CASE STUDY
Intel® Xeon® processor 5500 series
Virtualization

Maximizing File Transfer Performance Using 10Gb Ethernet and Virtualization

A FedEx Case Study

FedEx Corporation

FedEx Corporation is a worldwide information and business solution company, with a superior portfolio of transportation and delivery services.

THE CHALLENGE

Spiraling data center network complexity, cable maintenance and troubleshooting costs, and increasing bandwidth requirements led FedEx Corporation to investigate techniques for simplifying the network infrastructure and boosting file transfer throughput.

In response to this challenge, FedEx—in collaboration with Intel—conducted a case study to determine the most effective approach to achieving near-native 10-gigabit file transfer rates in a virtualized environment based on VMware vSphere* ESX* 4 running on servers powered by the Intel® Xeon® processor 5500 series. The servers were equipped with Intel® 10 Gigabit AF DA Dual Port Server Adapters supporting direct-attach copper twinaxial cable connections. For this implementation, Intel® Virtual Machine Device Queues (VMDq) feature was enabled in VMware NetQueue*.

File transfer applications are widely used in production environments, including replicated data sets, databases, backups, and similar operations. As part of this case study, several of these applications are used in the test sequences. This case study:

- Investigates the platform hardware and software limits of file transfer performance
- Identifies the bottlenecks that restrict transfer rates
- Evaluates trade-offs for each of the proposed solutions
- Makes recommendations for increasing file transfer performance in 10 Gigabit Ethernet (10G) native Linux* and a 10G VMware virtualized environment

The latest 10G solutions let users cost-effectively consolidate the many Ethernet and FibreChannel adapters deployed in a typical VMware ESX implementation. VMware ESX, running on Intel Xeon processor 5500 series-based servers, provides a reliable, high-performance solution for handling this workload.

THE PROCESS

In the process of building a new data center, FedEx Corporation, the largest express shipping company in the world, evaluated the potential benefits of 10G, considering these key questions:

- Can 10G make the network less complex and streamline infrastructure deployment?
- Can 10G help solve cable management issues?
- Can 10G meet our increasing bandwidth requirements as we target higher virtual machine (VM) consolidation ratios?

Cost Factors

How does 10G affect costs? Both 10GBASE-T and Direct Attach Twinax cabling (sometimes referred to as 10GSFP+Cu or SFP+ Direct Attach cost less than USD 400 per adapter port. In comparison, a 10G Short Reach (10GBASE-SR) fiber connection costs approximately USD 700 per adapter port.[1]

In existing VMware production environments, FedEx had used eight 1GbE connections implemented with two quad-port cards (plus one or two 100/1000 ports for management) in addition to two 4Gb FibreChannel links. Based on market pricing, moving the eight 1GbE connections on the quad-port cards to two 10GBASE-T or 10G Twinax connections is cost effective today. Using two 10G connections can actually consume less power than the eight 1GbE connections they replace—providing additional savings over the long term. Having less cabling typically reduces instances of wiring errors and lowers maintenance costs, as well. FedEx used IEEE802.1q trunking to separate traffic flows on the 10G links.

Figure 1. Configuration of the server wiring when using eight 1GbE connections.

Figure 2. Configuration of the server wiring when two 10G connections replace eight 1GbE connections.

Engineering Trade-offs

Engineering trade-offs are also a consideration.

Direct Attach/Twinax 10G cabling has a maximum reach of seven meters for passive cables, which affects the physical wiring scheme to be implemented. The servers have to be located within a seven-meter radius of the 10G switches to take advantage of this new technology. However, active cables are available that can extend this range if necessary. A key feature of the Twinax cable technology is that it uses exactly the same form-factor connectors that the industry-standard SFP+ optical modules use. Using this technology allows you to select the most cost-effective and power-efficient passive Twinax for short reaches and then move up to active Twinax, SR fiber, or even Long Reach (LR) fiber for longer runs. Twinax adapters consume approximately 3W per port when using passive cabling.

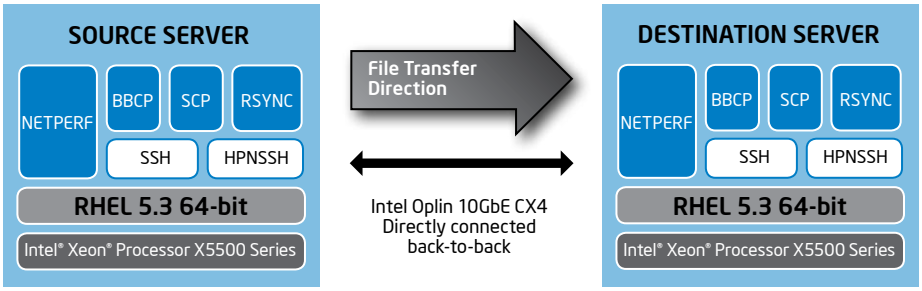
10GBASE-T's maximum reach of 100 meters makes it a flexible, data center-wide deployment option for 10GbE. 10GBASE-T is also backwards-compatible with today's widely deployed Gigabit Ethernet infrastructures. This feature makes it an excellent technology for migrating from GbE to 10GbE, as IT can use existing infrastructures and deployment knowledge. Current-generation 10GBASE-T adapters use approximately 10W per port, and upcoming products will consume even less, making them suitable for integration onto motherboards in future server generations.

Framing the Challenge

File transfer applications are widely used in production environments to move data between systems for various purposes and to replicate data sets across servers and applications that share these data sets. FedEx, for example, uses ftp, scp, and rsync for data replication and distribution in their production networks.

