WHITE PAPER
Intel® Ethernet 10 Gigabit Server Adapters
Networking for Virtualization

Optimizing Quality of Service for VMware vSphere* 4 Networking with Intel® Ethernet 10 Gigabit Server Adapters

VMware vSphere* 4.1 provides new features, capabilities, and performance increases specifically for 10 Gigabit Ethernet network uplinks.

The industry transition to 10 Gigabit Ethernet (10GbE) has enabled network unification and consolidation, as well as robust support for virtualized usage models. At the same time, however, combining traffic that has been traditionally separated on multiple physical networks and server adapters creates new considerations in terms of guaranteeing quality of service (QoS). This paper continues a series that began with the paper, "Simplify VMware vSphere* 4 Networking with Intel® Ethernet 10 Gigabit Server Adapters,"[1] to examine best practices for analyzing, monitoring, and controlling bandwidth in VMware vSphere 4.1 environments when using 10GbE.

The first paper in the series provides guidance and best practices for moving from Gigabit Ethernet (GbE) to 10GbE networking in the virtualized data center. This paper builds on that discussion with a focus on QoS to provide network architects and decision makers with an overview of the changes in networking capabilities introduced in vSphere 4.1, specifically with regard to the synergies between those new features and Intel Ethernet 10 Gigabit Server Adapters.

Overview

The features and performance upgrades that the new version of vSphere offers build on the following best practices discussed in the prior paper:

- **Best Practice 1: Use virtual distributed switches to maximum effect.** Since many new features require the use of virtual distributed switches, this best practice is critical going forward.
- **Best Practice 2: Streamline configuration using port groups.** This best practice is essentially unchanged with vSphere 4.1, but the present paper has additional details on the traffic shaping feature found in both editions.
- **Best Practice 3: Use VLANs with VLAN Trunking.** This important best practice remains essentially unchanged with vSphere 4.1, relative to vSphere 4.0.
- **Best Practice 4: Use dynamic logical segmentation across two 10GbE ports.** This best practice is expanded in this paper to better support decision making around server network connections.
- **Best Practice 5: Proactively move virtual machines (VMs) away from network hardware failures with VMware vMotion.*** Significant changes to vMotion in vSphere 4.1 require this best practice to be updated.

In particular, the teams at Intel and VMware have extensively tested and discussed the best practices around dynamic logical segmentation across two 10GbE ports in the six months since the first white paper was published. Those discussions and continuing product developments to both switches and network controllers have led to a paradigm shift in the data center, as described in the technology brief, "Virtual Switches Demand Rethinking Connectivity for Servers[2]."

Table of Contents

Gauging Bandwidth and Segmentation Requirements

Using 10GbE connections when deploying virtualization can make data centers more cost effective and easier to manage. Ethernet bandwidth and connectivity requirements should be established with due regard to which vSphere features will be used. That approach allows use cases to be developed to create appropriate network designs. The key is to fully understand the actual bandwidth requirements, based on bandwidth analysis and traffic characterizations, before implementing any designs. Consider the base recommended network model for connecting ESXi* hosts:

- A vNetwork Distributed Switch (vDS) for VM Ethernet connectivity
- Two 10GbE uplinks
- Port groups and VLANs to separate traffic types for performance, isolation, and security

While this configuration is covered in the white paper, "Simplify VMware vSphere* 4 Networking with Intel® Ethernet 10 Gigabit Server Adapters,[1] new features and enhancements in the VMware vSphere* 4.1 release make it worthwhile to revisit existing and future network designs. Additional discussions can be found in Intel blogs on the subject.[3]

This dual 10GbE uplink configuration replaces the previous multiple GbE configuration that was used prior to 10GbE becoming mainstream. While it may seem intuitive to try to divide a 10GbE connection into multiple network connections to mimic the physical separation of a GbE architecture, doing so adds complexity and additional management layers. Moreover, it also significantly reduces the bandwidth and simplification benefits that the move to 10GbE provides. In such cases, new practices specifically created for use with 10GbE are strategically vital. The first step in determining bandwidth requirements is to identify what type of traffic will be deployed on the network and how:

- **Identify the vSphere features to be used.** Many capabilities such as VMware Fault Tolerance (VMware FT) and vMotion can use large amounts of bandwidth. These kernel-based features can actually require more peak bandwidth capabilities than the VMs on the host.
- **Classify the types of applications the VMs will be running on the host.** Some VMs are memory intensive and CPU intensive with little I/O, while others require only low memory but high I/O and CPU. Understanding the specific characteristics and requirements of the relevant VMs is critical to identifying where bottlenecks may reside.
- **Consider the average number of VMs per host.** This characteristic also has direct bearing on expected average and peak Ethernet bandwidth requirements. The optimal number is becoming more and more dynamic as vMotion and Dynamic Resource Scheduling become more prevalent in data center deployments, so a balance of peak and idle requirements must be considered.
- **Identify usage of IP-based storage such as iSCSI or NAS.** Associated usage models require moving large amounts of data around the network, which has a direct impact on bandwidth requirements. Based on those requirements, network architects must decide whether IP-based storage will be unified with data traffic or remain on its own network.

Security requirements may vary between different services and other aspects of a data center. In most cases, VLANs provide adequate separation between traffic types, although physical separation may be desirable in some cases. The number of 10GbE uplinks needed may therefore be based in part on physical security requirements, rather than bandwidth requirements. Implementers are encouraged to refer to additional security and hardening documentation

from VMware.[4] VMware can also provide guidance with regard to the adoption of new methods and architectures when implementing virtualization.

Once these considerations are well understood, the next step is to determine their impact on QoS requirements. Bandwidth control may or may not be needed to ensure the proper allocation of network resources to support QoS and to ensure that all 10GbE ports are performing at optimal levels. More specifically, network engineers must determine what areas of the network require bandwidth control to meet these requirements. The remainder of this paper addresses the process of identifying those network areas in terms of three types of best practices: analysis, monitoring, and control.

Best Practices for QoS Analysis

To ensure optimal availability of throughput on 10GbE uplinks, the proper performance-enhancing features must be enabled and used. For example, using a dual-port 10GbE server adapter on a PCI Express* (PCIe*) Gen 2 x8 connection and enabling VMware NetQueue* is vital in order to get 10 gigabits per second (Gbps) of throughput. Without NetQueue enabled, the hypervisor's virtual switch is restricted to the use of a single processor core, and its processing limitations constrain receive-side (Rx) throughput, in most cases, to 4–6 Gbps.

Relative to GbE ports, this bottleneck assumes even greater importance after migrating to 10GbE. Intel has worked with VMware to deliver support for Intel® Virtual Machine Device Queues (Intel® VMDq),[5] which provides multiple network queues and a hardware-based sorter/classifier built into the network Intel® Ethernet Controller. In combination with NetQueue, VMDq spreads the network processing over multiple queues and CPU cores, allowing for near-native 9.5 Gbps throughput.[6]

