Multi-NIC Networking Performance in ESX 3.0.1 and XenEnterprise 3.2.0

Just a few years ago, generating enough network traffic to push a single 1 Gbps network adapter to capacity would saturate a computer's main processor. However, processors have advanced in these few years to the point that 1 Gbps networking does not stress a modern native or virtualized system.

In order to better expose the real virtualization overhead of high-throughput networking, it is now necessary to increase the load on the system beyond a single 1 Gbps link.

In this paper, we characterize the virtualization overheads of virtual machines by measuring the performance under heavy networking in a uniprocessor virtual machine. This was done by configuring multiple 1 Gbps Ethernet adapters ("NICs") in the server, each associated with a netperf or netserver instance running over a unique subnet and port.

The hypervisors tested were ESX Server 3.0.1 (referred to as "ESX301") and XenEnterprise 3.2.0 (referred to as "XE320"). Both hypervisors were installed with no modifications or tuning. For each hypervisor, the corresponding "tools" packages containing paravirtualized (PV) network drivers was installed in the guest[1]. The hypervisors were installed on a 4-core, 3 GHz HP DL380G5 system with 16GB of memory.

For the virtualized tests, a single virtual machine was configured for each hypervisor with one processor and 1GB of memory, running Windows Server 2003 Release 2 Enterprise Edition (32-bit). During the tests, the virtual machine under test was the only one running in the hypervisor.

The baseline native tests were run on the same 3 GHz HP DL380G5 system, configured with 1 CPU and 1GB memory through `boot.ini` parameters. This system ran the same operating system as was used for the virtualized tests, but used the ACPI multiprocessor HAL instead of the ACPI uniprocessor HAL used in the guests.

The client machine was a 4-core, 2.6 GHz HP DL385G2 system with the same NICs, same memory, and same operating system as the server system, but configured with the full hardware resources of the machine to ensure that the client was not a bottleneck. The server and client systems were connected directly with cross-over Ethernet cables.

For the send measurements, the client ran one or more instances of netserver, one for each configured NIC, while the server ran a corresponding number of netperf processes. For the receive measurements, the server ran one or more instances of netserver, while the client ran a corresponding number of netperf processes. Netperf was configured for TCP stream, 64 KB socket size, and 8 KB message size. This configuration was chosen to get high throughput in the single-NIC case. Other networking configurations may behave differently. Since XenEnterprise 3.2.0 supports a maximum of three virtual NICs in a virtual machine, we do

1. The reader may note that in our earlier paper (http://www.vmware.com/pdf/hypervisor_performance.pdf) we used open-source Xen 3.0.3 without PV drivers. This was because at the time of those tests (Fall 2006) open-source Xen did not have PV drivers available for Windows, and XenSource did not have a product supporting Windows at all.

not present four-NIC numbers here. Full hardware and software configuration details are given at the end of this Tech Note.