In addition to the considerations associated with cable consolidation, cost factors, and the power advantages of using 10G, another key question remained: Can today's servers effectively take advantage of 10G pipes? Using ftp, FedEx was able to drive 320 Mbps over a 1G connection. Initial 10G testing, however, indicated that they could only achieve 560 Mbps, despite the potential capabilities of 10x faster pipes.

Plugging a 10G NIC into a server does not automatically deliver 10 Gbps of application level throughput. An obvious question arises: What can be done to maximize file transfer performance on modern servers using 10G?



Hardware	Intel® Xeon® processor X5560 series @ 2.8 GHz (8 cores, 16 threads); SMT, NUMA, VT-x, VT-d, EIST, Turbo Enabled (default in BIOS); 24 GB Memory; Intel 10GbE CX4 Server Adapter with VMDq
Test Methodology	RAM disk used, not disk drives. We are focused on network I/O, not disk I/O
What is being transferred?	Directory structure, part of Linux repository: ~8 G total, ~5000 files, variable file size, average file size ~1.6 MB
Data Collection Tools Used	Linux * utility "sar": Capture receive throughput and CPU utilization
Application Tools used	Netperf (common network micro-benchmark); OpenSSH, OpenSSL (standard Linux layers); HPN-SSH (optimized version of OpenSSH); scp, rsync (standard Linux file transfer utilities); bbcp ("BitTorrent-like" file transfer utility)

Figure 3. Native test configuration details.

FedEx and Intel delved deeper into the investigation, using several common file transfer tools on both native Linux* and VMware Linux VMs. The test environment featured Red Hat Enterprise Linux (RHEL) 5.3 and VMware vSphere ESX4 running on Intel Xeon processor 5500 series-based servers. These servers were equipped with Intel 10 Gigabit AF DA Dual Port Server Adapters supporting direct attach copper twinaxial (10GBASE-CX1) cable connections and the VMDq feature supported by VMware NetQueue.

Native Test Configuration

Figure 3 details the components of the native test configuration. The test systems were connected back-to-back over 10G to eliminate variables that could be caused by the network switches themselves. This should be considered a best-case scenario because any switches add some finite amount of latency in real-world scenarios, possibly degrading performance for some workloads.

A RAM disk, rather than physical disk drives, was used in all testing to focus on the network I/O performance rather than being limited by disk I/O performance.

The default bulk encryption used in OpenSSH and HPN-SSH is Advanced Encryption Standard (AES) 128-bit. This was not changed during testing.

The application test tools included the following:

- **netperf:** This commonly used network-oriented, low-level synthetic, micro-benchmark does very little processing beyond forwarding the packets. It is effective for evaluating the capabilities of the network interface itself.
- **OpenSSH, OpenSSL:** These standard Linux layers perform encryption for remote access, file transfers, and so on.
- **HPN-SSH:** This optimized version of OpenSSH was developed by the Pittsburgh Supercomputer Center (PSC). For more details, visit **www.psc.edu/networking/projects/hpn-ssh/**
- **scp:** The standard Linux secure copy utility
- **rsync:** The standard Linux directory synchronization utility
- **bbcp:** A peer-to-peer file copy utility (similar to BitTorrent) developed by the Stanford Linear Accelerator Center (SLAC). For more details, go to **www.slac.stanford.edu/~abh/bbcp/**

Synthetic Benchmarks versus Real File Transfer Workloads for Native Linux*

The following two sections examine the differences between synthetic benchmarking and benchmarks generated during actual workloads while running native Linux.

Synthetic Benchmarks

Figure 4 shows the results of two test cases using netperf. In the first case, the 10G card was plugged into a PCIe* Gen1 x4 slot, which limited the throughput to about 6 Gbps because of the PCIe bus bottleneck. In the second case, the card was plugged into a PCIe Gen1 x8 slot, which allowed full throughput at near line rate.

PCIe slots can present a problem if the physical slot size does not match the actual connection to the chipset. To determine which slots are capable of full PCIe width and performance, check with the system vendor. The proper connection width can also be verified using system tools and log files. PCIe Gen1 x8 (or PCIe Gen2 x4, if supported) is necessary to achieve 10 Gbps throughput for one 10G port. A dual-port 10G card requires twice the PCIe bus bandwidth.

As demonstrated, achieving 10 Gbps transfer rates is quite easy using a synthetic benchmark. The next section looks at a case where actual workloads are involved.

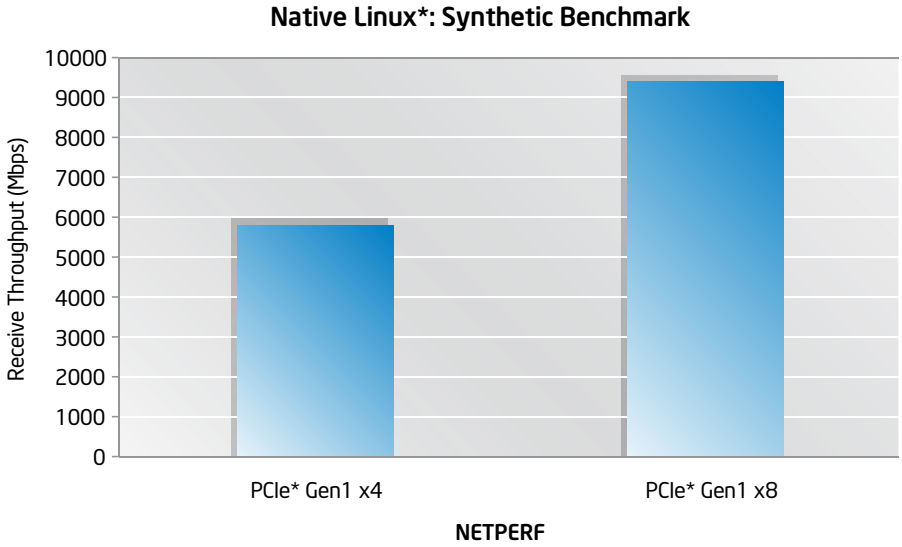


Figure 4. Synthetic benchmark for native Linux*.