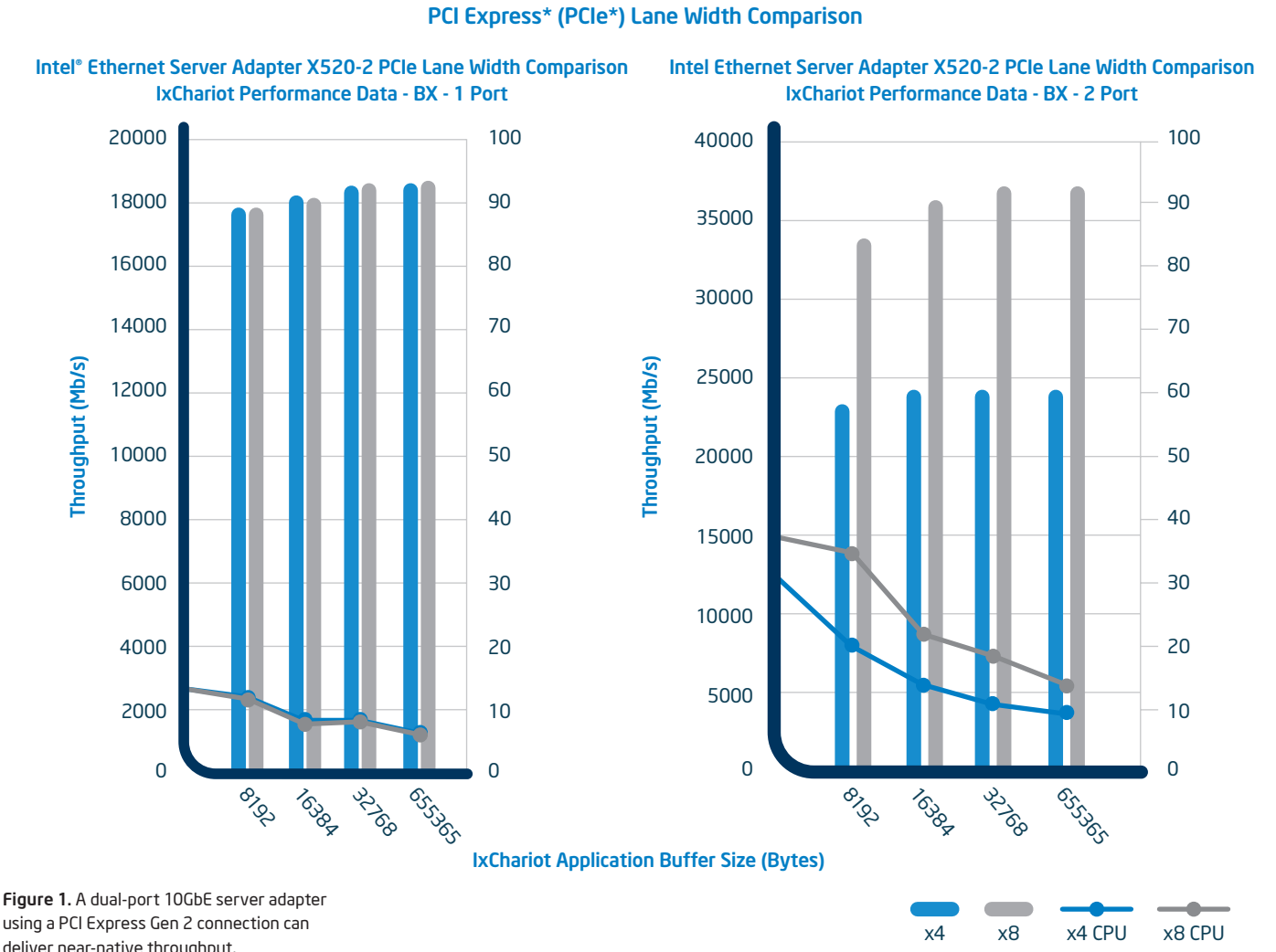

Figure 1. A dual-port 10GbE server adapter using a PCI Express Gen 2 connection can deliver near-native throughput.

QoS Best Practice 1: Use Dual-Port 10GbE Server Adapters and Verify Adequate PCI Express* Connectivity

Using dual-port 10GbE server adapters helps make optimal use of PCIe motherboard slots, allowing for future expansion. Therefore, when available PCIe connectivity on the motherboard permits it, dual-port server adapters are preferable to single-port ones.

To provide near-native 10 Gbps throughput, it is necessary to ensure that there is enough PCIe bandwidth available to the adapter or LAN-on-Motherboard connection. With regard to PCIe requirements, one must consider both speed (for example, PCIe Gen 1 or PCIe Gen 2) and channel width (for example, x4 or x8). The minimum PCIe requirements for one-port and two-port server adapters are as follows:

- **One-port 10GbE server adapters** require PCIe Gen 1 x8 or PCIe Gen 2 x4 connections
- **Two-port 10GbE server adapters** require PCIe Gen 2 x8 connections

While two-port PCIe Gen 1 adapters are available, it is important to note that the maximum unidirectional bandwidth of about 12 Gbps is shared between the two ports. Most of the Gen 1 adapters were released prior to Gen 2 being widely available and were the only option. Further, the size of a physical PCIe connector is not necessarily an accurate indication of channel width. For example, an x8 physical connector on a motherboard may provide only x4 or even x1 connectivity. Refer to the server documentation to verify the proper adapter placement.

Testing shows a bidirectional limit of approximately 25 Gbps when using a dual-port adapter on a PCIe Gen 1 connection. PCIe Gen 2 can provide enough bandwidth for a dual-port 10GbE adapter, and when used in conjunction with NetQueue, near-native throughput is possible. Testing shows greater than 30 Gbps of bidirectional traffic across a two-port 10GbE adapter using a PCIe Gen 2 connection and larger buffer size, as shown in Figure 1.[6] The maximum bandwidth that ESX* can use will continue to increase as newer and more powerful server platforms become available, making this an important best practice going forward. Also note that while PCIe Gen 3 will be available on servers by the end of 2011, the additional bandwidth is not needed on dual-port 10GbE controllers for full bidirectional line rate.

QoS Best Practice 2: Use VMware NetQueue with VMDq-enabled Intel® Ethernet 10 Gigabit Controllers

VMware supports NetQueue, a performance technology that significantly improves performance in 10GbE virtualized environments by aligning and spreading the network I/O processing across multiple processor cores. Even with today's high performance servers and hypervisor improvements, it can be difficult to achieve near-native performance without spreading the load across multiple cores. While NetQueue is enabled by default in ESX versions 4.0 and 4.1, when using Ethernet controllers that support NetQueue, the following procedure can be used to verify that it is enabled.[7]

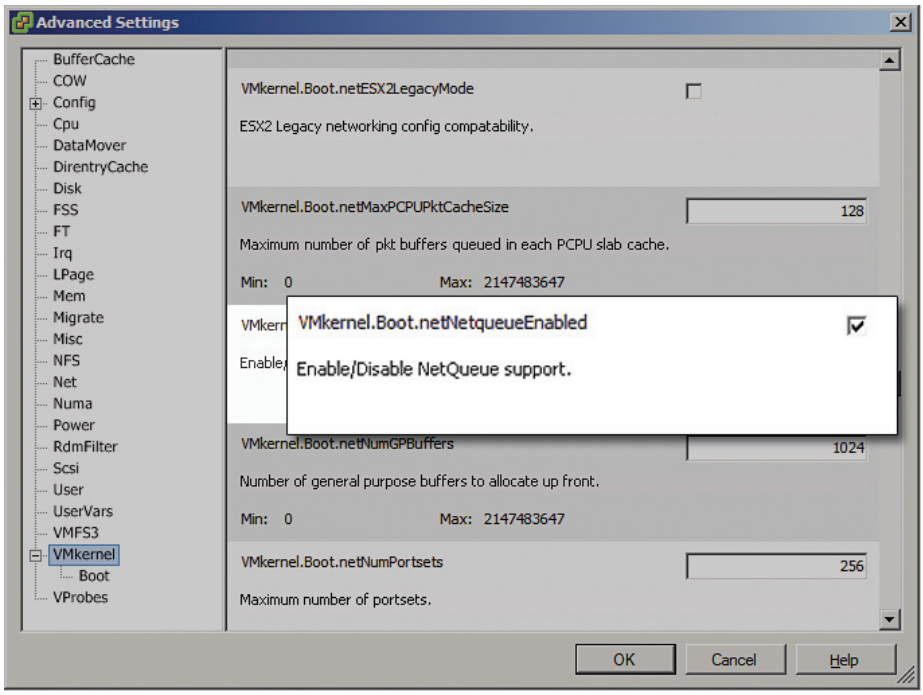

Figure 2. NetQueue* is enabled in the VMkernel using the Advanced Settings dialogue box.

Name	PCI	Driver	Link	Speed	Duplex	MAC Address	MTU	Description
vmnic0	0000:04:00.00	ixgbe	Up	10000Mbps	Full	00:1b:21:5a:6b:28	1500	Intel Corporation Ethernet Server Adapter X520-2
vmnic1	0000:04:00.01	ixgbe	Down	0Mbps	Half	00:1b:21:5a:6b:29	1500	Intel Corporation Ethernet Server Adapter X520-2
vmnic2	0000:07:00.00	ixgbe	Up	10000Mbps	Full	00:1b:21:5a:69:f0	1500	Intel Corporation Ethernet Server Adapter X520-2
vmnic3	0000:07:00.01	ixgbe	Down	0Mbps	Half	00:1b:21:5a:69:f1	1500	Intel Corporation Ethernet Server Adapter X520-2

Figure 3. Output of esxcfg-nics -l, showing two 10GbE controllers, each with one port active and one port ready for future expansion.