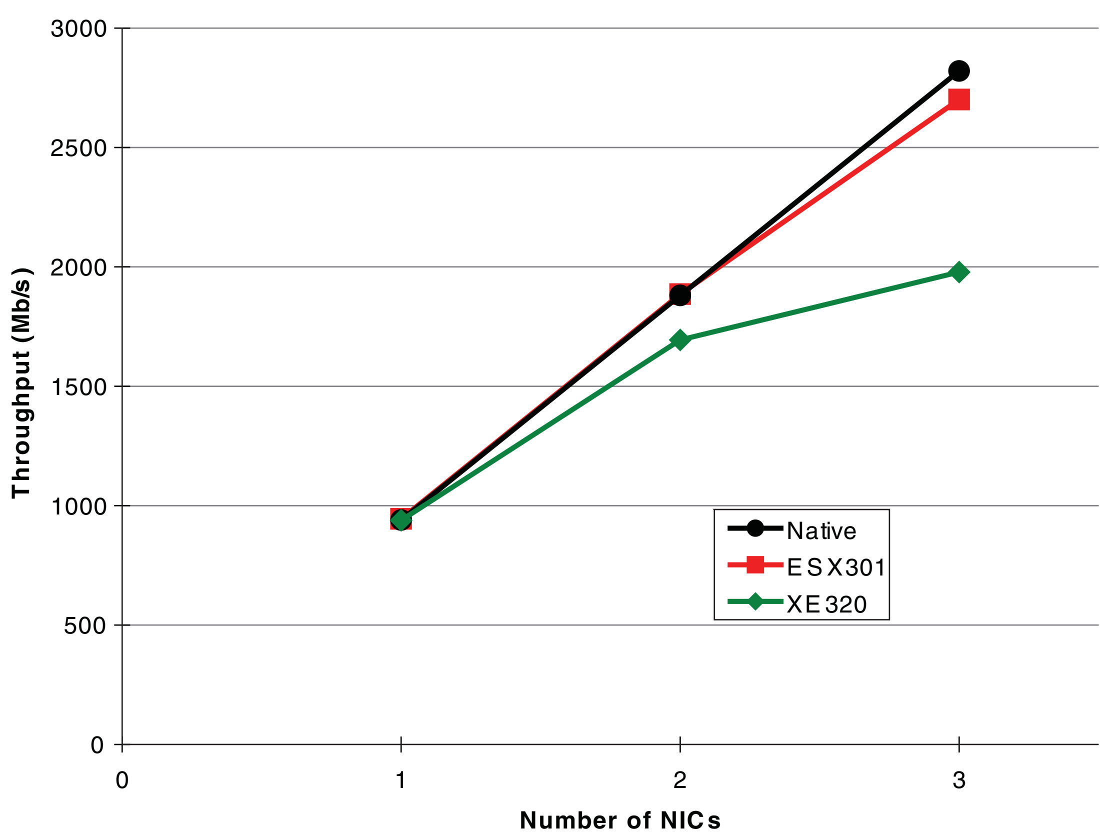
The results for send are shown in Figure 1; the results for receive are shown in Figure 2. In both graphs "Number of NICS" was the number of NICs configured for each test:

- For the virtualized tests, this was the number of virtual NICs configured in each virtual machine. Each virtual NIC was associated with exactly one physical NIC.
- For the native tests, this was the number of physical NICs configured in the server.
- For all tests, this was the number of physical NICs configured in the client.

It can be seen from these graphs that:

- With one NIC configured, the two hypervisors were each within a fraction of one percent of native throughput for both cases. Virtualization overhead had no effect for this lightly-loaded configuration.
- With two NICs, ESX301 had essentially the same throughput as native, but XE320 was slower by 10% (send) and 12% (receive), showing the effect of CPU overhead.
- With three NICs, ESX301 is close to its limit for a uniprocessor virtual machine, with a degradation compared to native of 4% for send and 3% for receive. XE320 is able to achieve some additional throughput using three NICs instead of two, but the performance degradation compared to native is substantial: 30% for send, 34% for receive.