Benchmarks Based on Actual Workloads

Real applications present their own unique challenges. Figure 5 shows the earlier netperf results as a reference bar on the left and seven test cases to the right. Tool choice obviously matters, but the standard tools are not very well threaded, so they don't take full advantage of the eight cores, 16 threads, and NIC queues (more than 16) available in this particular hardware platform. The scp tool running over standard ssh, or scp(ssh) in Figure 6, and the rsync(ssh) case both achieve only 400–550 Mbps, about the same as FedEx's initial disappointing results with ftp. Multi-threaded file transfer tools offer a potential performance boost, and two promising candidates emerged during a Web search: HPN-SSH from PSC and bbcp from SLAC.

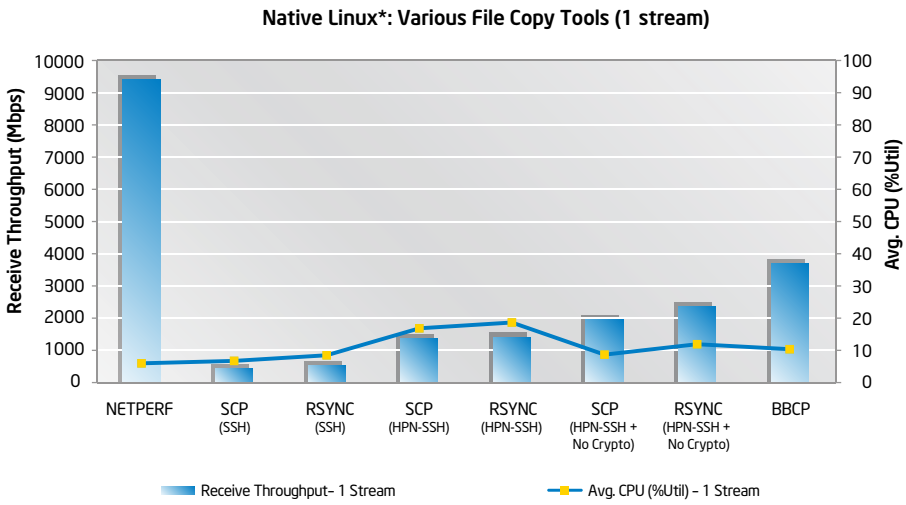


Figure 5. Comparison of various file copy tools (one stream).

Using the HPN-SSH layer to replace the OpenSSH layer drives the throughput up to about 1.4 Gbps. HPN-SSH uses up to four threads for the cryptographic processing and more than doubles application layer throughput. Clearly, performance is moving in the right direction.

If improving the bulk cryptographic processing helps that much, what could the team achieve by disabling it entirely? HPN-SSH offers that option as well. Without bulk cryptography, scp achieves 2 Gbps and rsync achieves 2.3 Gbps. The performance gains in these cases, however, rely on bypassing encryption for file transfers. HPN-SSH provides significant performance advantages over the default OpenSSH layer that is provided in most Linux distributions; this approach warrants further study.

According to a presentation created by SLAC titled "P2P Data Copy Program bbcp" (**http://www.slac.stanford.edu/grp/scs/paper/chep01-7-018.pdf**), bbcp encrypts sensitive passwords and control information but does not encrypt the bulk data transfer. This design trade-off sacrifices privacy for speed. Even without encrypting the bulk data, this can still be an effective trade-off for many environments where data privacy is not a critical concern. With bbcp, using the default four threads, the file transfer rates reached 3.6 Gbps. This represents the best results so far, surpassing even the HPN-SSH cases with no cryptographic processing, and it's more than six times better than the initial test results with scp(ssh). The bbcp approach is very efficient and bears further consideration.

Based on these results, FedEx is actively evaluating the use of HPN-SSH and bbcp in their production environments. None of the techniques tried so far, however, has even come close to achieving 10 Gbps of throughput—not even reaching half of that target. At this point, the Intel and FedEx engineering team focused on identifying any bottlenecks preventing the 10 Gbps file transfer target from being reached. Other than rewriting all of the tools to enhance the performance through multi-threading, the team also wanted to know if any other available techniques could boost file transfer rates?

To gain a better understanding of the performance issues, the engineering team ran eight file transfers in parallel streams to attempt to drive up aggregate file transfer throughput performance and obtain better utilization of the platform hardware resources. Figure 6 indicates the results.

In Figure 6, the first four red bars show that using eight parallel streams overcomes the threading limits of these tools and drives aggregate bulk encrypted throughput much higher:

- 2.7 Gbps with scp(ssh)
- 3.3 Gbps with rsync(ssh)
- 4.4 Gbps with scp(HPN-SSH)
- 4.2 Gbps with rsync(HPN-SSH)

These results demonstrate that using more parallelism dramatically scales up performance by more than five times, but the testing did not demonstrate eight times the throughput when using eight threads in parallel. The resolution to the problem does not lie in simply using the brute-force approach and running more streams in parallel.

These results also show that bulk encryption is expensive, in terms of both throughput and CPU utilization. HPN-SSH, with its multi-threaded cryptography, still provides a significant benefit, but not as dramatic a benefit as in the single-stream case.

The results associated with the remaining three red bars of Figure 6 are instructive. The first two cases use HPN-SSH with no bulk cryptography, and the third case is the eight-thread bbcp result in which bulk data transfers are not encrypted. These results demonstrate that it is possible to achieve nearly the same 10 Gbps line rate throughput number as the netperf micro-benchmark result when running real file transfer applications.