To verify that VMDq has been successfully enabled:

1. Verify that NetQueue is Enabled in the VMkernel using the VMware Infrastructure (VI) Client:
   a. Choose **Configuration** > **Advanced Settings** > **VMkernel**
   b. Verify that the **VMkernel.Boot. netNetqueueEnabled** check box is selected (see Figure 2)
2. Query which ports have loaded the driver using esxcfg-nics -l. Then query the statistics using ethtool. If VMDq is enabled, statistics for multiple transmit and receive queues are shown.
   # esxcfg-nics –l

This command provides a list of vmnics that are on the host and provides the vmnic # to use in the next command to list NIC statistics to show all the queues available on the specific vmnic.

3. Use the ethtool –S command to show the vmnic statics.
   # ethtool -S vmnicN
   Where N is the vmnic # of the vmnic that you want the statics on.

While all Intel Ethernet 10GbE controllers have multiple receive queues to scale across multiple CPU cores and help provide near-native 10 Gbps throughput, similar controllers from other manufacturers may not have this functionality. Additionally, not all adapters that support multiple receive queues have driver and hardware support to achieve near-10 Gbps throughput. Therefore, it is critical to test the maximum throughput of an adapter when deciding whether it will meet a specific set of requirements.

NIC Statistics

```
rx_packets: 201457868
tx_packets: 21707296
rx_bytes: 297674697298
tx_bytes: 5885774324
lsc_int: 1
tx_busy: 0
non_eop_descs: 0
rx_errors: 0
tx_errors: 0
rx_dropped: 0
tx_dropped: 0
multicast: 2001990
broadcast: 70459
rx_no_buffer_count: 0
collisions: 0
rx_over_errors: 0
rx_crc_errors: 0
rx_frame_errors: 0
rx_fifo_errors: 0
rx_missed_errors: 0
tx_aborted_errors: 0
tx_carrier_errors: 0
tx_fifo_errors: 0
tx_heartbeat_errors: 0
tx_timeout_count: 0
tx_restart_queue: 0
rx_long_length_errors: 0
rx_short_length_errors: 0
tx_tcp4_seg_ctxt: 883844
tx_tcp6_seg_ctxt: 0
tx_flow_control_xon: 0
rx_flow_control_xon: 0
tx_flow_control_xoff: 0
rx_flow_control_xoff: 0
rx_csum_offload_good:
 198662157
rx_csum_offload_errors: 0
tx_csum_offload_ctxt:
 18405204
low_latency_interrupt: 0
alloc_rx_page_failed: 0
rx_hdr_split: 0
alloc_rx_buff_failed: 0
rx_no_dma_resources: 0
hw_rsc_count: 0
tx_queue_0_packets: 21707300
tx_queue_0_bytes: 5885775142
tx_queue_1_packets: 0
tx_queue_1_bytes: 0
tx_queue_2_packets: 0
tx_queue_2_bytes: 0
tx_queue_3_packets: 0
tx_queue_3_bytes: 0
tx_queue_4_packets: 0
tx_queue_4_bytes: 0
tx_queue_5_packets: 0
tx_queue_5_bytes: 0
tx_queue_6_packets: 0
tx_queue_6_bytes: 0
tx_queue_7_packets: 0
tx_queue_7_bytes: 0
rx_queue_0_packets: 11312977
rx_queue_0_bytes: 9925726966
rx_queue_1_packets: 36895994
rx_queue_1_bytes:
 55736076500
rx_queue_2_packets: 47142685
rx_queue_2_bytes: 71373743266
rx_queue_3_packets: 44314104
rx_queue_3_bytes:
 67091299624
rx_queue_4_packets: 45046987
rx_queue_4_bytes:
 68200883038
rx_queue_5_packets: 5058088
rx_queue_5_bytes: 7657705704
rx_queue_6_packets: 4118692
rx_queue_6_bytes: 6231167104
rx_queue_7_packets: 8053350
rx_queue_7_bytes: 12192756320
```

Figure 4. Output of ethtool -S vmnicN.

QoS Best Practice 3: Use VMXNET3 Virtual Network Device in Microsoft Windows* VMs

It is important to understand the different VM drivers used in vSphere so the bandwidth shown in the monitoring tools can be associated with the correct types of traffic and network administrators can verify that the correct one is installed. There are three types of virtual network adapters available for VMs in VMware vSphere:

- **vmxnet** is a paravirtualized device that works only if VMware Tools is installed within the guest OS. This adapter is optimized for virtual environments and designed for high performance.
- **vlance** emulates the AMD Lance* PCNet32 Ethernet adapter. It is compatible with most 32-bit guest OSs and can be used without VMware Tools.
- **e1000** emulates the Intel® Gigabit Ethernet adapters and is used in either 64-bit or 32-bit VMs. It can be used without VMware Tools.

Two other virtual adapters are available through VMware technology. **Vswif** is a paravirtualized device similar to vmxnet that the VMware ESX service console uses. **Vmknic** is a device in the VMkernel that the TCP/IP stack uses to serve Network File System (NFS) and software iSCSI clients.

On Intel® architecture-based servers that use Intel® Ethernet controllers and adapters, check to ensure that the VMXNET3 Virtual Network Device is enabled in each VM. This practice provides the latest performance enhancements to minimize the overhead of network virtualization. VMXNET3 achieves a higher throughput than enhanced VMXNET2 for a majority of the tests on Microsoft Windows Server* 2008. Refer to the VMware paper "Performance Evaluation of VMXNET3 Virtual Network Device"[8] for more details.

QoS Best Practice 4 : Use Dynamic Logical Segmentation across Two 10GbE Uplinks to Increase Bandwidth and Balance Loads

The introduction of Dynamic Logical Segmentation in vSphere 4.1 results in subtle but important changes to best practice 4 in the paper, "Simplify VMware vSphere* 4 Networking with Intel® Ethernet 10 Gigabit Server Adapters."[1]

An issue with most port-teaming solutions is that VM traffic is allocated to a specific port and more or less stays on that port. This can cause some ports to have heavy traffic while others are underutilized. Load-based teaming (LBT) is a new traffic-management feature of the vNetwork Distributed Switch (vDS) introduced with vSphere 4.1. LBT avoids network congestion on the ESX/ESXi host uplinks caused by imbalances in the mapping of traffic to those uplinks. This feature enables customers to optimally use and balance network loads over the available physical uplinks attached to each ESX/ESXi host. LBT helps avoid situations where one link may be congested while others are relatively underused.

LBT dynamically adjusts the mapping of virtual ports to physical NICs to best balance the network load entering or leaving the ESX/ESXi 4.1 host. When LBT detects an ingress- or egress-congestion condition on an uplink, signified by a mean utilization of 75 percent or more over a 30-second period, it will attempt to move one or more of the virtual ports to less-used links within the team. LBT is an additional load-balancing policy available within the teaming and failover of a dvPortGroup on a vDS. LBT appears as the option **Route based on physical NIC load**, as shown in Figure 5. LBT is not available on the vNetwork Standard Switch (vSS).

By ensuring optimal utilization of all server adapters, the load-based teaming capability of VMware vSphere 4.1 prevents one port from being overloaded while others may be underutilized, as illustrated in Figure 6. Therefore, a smaller number of ports can support a larger amount of network traffic, at the same time helping to ensure high levels of network performance for the virtualized environment as a whole.

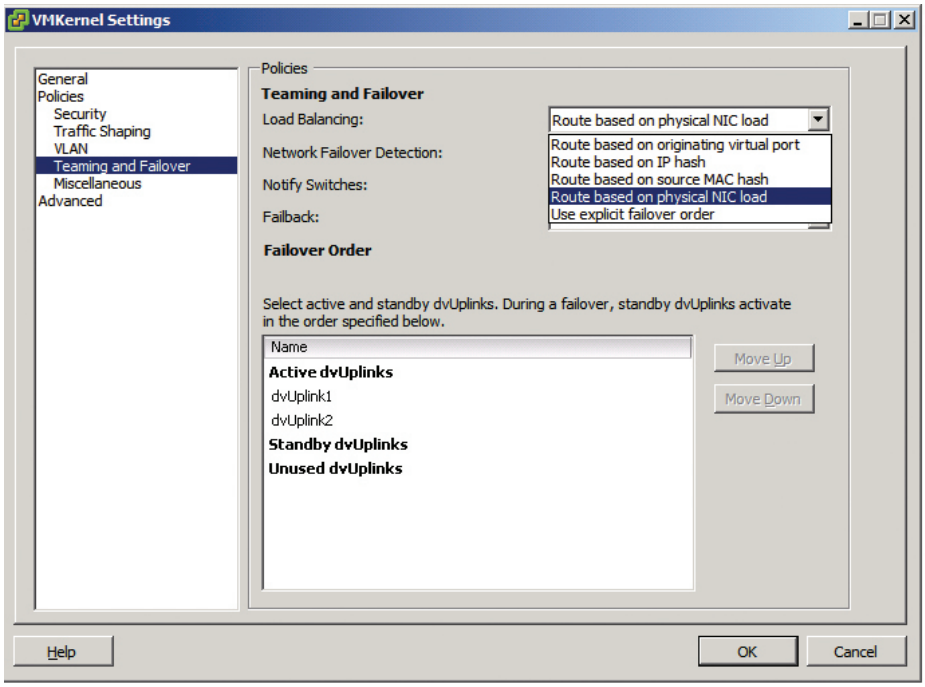

Figure 5. Load-based teaming is one of several load-balancing policy options available within the teaming and failover of a dvPortGroup on a vNetwork Distributed Switch.