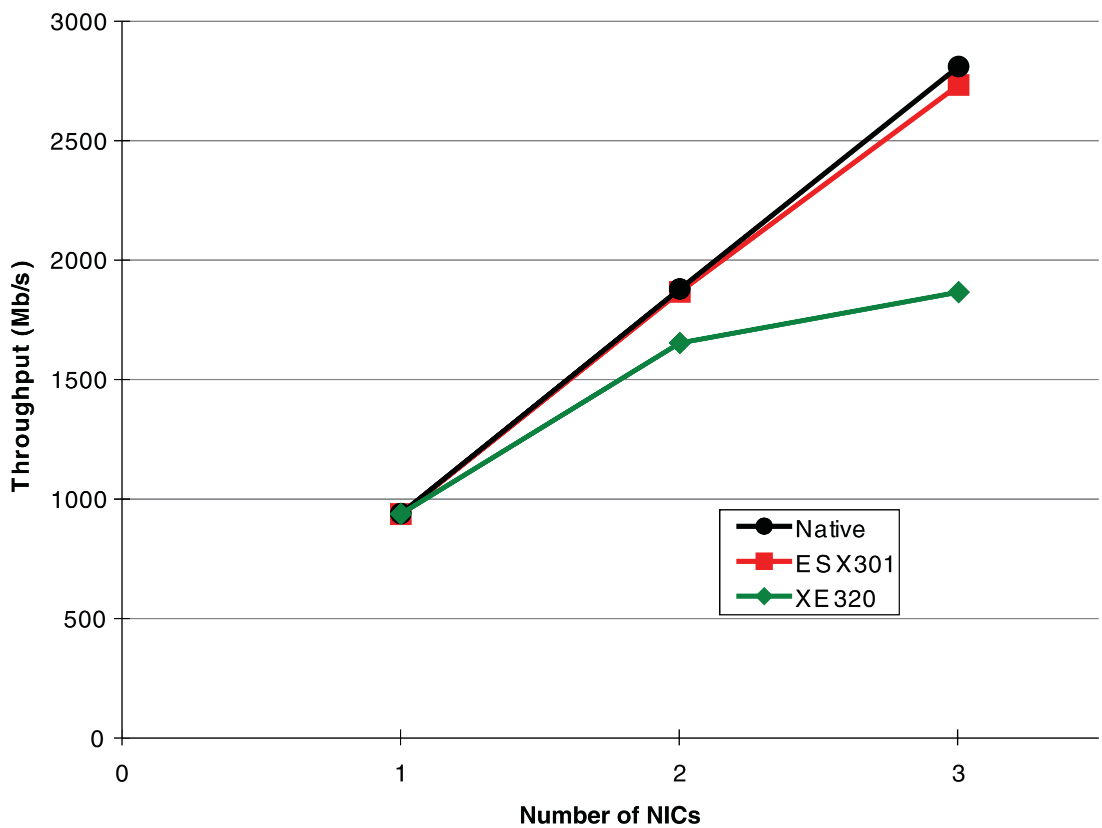
Figure 1. Netperf Send Results



With four NICs, the native system achieved substantial additional throughput, but clearly reached its performance limit. ESX301 can support up to four virtual NICs in a guest. While the fourth NIC is not very useful for achieving additional throughput, it came in handy for connecting to a public network to install applications, move data, etc. All versions of Xen are limited to three NICs in the guest. Communicating with the guest thus required reconnecting one of the virtual NICs to a different physical NIC. In XenEnterprise, the guest needs to be powered down before this kind of network reconfiguration can be performed. In an ESX301 guest, physical NICs can be linked and unlinked to virtual NICs while the guest is running.

Networking will move to a 10 Gbps standard in the near future. At this speed a uniprocessor system will not be able to saturate a single NIC, at least for a while, and at that time the use of a single NIC will no longer be a limiting factor for network performance. In the meantime, performance of virtualized applications is best measured by driving the system close to saturation through the use of sufficient hardware resources.

Figure 2. Netperf Receive Results



Configurations

Server Configuration (Virtualized System)

Hardware

Base hardware: HP DL380G5, 4-core Intel Woodcrest Xeon 5160 3 GHz

Memory: 16GB DDR2 667

Disks: Four 146GB 10,000 RPM Serial SCSI

NICs: Two Intel PRO/1000 PT Dual Port Server Adapters, Intel 82571EB Gigabit Ethernet Controller (rev 06)

Hypervisor Configuration

ESX301: ESX Server 3.0.1 GA release using BT monitor, 32 bit

XE320: XenEnterprise 3.2.0. Based on open-source Xen 3.0.4-1, Intel VT, 32 bit.
Dom0: XenLinux 2.6.16.38, 32 bit, 512 MB

Guest Configuration

Virtual hardware: 1 processor, 1GB memory, 1, 2, or 3 NICs (using paravirtualized drivers)

Operating system: Windows Server 2003 Release 2 Enterprise Edition (32-bit), ACPI uniprocessor HAL

Server Configuration (Native System)

Base hardware: HP DL380G5, 4-core Intel Woodcrest Xeon 5160 3 GHz (limited to one processor through `boot.ini` parameters)

Memory: 16GB DDR2 667 (limited to 1GB through `boot.ini` parameters)

Disks: Four 146GB 10,000 RPM Serial SCSI

NICs: Two Intel PRO/1000 PT Dual Port Server Adapters, Intel 82571EB Gigabit Ethernet Controller (rev 06)

Operating system: Windows Server 2003 Release 2 Enterprise Edition (32-bit), ACPI multiprocessor HAL

Client Configuration

Base hardware: HP DL385G2, 4-core AMD Opteron 2218 Rev. F 2.6 GHz

Operating system: Windows Server 2003 Release 2 Enterprise Edition (32-bit), ACPI multiprocessor HAL

Memory: 16GB DDR2 667

Disks: Two 146GB 10,000 RPM Serial SCSI

NICs: Two Intel Pro/1000 PT Dual Port Server Adapter, Intel 82571EB Gigabit Ethernet Controller (rev 06)

VMware, Inc. 3145 Porter Drive Palo Alto, CA 94304 www.vmware.com

Revision 20070424 Item: ESX-ENG-Q207-384