As this testing indicates, using multiple parallel streams and disabling bulk cryptographic processing is effective for obtaining near 10-Gbps line rate file

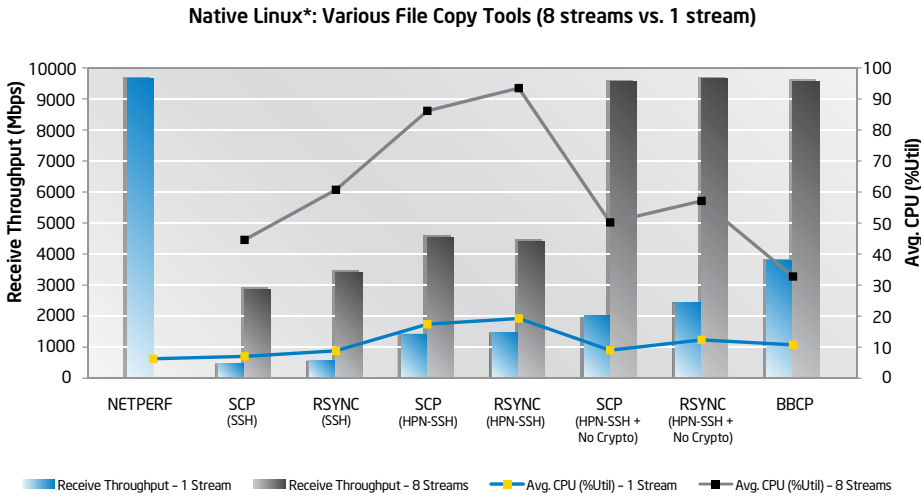


Figure 6. Comparison of various file copy tools (eight streams).

transfer throughput in Linux. These trade-offs may or may not be applicable for a given network environment, but they do indicate an effective technique for gaining better performance with fewer trade-offs in the future. Intel is continuing to work with the Linux community to find solutions to increase file transfer performance.

The next-generation Intel Xeon 5600 processor family, code-named Westmere, and other future Intel Xeon processors will have a new instruction set enhancement to improve performance for Advanced Encryption Standard (AES) bulk encryption called AES-NI. This advance promises to deliver faster file transfer rates when bulk encryption is turned on with AES-NI-enabled platforms.

Best Practices for Native Linux

Follow these practices to achieve the best performance results when performing file transfer operations under native Linux:

- **Configuration:** PCIe Gen1 x8 **minimum** for 1x 10G port; PCIe Gen2 x8 **minimum** for 2x 10G ports on one card
- **BIOS settings:** Turn ON Energy Efficient mode (Turbo, EIST), SMT, NUMA
- Turn ON Receive (Rx) Multi-Queue (MQ) support (enabled by default in RHEL); Transmit (Tx) is currently limited to one queue in RHEL, SLES 11RC supports MQ Tx
- Factor in these limitations of Linux file transfer tools and SSH/SSL layers:
  - –scp and ssh: single threaded
  - –rsync: dual threaded
  - –HPN-SSH: four cryptography threads and single-threaded MAC layer
  - –bbcp: encrypted setup handshake, but not bulk transfer; defaults to four threads
- Use multiple parallel streams to overcome tool thread limits and maximize throughput.
- Bulk cryptography operations limit performance. Disable cryptography for those environments where it is acceptable. The Intel Xeon 5600 processor family (code-named Westmere) and future Intel Xeon processors will improve bulk cryptographic performance using AES-NI.

Achieving Native Performance in a Virtualized Environment

Earlier sections illustrated effective techniques for maximizing file transfer performance using various tools in the Linux native environment and described some of the factors that limit file transfer performance.

The following sections examine performance in a virtualized environment to determine the level of network throughput that can be achieved for both synthetic benchmarks and for various file transfer tools.

In this virtualized environment, the test systems are provisioned with VMware vSphere ESX4 running on Intel Xeon processor 5500 series-based servers. The test systems are connected back-to-back over 10G with VMDq enabled in VMware NetQueue. The VMs on these servers were provisioned with RHEL 5.3 (64 bit) and, as in native cases, the test team used the same application tools and test methodology except in one instance: The team used the esxtop utility for measuring the servers' throughput and CPU utilization.

Within the virtualized environment, these test scenarios were used:

- One virtual machine with eight vCPU and 12 GB RAM
- One virtual machine (eight vCPU and 12 GB RAM) with VMDirectPath I/O
- Eight virtual machines, with each virtual machine having one vCPU and 2 GB RAM
- Eight virtual machines, both with and without the VMDq feature

CASE 1: One Virtual Machine with Eight vCPUs and 12 GB RAM

Each server had one VM configured with eight vCPUs and 12 GB of RAM, using RHEL 5.3 (64 bit) as the guest operating system (OS). The test team ran netperf as a micro-benchmark and also various file transfer tools for transferring a directory structure of approximately 8 GB from the VM on the first server to the VM on the second server. Figure 7 shows the test configuration.

Similar to the testing done in the native Linux case, the test team compared the data from netperf, one stream of file transfer, and eight parallel file transfers. Figure 8 shows the receive network throughput and total average CPU utilization for the micro-benchmark, such as netperf, and various file transfer tools when running a single stream of copy. The figure compares the results for the native data with the data results in the virtualized environment.