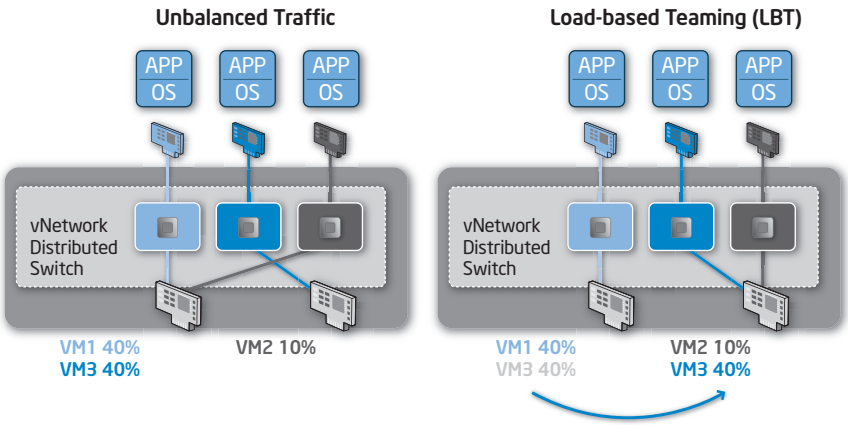

Figure 6. Load-based teaming can help balance the traffic load between physical server adapters.

QoS Best Practice 5: Determine Performance of Native versus Offload Configurations

Testing in Intel's performance labs shows some native software initiators outperform hardware-offloaded functions. For example, hardware-based initiators that are found on iSCSI offloads or Host Bus Adapters (HBAs) might not have equivalent or similar performance benefits on the most current server platforms as compared to those they had on older platforms. In some cases, CPU utilization might be lower using hardware offloads, but the overall number of I/O operations per second is also significantly lower.

With the power and speed of today's server platforms and 10GbE network connections, offload processors can easily become overwhelmed, requiring flow controls to be put in place to reduce the amount of data submitted to the offload processor. Such flow controls can reduce the benefits of offload processors significantly.

VMware has made significant performance improvements for iSCSI storage, including iSCSI boot support on Intel® Ethernet adapters. Using software initiators instead of an offload allows IT organizations to take advantage of a combination of new in-guest virtualization-optimized iSCSI drivers and VMkernel-level storage stack optimizations. These factors can dramatically improve performance for I/O-intensive applications such as databases and messaging.

Several features in vSphere 4.1 also provide prioritization for network and storage I/O that cannot be used if the traffic is offloaded to an HBA. The tradeoffs associated with not using software-controlled initiators may be prohibitive on newer platforms. Having to use a separate management and monitoring tool to configure the hardware can also add significant complexity and cost to the solution.

To determine whether an offload is needed, run workloads that correspond to both maximum throughput and real-world scenarios using the offload enabled, and compare the performance results to similar cases when the the OS-native software initiators are used instead of offloading the function. It is also necessary to check VMware's documentation to determine the compatibility of individual vSphere 4.1 functions and features with offload. See the ESX Configuration Guide[9] and the ESXi Configuration Guide.[10]

ESX can use different types of adapters to provide iSCSI connections. Two examples are native software-based iSCSI adapter/initiators and dependent hardware iSCSI adapters, as shown in Figure 7 and described below.

Native Software-based iSCSI Adapter/Initiators

A software iSCSI adapter is a VMware code built into the VMkernel. It allows a host to connect to the iSCSI storage device through standard network adapters. The software iSCSI adapter handles iSCSI processing while communicating with the network adapter. The software iSCSI adapter allows the use of iSCSI technology without the need to purchase specialized hardware. The host needs only a standard network adapter for network connectivity. iSCSI and network processing is done primarily by a host CPU but can be assisted by stateless offloads in the adapter/controller.

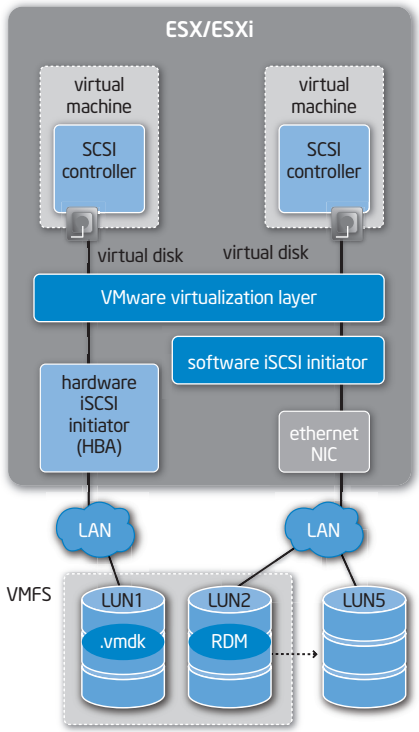

Figure 7. Native software-based iSCSI adapter/initiators and dependent hardware iSCSI adapters are examples of adapters ESX uses to provide iSCSI connections.

Dependent Hardware iSCSI Adapters

Dependent hardware iSCSI adapters depend on VMware networking and iSCSI configuration and management interfaces provided by VMware. This type of adapter can be a card that presents a standard network adapter and iSCSI offload functionality for the same port. The iSCSI offload functionality depends on the host's network configuration to obtain the IP, MAC, and other parameters used for iSCSI sessions. An example of a dependent adapter is the iSCSI-licensed Broadcom 5709 NIC. Hardware iSCSI adapters might need to be licensed to appear in the vSphere Client or vSphere command-line interface. A hardware iSCSI adapter is a third-party adapter that offloads iSCSI and network processing from the host.

Dependent Hardware iSCSI Adapter Limitations

The following limitations relate to the use of dependent hardware iSCSI adapters:

- **No IPv6:** IPv6 configuration cannot be used with dependent hardware iSCSI adapters.
- **No DHCP:** DHCP cannot be used with dependent hardware iSCSI adapters.
- **No Routing:** Routing is not available for dependent hardware iSCSI adapters.
- **No Network I/O Control:** The iSCSI traffic resource pool shares do not apply to iSCSI traffic on a dependent hardware iSCSI adapter.
- **No Traffic Shaping:** A traffic-shaping policy is defined by three characteristics: average bandwidth, peak bandwidth, and burst size. Traffic-shaping policies do not apply to iSCSI traffic on a dependent hardware iSCSI adapter.
- **No iSCSI traffic reporting:** When a dependent hardware iSCSI adapter is used, performance reporting for a NIC associated with the adapter might show little or no activity, even when iSCSI traffic is heavy. This behavior occurs because the iSCSI traffic bypasses the regular networking stack.
- **Security Vulnerabilities:** If any security vulnerabilities exist in the iSCSI device software, data can be at risk through no fault of ESX. To lower this risk, network administrators should install all security patches that the storage equipment manufacturer provides and limit the devices connected to the iSCSI network.

