end latency is an important metric to gauge networking performance. Certain applications, such as database applications and interactive desktop workloads, are more latency sensitive compared to other applications. The experiments show that ESX Server does not introduce any significant delays in the network path.

[Figure 6](#) shows the number of request-responses per second for the 64-bit and 32-bit versions of Windows with both 1 Gbps and 10 Gbps physical NICs. A message size of one byte was used for all latency experiments.

In the current design, latency in the e1000 emulation path is higher than in the vmxnet emulation path. This is evident in the results of the netperf TCP_RR tests shown in [Figure 6](#). For all configurations, requests per second are higher for vmxnet than for e1000. Because the requests per second are inversely proportional to latency, latencies are lower with vmxnet than with e1000.

Figure 6. TCP Request-Responses per Second

Conclusion

This study compares performance results for e1000 and vmxnet virtual network devices on 32-bit and 64-bit guest operating systems using the netperf benchmark. The results show that when a virtual machine is running with software virtualization, e1000 is better in some cases and vmxnet is better in others. Vmxnet has lower latency, which sometimes comes at the cost of higher CPU utilization. When hardware virtualization is used, vmxnet clearly provides the best performance.

References

- The netperf Web site at <http://www.netperf.org/netperf/>
- Sugerman Jeremy, Venkitachalan Ganesh, Lim Beng-Hong. Virtualizing I/O devices on VMware Workstation's Hosted Virtual Machine Monitor. Usenix, June 2001. <http://www.usenix.org/publications/library/proceedings/usenix01/sugerman/sugerman.pdf>.
- Adams K. and Agesen O. A Comparison of Software and Hardware Techniques for x86 Virtualization. ASPLOS, October 2006. http://www.vmware.com/pdf/asplos235_adams.pdf
- VMware, Inc. white paper. Virtualization Overview. <http://www.vmware.com/pdf/virtualization.pdf>
- VMware Virtual Networking Concepts (VMTN) <http://www.vmware.com/resources/techresources/997>

VMware, Inc. 3401 Hillview Ave., Palo Alto, CA 94304 www.vmware.com

Copyright © 2008 VMware, Inc. All rights reserved. Protected by one or more of U.S. Patent Nos. 6,397,242, 6,496,847, 6,704,925, 6,711,672, 6,725,289, 6,735,601, 6,785,886, 6,789,156, 6,795,966, 6,880,022, 6,944,699, 6,961,806, 6,961,941, 7,069,413, 7,082,598, 7,089,377, 7,111,086, 7,111,145, 7,117,481, 7,149,843, 7,155,558, 7,222,221, 7,260,815, 7,260,820, 7,269,683, 7,275,136, 7,277,998, 7,277,999, 7,278,030, 7,281,102, and 7,290,253; patents pending. VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Microsoft, Windows and Windows NT are registered trademarks of Microsoft Corporation. Linux is a registered trademark of Linus Torvalds. All other marks and names mentioned herein may be trademarks of their respective companies.
Revision 20080304 Item: PS-049-PRD-01-01