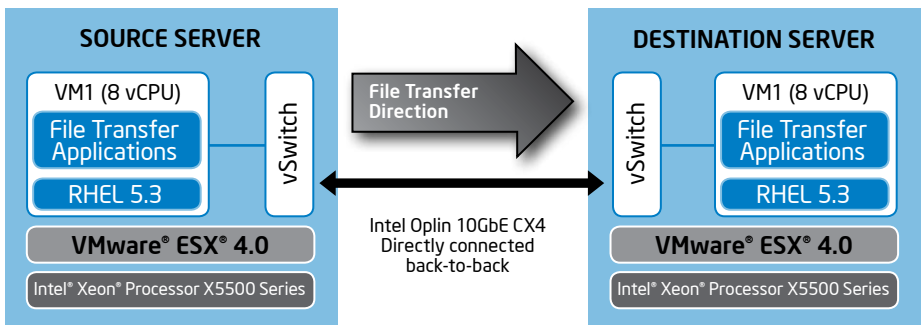


Figure 7. Test configuration for Case 1.

Figure 8 indicates that the throughput in a VM is lower across all cases when compared with the native case. In the virtualized case, the throughput for the netperf test results is 5.8 Gbps compared to 9.3 Gbps in the native case. Even in the case of file transfer tools, such as scp and rsync running over standard ssh, the throughput ranges from 300 Mbps to 500 Mbps, which is slightly lower when compared with the native case.

Using the HPN-SSH layer to replace the OpenSSH layer increases the throughput. Also, disabling cryptography increases the throughput, but not at as high a level as the line rate. The shape of the curve, however, is similar.

The same limitations that occurred in the native case (such as standard tools not being well threaded and cryptography adding to the overhead) also apply in this case. Because of this, the file transfer tools cannot take full advantage of multiple cores and the NIC queues.

Most of the tools and utilities—including ssh and scp—are single threaded; the rsync utility is dual threaded. Using the HPN-SSH layer to replace the OpenSSH layer helps increase the throughput. In HPN-SSH, the cryptography operations are multi-threaded (four threads), which boosts performance significantly. The single-threaded MAC layer, however, still creates a bottleneck. When HPN-SSH is run with cryptography disabled, the performance increases, but the benefits of encrypted data transfer are lost. This is similar to the case with bbcp, which is multi-threaded (using four threads by default), but the bulk transfer is not encrypted.

The next test uses eight parallel streams, attempting to work around the threading limitations of various tools. Figure 10 shows the receive network throughput and CPU utilization for various file transfer tools when running eight parallel streams of copies. This chart also includes comparisons with the native data results.

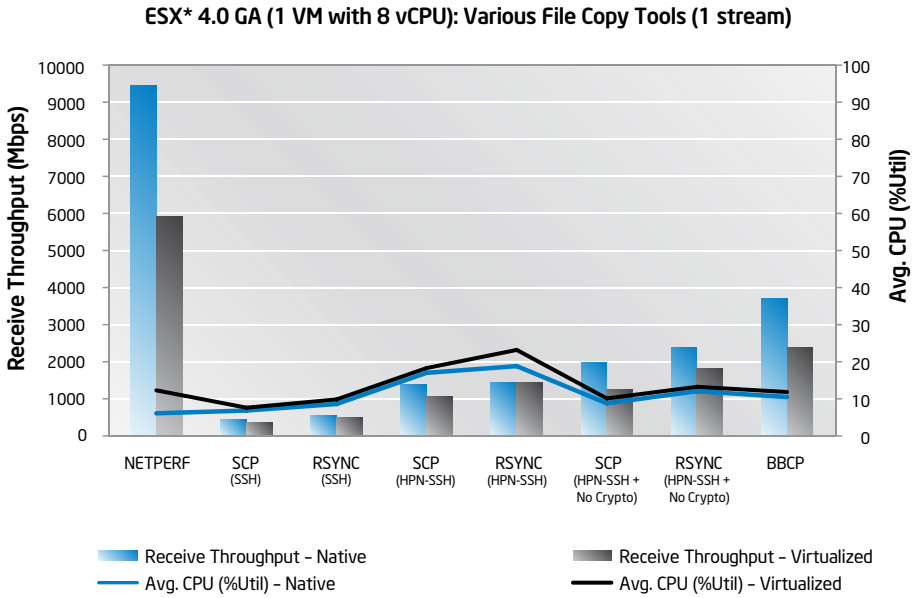


Figure 8. Throughput comparison of various file copy tools (one stream).

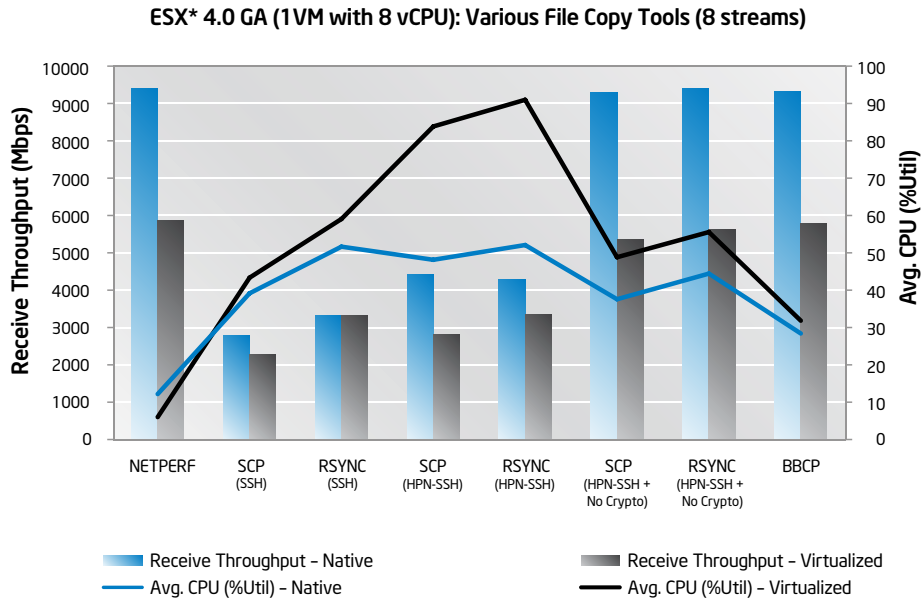


Figure 9. Throughput comparison of various file copy tools (eight streams)

From Figure 9 it's clear that using eight parallel streams overcomes the tools' threading limitations. Figure 9 also shows that cryptography operations are a limiter in the first four cases of the file transfer tools. Better throughput is indicated in the last three cases, in which the copies were made without using cryptography.