Dependent Hardware iSCSI Considerations from VMware Regarding Dependent Hardware iSCSI Adapters

When dependent hardware iSCSI adapters are used with ESX, certain considerations apply:

- When any dependent hardware iSCSI adapter is used, performance reporting for a NIC associated with the adapter might show little or no activity, even when iSCSI traffic is heavy. This behavior occurs because the iSCSI traffic bypasses the regular networking stack.
- Some dependent hardware iSCSI adapters perform data reassembly in hardware, which has a limited buffer space.
- When a dependent hardware iSCSI adapter is used in a congested network or under load, network administrators should enable flow control to avoid performance degradation. Flow control manages the rate of data transmission between two nodes to prevent a fast sender from overrunning a slow receiver. For best results, enable flow control at the end points of the I/O path, at the hosts and iSCSI storage systems.
- Some dependent hardware iSCSI adapters do not support IPv6 and Jumbo Frames. Check VMware's documentation for specific adapter limitations.

Note that other feature incompatibilities may exist when using offload engines. For example, vCenter Server Heartbeat* is incompatible with TCP Offload Engine (TOE), a common feature of some non-Intel 10GbE cards. Because vCenter Server Heartbeat is intended to manage the passing or filtering of selected IP addresses, the following TOE features must be disabled on all network adapters prior to installing vCenter Server Heartbeat:

- Offload IP Security
- Offload TCP Segmentation
- Offload TCP/IP Checksum

Using VMware NetQueue with Intel VMDq, VMXNET3, PCIe Gen 2 x8 connections, and LBT will provide the highest levels of performance so the next step of monitoring bandwidth and traffic will provide the best data on which to make control decisions.

Best Practices for QoS Monitoring

The key to understanding the needs for QoS controls is to test the network configuration with benchmarking and load-generation tools to determine maximum throughput and typical or "real-world" workloads. The basic rule here is to understand the bottlenecks and assess their impact.

QoS Best Practice 6: Use resxtop and vSphere Management Assistant to View and Monitor Network Performance

Deploying the vSphere Management Assistant (vMA) allows you to run resxtop[11] (remote esxtop) from the command line and remotely connect to ESX/ESXi hosts directly or through vCenter* Server to monitor various aspects of performance. The Network panel in **resxtop** displays server-wide network utilization statistics.

Statistics are arranged by port for each virtual network device configured. For physical network adapter statistics, see the row in Table 1 that corresponds to the port to which the physical network adapter is connected. For statistics on a virtual network adapter configured in a particular VM, see the row corresponding to the port to which the virtual network adapter is connected.

COLUMN	DESCRIPTION
PORT-ID	Virtual network device port ID
UPLINK	Y means the corresponding port is an uplink; N means it is not
UP	Y means the corresponding link is up; N means it is not
SPEED	Link speed in megabits per second
FDUPLX	Y means the corresponding link is operating at full duplex; N means it is not
USED-BY	Virtual network device port user
DTYP	Virtual network device type: H means "hub" and S means "switch"
DNAME	Virtual network device name
PKTTX/s	Number of packets transmitted per second
PKTRX/s	Number of packets received per second
MbTX/s	Megabits transmitted per second
MbRX/s	Megabits received per second
%DRPTX	Percentage of transmit packets dropped
%DRPRX	Percentage of receive packets dropped
TEAM-PNIC	Name of the physical NIC used for the team uplink
PKTTXMUL/s	Number of multicast packets transmitted per second
PKTRXMUL/s	Number of multicast packets received per second
PKTTXBRD/s	Number of broadcast packets transmitted per second
PKTRXBRD/s	Number of broadcast packets received per second

Table 1. Network panel statistics

Using a traffic generator such as NTTTCP or NetPerf to send traffic to multiple VMs on a host will drive receive-side traffic on the port groups associated with the target VMs, while using a VMkernel feature such as vMotion will show traffic on the port associated with the VMkernel. This allows the setting up of different port groups with different traffic loads while using different kernel features to see how much traffic is being generated and what the maximum bandwidth is on the adapters. There are several video demos that show different configurations posted on the Intel® Server Room site and YouTube*.

Utilizing a 10GbE connection, vMotion under vSphere 4.1 can use up to 8 Gbps of aggregate bandwidth, as opposed to approximately 1 Gbps in ESX 3.5 and 2.6 Gbps in ESX 4.0, as shown in Figure 8.

Even with greater than 9.5 Gbps per port being sent to the VMs, vMotion is able to move up to eight VMs concurrently, and VMware vCenter* Server can adjust the amount of bandwidth allocated to vMotion so the VM traffic is not significantly affected by vMotion activity.

While monitoring with **esxtop** or **resxtop**, VMkernel traffic can be seen, together with all the other traffic on the different port groups. The best practice of using port groups to separate traffic types is an easy way to see how increasing one type of traffic affects others.

The increase of vMotion bandwidth also emphasizes the point that the advances in VMkernel functions are driving the need for 10GbE faster than actual VM-generated traffic. While some of these functions do not have consistent traffic, they can benefit from the higher bandwidth that 10GbE can provide. It also supports the move from the old GbE-based paradigm of providing dedicated ports to specific functions. This new paradigm of providing 10GbE uplinks to the vDS and allowing all traffic types to have access to the potential bandwidth will provide increased performance while simplifying network deployments.

Network architects should keep in mind that to fully test a 10GbE network connection, artificially exaggerated traffic might be required. Even so, such testing allows for throughput and impact modeling that can be extremely helpful in determining what kind of control measures need to be deployed.

Note: When using a dependent hardware iSCSI adapter, performance reporting for a NIC associated with the adapter might show little or no activity, even when iSCSI traffic is heavy. This behavior occurs because the iSCSI traffic bypasses the regular networking stack and needs to be calculated into the overall network bandwidth requirements.

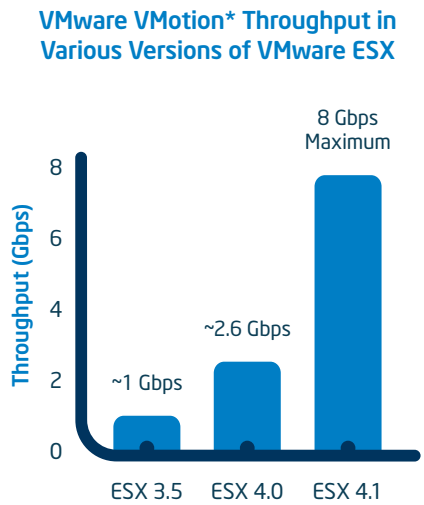

Figure 8. Successive versions of VMware ESX* each support higher levels of throughput for vMotion*.

Network Performance Enhancements in VMware vSphere* 4.1 to Test

vSphere 4.1 incorporates a number of network performance enhancements that affect native guest VM throughput and VMkernel-based ESX/ESXi applications, such as vMotion, VMware Fault Tolerance (FT) Logging, and NFS. These improvements include the following (note that observed performance increases will vary according to the platform and other external factors):

- **vMotion throughput.** Increases can generate as much as a 50 percent reduction in the time required to migrate a VM from one host to another.
- **vMotion concurrency.** vSphere will automatically increase the maximum number of concurrently allowed vMotion instances to eight (up from a maximum of two with vSphere 4.0) when 10GbE uplinks are employed.
- **NFS.** Throughput is increased for both read and write operations.
- **Native VM throughput.** This quantity also increases by 10 percent going out to the physical network - this is directly related to the vmxnet3 enhancements.
- **VM-to-VM Traffic.** In vSphere 4.1, the VM-to-VM traffic throughput improved by 2x, to up to 19 Gbps.

For more information on the specific enhancements in VMware vSphere 4.1, see the VMware document, "What's New in vSphere 4.1."[12]

Best Practices for QoS Control

Before any bandwidth control measures are taken, it is critical that thorough consideration be given to the actual bandwidth being used under maximum and expected workloads. The key is to note that two 10GbE ports can provide more than double the bidirectional bandwidth of as many as eight to 12 GbE ports, so many of the concerns of bandwidth contention found in a GbE network are not present in a 10GbE network. Like most QoS controls, they should be implemented only when actually needed, based on observed data.

The best practices described in this section are designed to remedy those situations where analysis and monitoring of the network shows that QoS issues are present, although lab testing suggests that QoS issues are unlikely to arise when using 10GbE networking.

The key to controlling traffic is to maintain the 10GbE connections as single uplink ports in the hypervisor. This practice enables unused throughput from one group to be used by other groups if needed. In addition to enabling all traffic types to take advantage of the 10GbE infrastructure, the environment is also less complex. The best practices in this section should be used only if network monitoring shows contention.

QoS Best Practice 7: Use Network I/O Control and Storage I/O Control to Handle Contention on Unified Networks

The Network I/O Control (NetIOC) feature available in vSphere 4.1 introduces a software-based approach to partitioning physical network bandwidth among the different types of network traffic flows. It does so by providing appropriate QoS policies enforcing traffic isolation, predictability, and prioritization, helping IT organizations overcome the contention that may arise as the result of consolidation. The experiments conducted in VMware performance labs using industry-standard workloads show that NetIOC:

- Maintains NFS and/or iSCSI storage performance in the presence of other network traffic such as vMotion and bursty VMs.
- Provides network service level guarantees for critical VMs.
- Ensures adequate bandwidth for VMware FT logging.
- Ensures predictable vMotion performance and duration.
- Facilitates situations where a minimum or weighted level of service is required for a particular traffic type, independent of other traffic types.

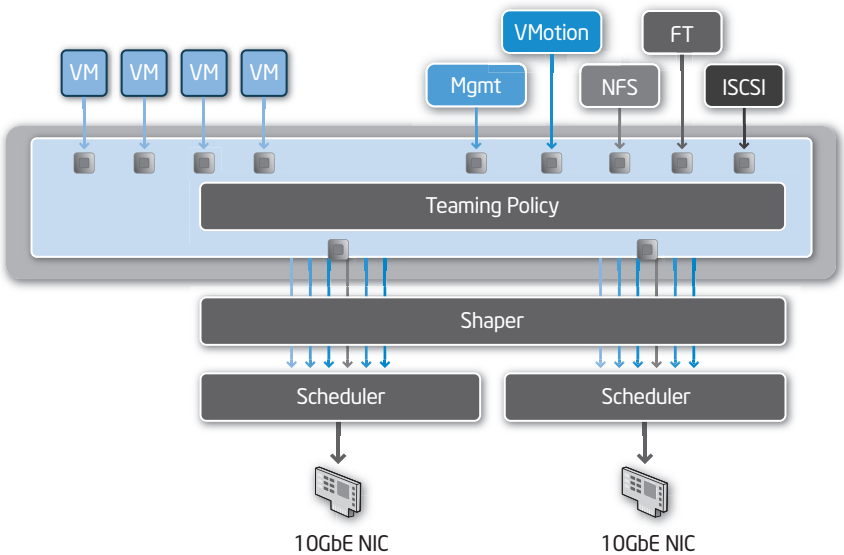

Figure 9. The NetIOC software-based scheduler manages bandwidth resources among varous types of traffic.

Resource Management Using Network I/O Control

As shown in Figure 9, NetIOC implements a software scheduler within the vDS to isolate and prioritize specific traffic types contending for bandwidth on the uplinks connecting ESX/ESXi 4.1 hosts with the physical network. NetIOC is able to individually identify and prioritize the following traffic types leaving an ESX/ESXi host on a vDS-connected uplink:

- VM traffic
- Management traffic
- iSCSI
- NFS
- VMware FT logging
- vMotion

NetIOC is particularly applicable to environments where multiple traffic types are converged over a pair of 10GbE interfaces. If an interface is oversubscribed (that is, more than 10 Gbps of data is contending for a 10GbE interface), NetIOC is able to ensure each traffic type is given a selectable and configurable minimum level of service.

Moving from GbE to 10GbE networking typically involves converging traffic from multiple GbE server adapters onto a smaller number of 10GbE ones, as shown in Figure 10. On the top of the figure, dedicated server adapters are used for several types of traffic, including iSCSI, VMware FT, vMotion and NFS. On the bottom of the figure, those traffic classes are all converged onto a single 10GbE server adapter, with the other adapter handling VM traffic.

In the case shown in Figure 10, the total bandwidth for VM traffic has gone from 4 Gbps to 10 Gbps, providing a nominal 2.5x increase, which should easily support the existing traffic and provide substantial headroom for growth and usage peaks. At the same time, however, some network administrators might want to explicitly address cases where different types of network traffic could contend for network bandwidth; for example, prioritizing certain traffic with particularly stringent latency requirements.

The first paper in this series[1] describes the value of data center bridging (DCB) to traffic prioritization within a single physical server adapter. Intel worked with the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF) to develop standards for DCB, which is supported in Intel Ethernet 10 Gigabit Server Adapter products. This standard is still being implemented in other elements of the network infrastructure, so VMware has built similar technology into VMware vSphere 4.1 using NetIOC, which can help administrators take optimal advantage of network bandwidth and guarantee minimum service levels for specific classes of traffic.

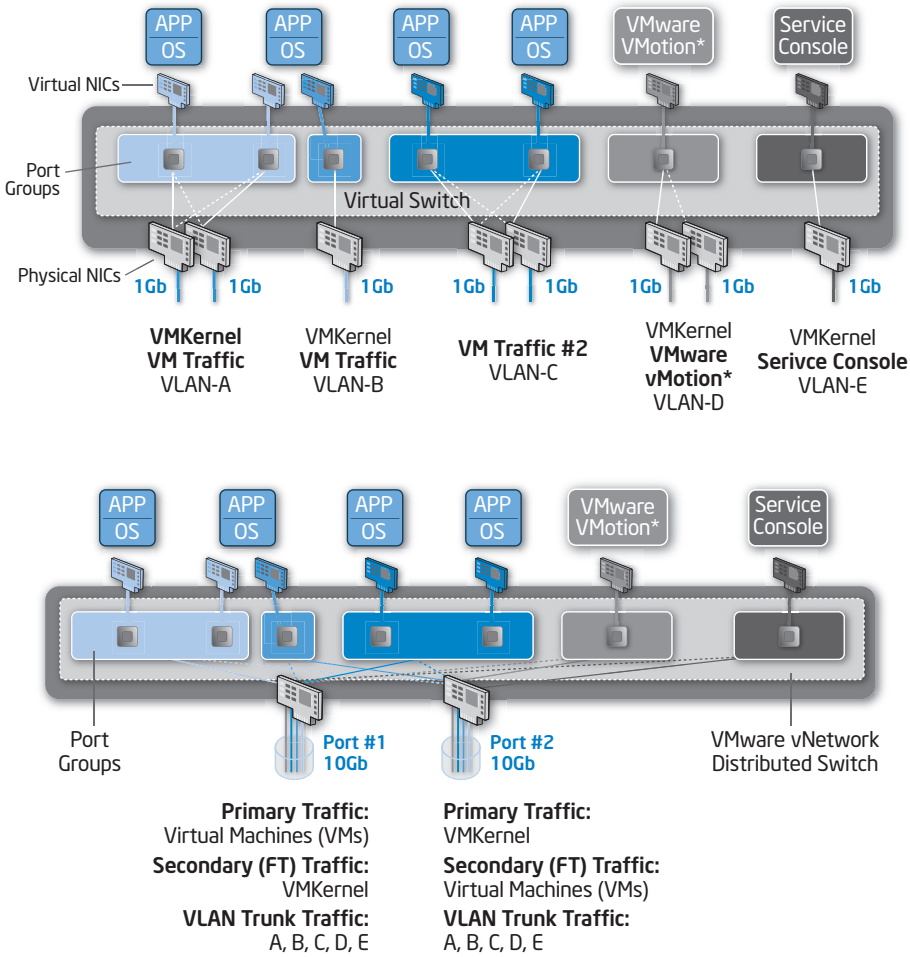

Figure 10. Traffic from multiple GbE server connections may be converged onto two 10GbE uplinks.

Figure 11 shows how NetIOC is configured through the vSphere Client on vCenter Server. The **Resource Allocation** tab within the vDS enables administrators to specify maximum (Host Limit, measured in megabits per second (Mb/s)) and minimum (Shares Value, represented as a proportion of the total) bandwidth on the physical server adapter to each traffic class.

In this example, the aggregate VM traffic is subject to a limitation of 500 Mb/s of bandwidth, regardless of how much is available. Because the assigned Shares Values add up to a total of 400 shares (100+100+50+50+50+50), and VM traffic has been assigned a minimum of 50 shares, it is guaranteed a minimum of 50/400, or one-eighth, of the total bandwidth available from the physical server adapter. This aspect of this new 4.1 feature specifically addresses the concerns that many administrators have voiced when discussing the move away from dedicated GbE ports to shared 10GbE ports.

Partitioning traffic the old-school way, either by physically segmenting the traffic on dedicated GbE ports or physically dividing up a 10GbE port into multiple ports with dedicated bandwidth limits using proprietary technologies adds unnecessary complexity and cost. NetIOC is a more effective way to segregate bandwidth because it is dynamic and limits traffic only when there is congestion on the port. The other methods place static limits and leave significant bandwidth unused, significantly reducing the value that 10GbE brings to virtualization.