Even though these results are better compared to relying on one-stream data, they still don't come close to the line rate (approximately 10 Gbps) achieved in the native case. Since the testing used one VM with eight vCPUs, the test team determined that this might be a good case for using direct assignment of the 10G NIC to the virtual machine.

CASE 2: One Virtual Machine with Eight vCPUs and VMDirectPath

The test bed setup and configuration in this case was similar to that of the previous case except that the 10G was direct assigned to the VM. Figure 10 shows the test configuration for this case.

The test team started with a synthetic benchmark netperf and then ran eight parallel streams of copies for various file transfer tools. The results shown in Figure 12 compare the performance numbers from the VM with DirectPath I/O to the performance numbers of the VM with no DirectPath I/O and native.

As Figure 11 illustrates, VMDirectPath (VT-d direct assignment) of the 10G NIC to the VM increases performance to a level that is close to the native performance results. Nonetheless, the trade-offs associated with using VMDirectPath are substantial.

A number of features are not available for VM configured with VMDirectPath, including VMotion*, suspend/resume, record/replay, fault tolerance, high availability, DRS, and so on. Because of these limitations, the use of VMDirectPath will continue to be a niche solution awaiting future developments that minimize them. VMDirectPath may be practical to use today for virtual security-appliance VMs since these VMs typically do not migrate from host to host. VMDirectPath may also be useful for other applications that have their own clustering technology and don't rely on the VMware platform features for fault tolerance.

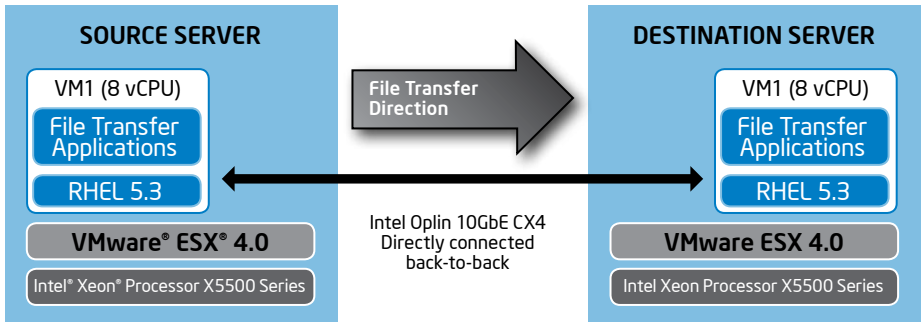


Figure 10. Test configuration for Case 2.

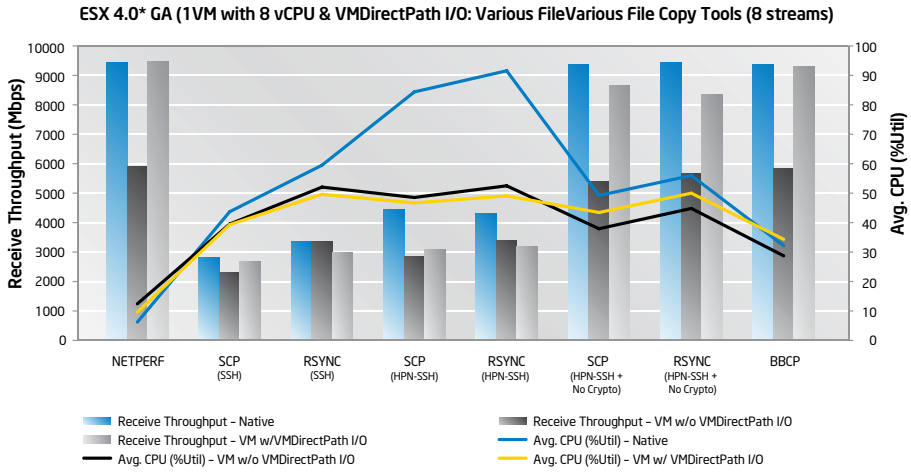


Figure 11. Throughput comparison of various file copy tools (eight streams) using VMDirectPath.

CASE 3: Eight Virtual Machines, Each with One vCPU and 2 GB RAM

The two previous cases incorporated a single large VM with eight vCPUs, which is something similar to a native server. In Case 3, the scenario includes eight virtual machines with each VM configured with one vCPU and 2 GB RAM. The guest OS is still RHEL 5.3 (64 bit). Both of the servers include eight virtual machines and the tests run one stream of copy per VM (so in effect eight parallel streams of copies are running). Figure 12 shows the configuration for this case.

Figure 13 indicates the performance results when running a synthetic micro-benchmark and real file transfer applications on eight VMs in parallel. Comparisons with the results for the native server are shown in blue.

As shown in Figure 13, the aggregate throughput with netperf across eight VMs reaches the same level of throughput as achieved in the native case. With the file transfer tools, cryptography still imposes performance limitations, but the multi-threaded cryptography in HPN-SSH improves performance when compared to the standard utilities, such as ssh. Larger benefits are gained when bulk cryptography is disabled, as indicated by the results shown by the last three red bars. Several of these results show that the virtualized case can equal the performance of the native case.