Note: NetIOC does not support the use of a dependent hardware iSCSI adapters. The iSCSI traffic resource pool shares do not apply to iSCSI traffic on a dependent hardware iSCSI adapter.

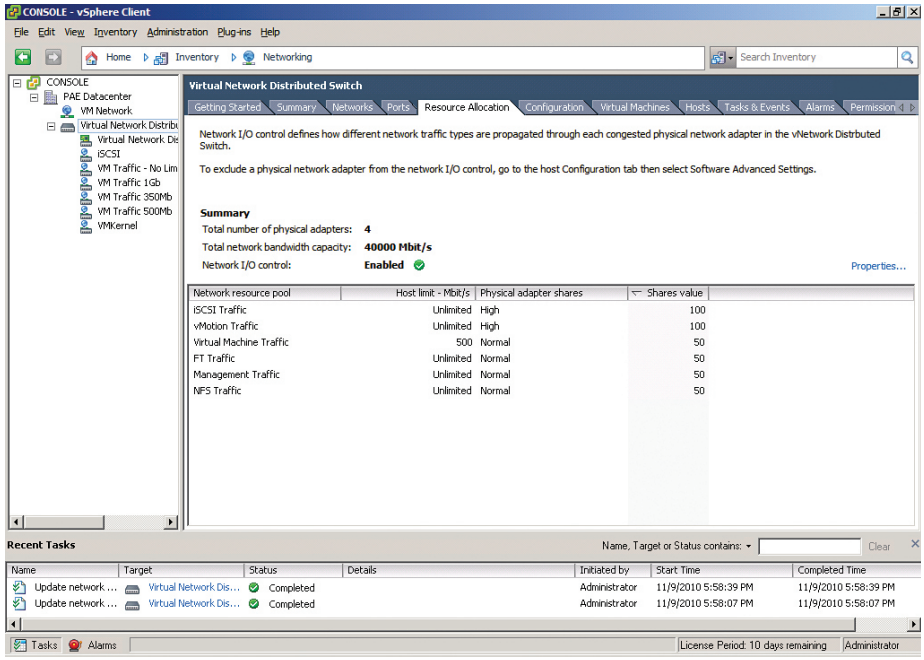

Figure 11. I/O Control configuration is performed from the vSphere* client.

Configuring the Shares Value

The **Shares** value specifies the relative importance of a traffic type scheduled for transmission on a physical server adapter. Shares are specified in abstract units between 1 and 100. The bandwidth for the link is divided among the traffic types according to their relative shares value. For example, consider the case of two 10GbE links; for a total of 20Gbps of bandwidth in each direction, with VM traffic set to 100 shares, vMotion traffic set to 50 shares, and VMware FT logging traffic set to 50 shares.

If VM traffic and vMotion traffic are both contending for the bandwidth on teamed 10GbE ports, the VM traffic (100 shares) will get 67 percent (13.4 Gbps) of the link, and vMotion (50 shares) will get 33 percent (6.7 Gbps) of the link. If all three of these traffic types are active and contending for the link, VM traffic (100 shares) will get 50 percent (10 Gbps), vMotion (50 shares) will get 25 percent (5 Gbps), and VMware FT logging (50 shares) will get 25 percent (5 Gbps). If no other traffic types are contending for the link at that moment, each traffic type can consume the entire link (or up to the host limit, if set).

Configuring the Limits Value

The **Limits** value specifies an absolute maximum limit on egress traffic for that traffic type on a host. Limits are specified in Mb/s. The limit is an aggregate for that traffic type and applies regardless of the number of physical server adapters in the NIC team.

Note: Limits are applied to the network traffic before the shares. Limits apply over a team, while shares schedule and prioritize traffic for each physical server adapter.

NetIOC Usage

Unlike limits, which are specified in absolute units of Mb/s, shares are used to specify the relative importance of specific flows. Shares are specified in abstract units with a value ranging from 1 to 100. In this section, an example describes the usage of shares.

Figure 12 highlights the following characteristics of shares:

- **In the absence of any other traffic**, a particular traffic flow gets 100 percent of the bandwidth available, even if it was configured with just 25 shares.
- **During periods of contention**, bandwidth is divided among the traffic flows based on their relative shares.

For further details, refer to the document, "VMware Network I/O Control: Architecture, Performance and Best Practices."[13]

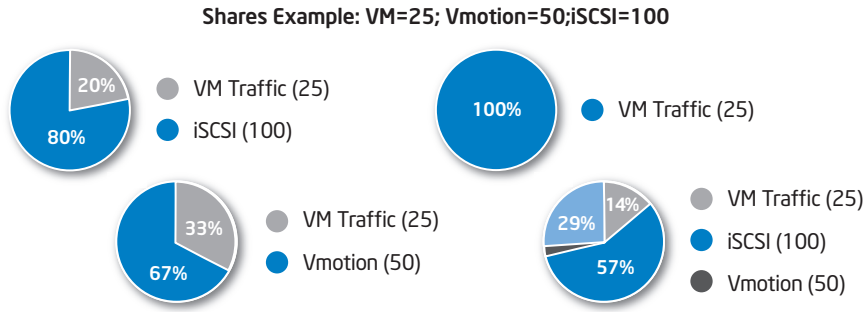

Figure 12. Shares values assign the relative importance to various types of traffic flows.

Storage I/O Control

To enforce QoS, it is necessary to be able to set the priority of access to central data stores by VMs across a virtualized environment. That is, higher-priority requests for data must take precedence over other types of requests to ensure appropriate latencies. VMware vSphere 4.1 achieves that prioritization for specific data stores using a system of shares and limits similar to that described above for NetIOC.

The VMware vSphere 4.1 Storage I/O Control mechanism monitors the latency associated with communication between a VM and data store. If network connectivity to the data store becomes congested (the latency exceeds a threshold defined by the administrator), the mechanism prioritizes access to the data store according to the shares and limits that have been defined in advance to meet QoS requirements.

QoS Best Practice 8: Limit Use of Traffic-Shaping Policies to Control Bandwidth on a Per-Port Basis Only When Needed

Before going into the details of how and when to use traffic shaping, it should be noted that this feature should be used sparingly. It is a somewhat limited way of using static bandwidth limits and segmentation to control traffic on a per-virtual-port basis. Traffic shaping on a per-port basis is very similar to the use of other technologies that statically segment 10GbE connections into multiple connections that do not allow bandwidth sharing when bandwidth is available. This method was more useful in older versions of ESX; with the release of 4.1 this approach is not as effective as using traffic-type QoS controls found in vSphere 4.1's Network I/O Control.

Traffic-shaping policies can be established for each port group and each dvPort or dvPort group. ESXi shapes outbound network traffic on vSwitches and both inbound and outbound traffic on a vDS. Traffic shaping restricts the network bandwidth available on a port, but it can also be configured to allow bursts of traffic to flow through at higher speeds. Traffic-shaping policy uses the following parameters:

- **Average bandwidth** establishes the number of bits per second to allow across a port, averaged over time: the allowed average load.
- **Peak bandwidth** is the maximum number of bits per second to allow across a port when it is sending or receiving a burst of traffic. This quantity limits the bandwidth used by a port whenever it is using its burst bonus.