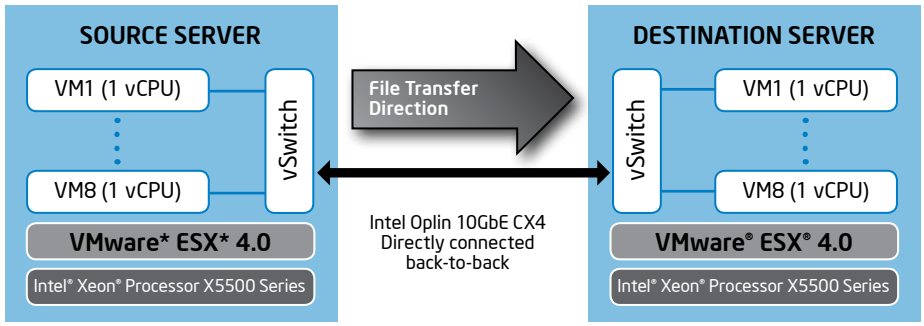


Figure 12. Test configuration for Case 3.

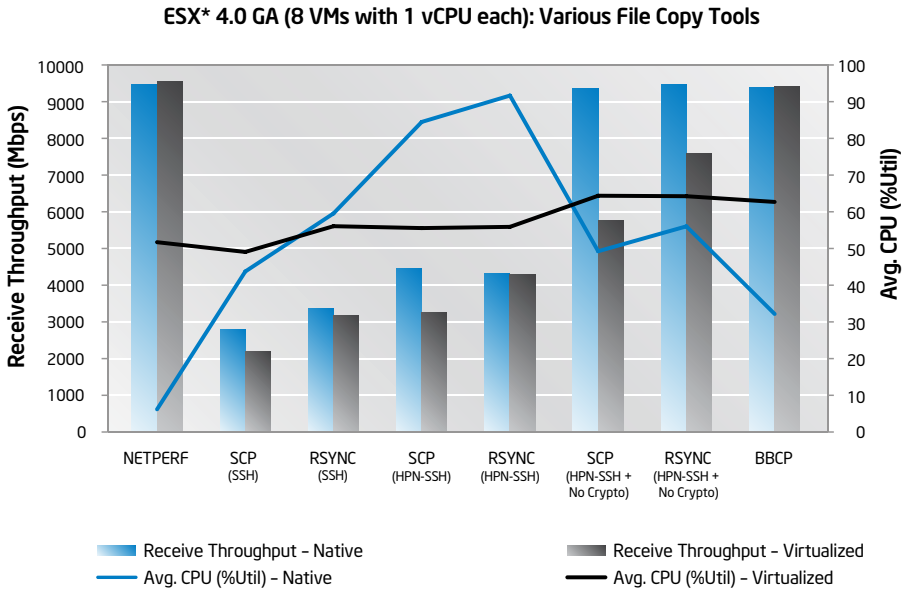


Figure 13. Throughput comparison of various file copy tools with 8 VMs and 1 stream per VM.

Sub Case: Eight Virtual Machines, with and without VMDq

Because the test configuration includes multiple VMs, the VMDq feature offers some advantages. This feature helps offload network I/O data processing from the virtual machine monitor (VMM) software to network silicon. The test team ran a sub case, comparing the receive throughput from various file transfer tools by enabling and disabling the VMDq and associated NetQueue feature.

The results of the first four copy operations in Figure 14 show that there is no difference in throughput, regardless of whether VMDq is enabled or disabled. This is primarily because of the limitations imposed by bulk cryptography operations. When running the file transfer applications tools by disabling the cryptography—as in last three cases—the advantages of the VMDq feature become clear. The results of the last three operations show the advantage of VMDq with the cryptography bottleneck removed. These data results indicate that VMDq improves performance with multiple VMs if there are no system bottlenecks in place (such as cryptography or slot width).

Based on the full range of test results, the test team developed a set of best practices to improve performance in virtualized environments, as detailed in the next section.

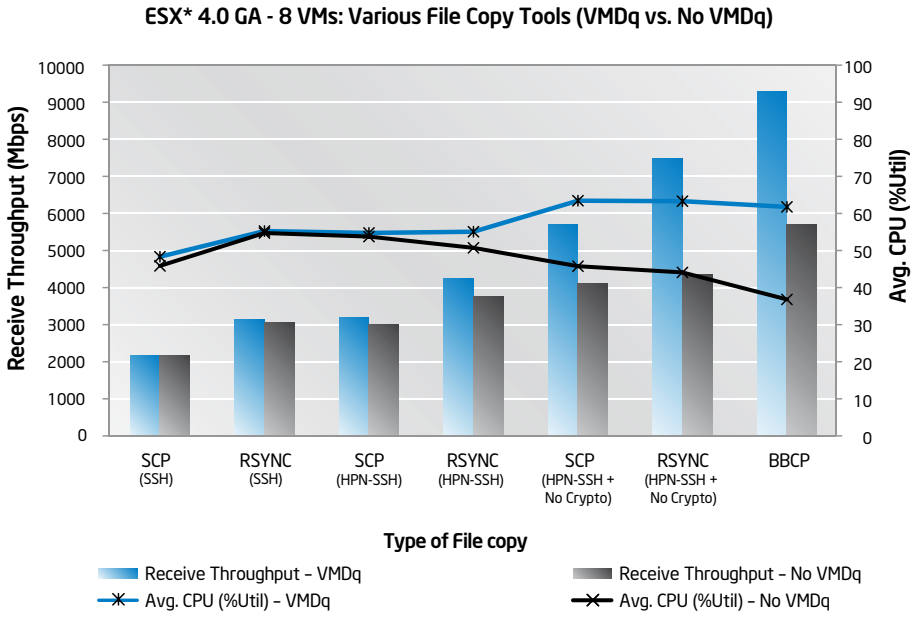


Figure 14. Throughput comparison of various file copy tools (eight streams) with and without Virtual Machine Device Queues (VMDq).

Best Practices for Virtualized Environments (ESX* 4.0)

Consider the following guidelines for achieving the best data transfer performance in a virtualized environment running VMware VSphere ESX 4.

- Confirm the actual width of PCIe slots.
  - –Physical slot size doesn't always match the actual connection to the chipset; check with your server vendor to determine which ones are really full width.
  - –Verify the proper connection using system tools.
  - –PCIe Gen1 x8 (or PCIe Gen2 x4, if supported) is required to achieve 10 Gbps throughput for one 10G port. PCIe Gen2 x8 is necessary for 2x 10G ports.
  - –The transfer rate limitation with PCIe Gen1 x4 is approximately 6 Gbps.
- By default, NetQueue/VMDq is enabled for both Rx and Tx queues.
  - –One queue is allocated per core/thread up to the hardware limits.
  - –Performance is improved with multiple VMs, if there are no system bottlenecks, such as slot width or cryptography.
- Use vmxnet3 (new in ESX 4.0) or vmxnet2 (ESX 3.5) instead of the default e1000 driver.
- Use multiple VMs to improve throughput rather than one large VM.
  - –Two vCPU VMs show better throughput than one vCPU VM in certain cases.
- Tool limitations, slot limitations, and cryptography limitations still apply, as in the native case.

The Results

This case study and the results obtained through the collaborative efforts of FedEx and Intel engineers suggest a number of ways in which file transfer performance in the data center can be maximized, depending on the tools and virtualization techniques used, as well as the hardware configuration.

The most important performance-related considerations, based on the observations and data results obtained during testing, are as follows:

- Be aware of potential PCIe slot hardware issues. Ensure that the PCIe bus width and speed provide sufficient I/O bandwidth for full performance.
- Consider Twinax cabling to greatly reduce the cost of 10G Ethernet. The SFP+ form factor provides a single physical interface standard that can cover the application range—from the lowest cost, lowest power, and shortest reach networking available to the longest reach situations that may be encountered. Choosing the cable type on a port-by-port basis can provide a cost-effective and flexible approach for environmental needs. When using this new cabling approach, 10G can be cost effective today, as compared to commonly deployed usage models with six, eight, or more 1G ports per server.
- Set the appropriate system configuration parameters. The test team evaluated each of the recommended settings detailed in the body of this case study and determined that these settings either improved performance or were neutral for the test workloads. Make sure the BIOS is set correctly and that the OS, VMM, and applications take full advantage of modern platform features.
- Synthetic benchmarks are not the same as real workloads. Although synthetic benchmarks can be useful tools to evaluate subsystem performance, they can be misleading for estimating application level throughput.
- Choose the right tools and usage models. Identify tool-threading limits, and use more parallelism when possible. Cryptography can be a bottleneck, so disable it if it is not required. If it is required, choose an implementation that has optimal performance.
- Set your expectations correctly. Configuration using multiple VMs will often outperform a single VM. Can your application scale in this fashion? In some cases multi-vCPU VM is better if the applications are well threaded. Moving directly from your physical server configurations and tunings to the nearest equivalent VM is an effective starting point, but ultimately may not be the best trade-off. Virtualization features, such as VMDq and NetQueue, will perform better when multiple VMs are being used and there are no system bottlenecks, such as PCIe bandwidth limitations or cryptography processing.

This case study demonstrates that it is possible to achieve close to 10G line rate throughput from today's servers powered by Intel Xeon processors running RHEL 5.3 and VMware ESX 4.0. Use the testing methods described to validate the 10G network you are building and to help identify and resolve problems. The results detailed in this case study make it clear that tool choices, usage models, and configuration settings are more important than whether the application is running in a virtualized environment.

For More Information

For answers to questions about server performance, data center architecture, or application optimization, check out the resources on The Server Room site: http://communities.intel.com/community/openportit/server. Intel experts are available to help you improve performance within your network infrastructure and achieve your data center goals.

AUTHORS

Chris Greer, chief engineer, FedEx Services

Chris Greer is a chief engineer of technical architecture at FedEx Services. He has worked for 12 years at FedEx as a network and systems engineer. Currently, he works to research and define network and server standards with an emphasis on server virtualization.

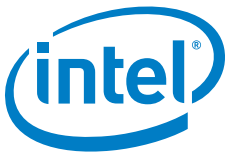
Bob Albers, I/O usage architect, Intel Corporation

Bob Albers is an I/O usage architect in Intel's Digital Enterprise Group's End-User Platform Integration team. He has over 32 years of experience in computer and network design and architecture, including 25 years at Intel. For the last several years Bob has been focused on understanding the usage models and improving the performance of network and security workloads running on native and virtualized Intel servers.

Sreeram Sammeta, senior systems engineer, Intel Corporation

Sreeram Sammeta is a senior systems engineer in Intel's Digital Enterprise Group's End-user Platform Integration group. He previously has worked in various engineering positions in Information Technology during the last 7 years. He holds an M.S in electrical engineering from SIU. His primary focus area is building end-to-end proof-of-concepts; exploring, deploying, and validating emerging end-user usage models. His recent interests include evaluating next generation data center virtualization technologies focusing on performance of virtualized network and security infrastructure appliances.

SOLUTION PROVIDED BY:



[1] Based on list prices for current Intel® Ethernet products.

Printed in USA 1010/BY/MESH/PDF ♻ Please Recycle 323329-002US