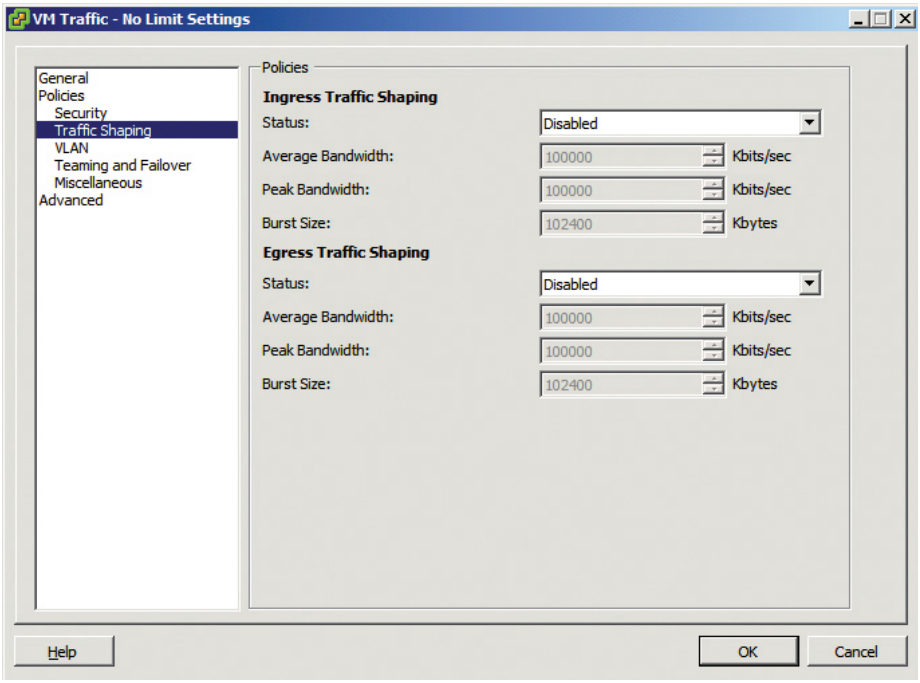

Figure 13. Traffic Shaping Disabled on a vDS.

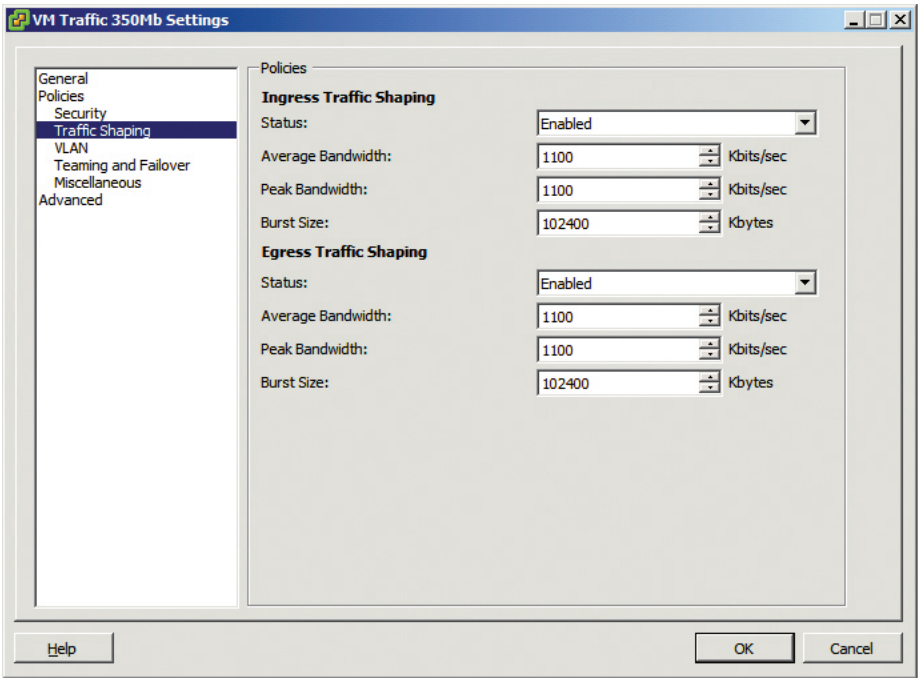

Figure 14. Traffic Shaping Enabled on a vDS.

- **Burst size** is the maximum number of bytes to allow in a burst. If this parameter is set, a port might gain a burst bonus if it does not use all its allocated bandwidth. Whenever the port needs more bandwidth than what is specified by **average bandwidth**, it might be allowed to temporarily transmit data at a higher speed if a burst bonus is available. This parameter limits the number of bytes that have accumulated in the burst bonus and thus transfers at a higher speed.

Traffic shaping can provide bandwidth limits for a dedicated virtual NIC (vNIC) in a VM that can use a lot of bandwidth, such as a vNIC used specifically for backup. Backup can take as much bandwidth as is available, so to limit the amount available, the best practice is to create a separate and dedicated vNIC in the VM and assign it to a port group that has traffic shaping enabled to statically limit the amount of bandwidth. This practice will allow both inbound and outbound traffic bandwidth to be limited to a specific level to provide predictable backup times and bandwidth usage.

Note: Traffic shaping is not supported when a dependent hardware iSCSI adapter is used. A traffic-shaping policy is defined by three characteristics: average bandwidth, peak bandwidth, and burst size. Traffic-shaping policies do not apply to iSCSI traffic on a dependent hardware iSCSI adapter.

Conclusion

VMware vSphere 4.1 represents a significant evolution in administrators' ability to use virtualization to meet advanced business needs. Because network connectivity based on 10GbE is a clear necessity to maximize the value of those advances, vSphere 4.1 is specifically designed with features that complement Intel Ethernet 10 Gigabit Server Adapters. Following the analysis, monitoring, and control best practices presented here will help administrators take full advantage of 10 Gigabit. Together, vSphere 4.1 and Intel Ethernet 10GbE Server Adapters add a new level of performance, flexibility, and intelligence to virtualized networks.

About the Authors

Brian Johnson is a Product Marketing Engineer for 10Gb Intel Ethernet Products at Intel Corporation. In this role he is responsible for product definition, development, and marketing of 10Gb silicon products along with virtualization, manageability, and security technologies. Brian has over 15 years of experience in server product planning and marketing, during which he has held various positions in strategic planning, product marketing, and product development.

Intel Technology Leader **Patrick Kutch** is a Technical Marketing Engineer (TME) for Intel Server I/O Virtualization and Manageability technologies. As a senior TME, he works with customers, providing both educational materials and answering questions ranging from technical to architectural. He has worked in nearly every facet of Server Manageability, from contributing to the IPMI specification to writing management software over a span of 12 years at Intel. Patrick has split his time between Manageability and I/O Virtualization for the past four years. Patrick frequently blogs about his technologies at http://communities.intel.com/community/wired.

Take the next step, and lead where others follow.

For more information about solutions from VMware and Intel, visit:
www.vmwareintelalliance.com

SOLUTION PROVIDED BY:

vmware | intel

The Foundation of Virtualization

[1] http://download.intel.com/support/network/sb/10gbe_vsphere_wp_final.pdf.

[2] http://download.intel.com/support/network/sb/paradigmshiftdatapaper.pdf.

[3] http://communities.intel.com/community/wired/blog/2010/07/07/simplify-vmware-vsphere-4-networking-with-intel-ethernet-10-gigabit-server-adapters.

[4] http://www.vmware.com/.

[5] Available on select Intel® Ethernet Controllers; see http://www.intel.com/network/connectivity/vtc_vmdq.htm.

[6] **Test Configuration:** Ixia* IxChariot* v7.1; 16 Clients Per Port Under Test; High Performance Throughput Script; File Size = 64-1K: 1,000,000 / 2K+:10,000,000 Bytes; Buffer Sizes=64 Bytes to 64 KB; Data Type – Zeroes; Data Verification Disabled; Nagles Disabled

System Under Test: Intel® S5520HC ("Hanlan Creek"); two Intel® Xeon® Processors X5680 (12M Cache, 3.33 GHz, 6.40 GT/s Intel® QPI); Intel® 5520 Chipset ("Tylersburg"); RAM: 12GB DDR3 @ 1333MHz; BIOS: 0050; Windows Server* 2008 R2 x64

Clients: SuperMicro* 6015T-TV; two Intel® Dual Core Xeon® processors 5160 @3.0GHz; 2 GB RAM; Intel® PRO/1000 PT Dual Port Server Adapter - v9.12.13.0 driver; Windows Server 2003 SP2 x64

Network Configuration: Force10 Networks* ExaScale* E1200i switch; Clients connected @ 1 Gbps

[7] Page 61 Chapter 5 Advanced Networking - ESXi Configuration Guide EN-000327-00.

[8] http://www.vmware.com/pdf/vsp_4_vmxnet3_perf.pdf.

[9] http://www.vmware.com/pdf/vsphere4/r41/vsp_41_esx_server_config.pdf.

[10] http://www.vmware.com/pdf/vsphere4/r41/vsp_41_esxi_server_config.pdf.

[11] http://communities.vmware.com/docs/DOC-3930.

[12] http://www.vmware.com/support/vsphere4/doc/vsp_41_new_feat.html.

[13] http://www.vmware.com/files/pdf/techpaper/VMW_Netioc_BestPractices.pdf.

1110/BJ/MESH/PDF ♻ Please Recycle 324580-001US

intel