VMware® Network I/O Control: Architecture, Performance and Best Practices

VMware vSphere™ 4.1

WHITE PAPER

vmware®

Executive Summary

Network convergence using 10GbE technology provides enormous opportunities for IT administrators and architects to simplify the physical network infrastructure while improving performance. Administrators and architects need a simple and reliable way to enable prioritization of critical traffic over the physical network if and when contention for those resources occurs.

The Network I/O Control (NetIOC) feature available in VMware® vSphere™ 4.1 ("vSphere") addresses these challenges by introducing a software approach to partitioning physical network bandwidth among the different types of network traffic flows. It does so by providing appropriate quality of service (QoS) policies enforcing traffic isolation, predictability and prioritization, therefore helping IT organizations overcome the contention resulting from 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 virtual machines.
- Provides network service level guarantees for critical virtual machines.
- Ensures adequate bandwidth for VMware Fault Tolerance (VMware FT) logging.
- Ensures predictable vMotion performance and duration.
- Facilitates any situation where a minimum or weighted level of service is required for a particular traffic type independent of other traffic types.

Organization of This Paper

This paper is organized to help the reader understand the use cases, technology and best practices for using NetIOC in a vSphere environment.

The sections that follow discuss:

- Use cases and application of NetIOC with 10GbE in contrast to traditional 1GbE deployments
- The NetIOC technology and architecture used within the vNetwork Distributed Switch (vDS)
- How to configure NetIOC from the vSphere Client
- Examples of NetIOC usage to illustrate possible deployment scenarios
- Results from actual performance tests using NetIOC to illustrate how NetIOC can protect and prioritize traffic in the face of network contention and oversubscription
- Best practices for deployment

Moving from 1GbE to 10GbE

Virtualized datacenters are characterized by newer and complex types of network traffic flows such as vMotion and VMware FT logging traffic. In today's virtualized datacenters where 10GbE connectivity is still not commonplace, networking is typically based on large numbers of 1GbE physical connections that are used to isolate different types of traffic flows and to provide sufficient bandwidth.

NETWORK TRAFFIC	NICs PROVISIONED
Management Traffic	2GbE NICs
Virtual Machine Traffic	2–4GbE NICs
vMotion Traffic	1GbE NIC
FT Traffic	1GbE NIC
iSCSI Traffic	2GbE NICs
NFS Traffic	2GbE NICs

Table 1. Typical Deployment and Provisioning of 1GbE NICs with vSphere 4.0

Provisioning a large number of GbE network adapters to accommodate peak bandwidth requirements of these different types of traffic flows has a number of shortcomings:

- Limited bandwidth: Flows from an individual source (virtual machine, vMotion interface, and so on) are limited and bound to the bandwidth of a single 1GbE interface even if more bandwidth is available within a team
- Excessive complexity: Use of large numbers of 1GbE adapters per server leads to excessive complexity in cabling and management, with an increased likelihood of misconfiguration
- Higher capital costs: Large numbers of 1GbE adapters requires more physical switch ports, which in turn leads to higher capital costs including additional switches and rack space
- Lower utilization: Static bandwidth allocation to accommodate peak bandwidth for different traffic flows means poor average network bandwidth utilization

10GbE provides ample bandwidth for all the traffic flows to coexist and share the same physical 10GbE link. Flows that were limited to the bandwidth of a single 1GbE link are now able to use as much as 10GbE.

While the use of a 10GbE solution greatly simplifies the networking infrastructure and addresses all the shortcomings listed above, there are a few challenges that still need to be addressed to maximize the value of a 10GbE solution. One means of optimizing the 10GbE network bandwidth is to prioritize the network traffic by traffic flows. This ensures that latency-sensitive and critical traffic flows can access the bandwidth they need.

NetIOC enables the convergence of diverse workloads on a single networking pipe. It provides sufficient controls to the vSphere administrator in the form of limits and shares parameters to enable and ensure predictable network performance when multiple traffic types contend for the same physical network resources.

NetIOC Architecture

NetIOC provides the vSphere administrator with the capability to efficiently and predictably use physical networking resources.

Prerequisites for NetIOC

NetIOC is only supported with the vNetwork Distributed Switch (vDS). With vSphere 4.1, a single vDS can span up to 350 ESX/ESXi hosts, providing a simplified and more powerful management environment versus the per-host switch model using the vNetwork Standard Switch (vSS). The vDS also provides a superset of features and capabilities over that of the vSS, such as network vMotion, bi-directional traffic shaping and private VLANs.

Configuring and managing a vDS involves use of distributed port groups (DV Port Groups) and distributed virtual uplinks (dvUplinks). DV Port Groups are port groups associated with a vDS similar to port groups available with vSS. dvUplinks provide a level of abstraction for the physical NICs (vmnics) on each vSphere host.

NetIOC Feature Set

NetIOC provides users with the following features:

- Isolation: ensure traffic isolation so that a given flow will never be allowed to dominate over others, thus preventing drops and undesired jitter
- Shares: allow flexible networking capacity partitioning to help users to deal with overcommitment when flows compete aggressively for the same resources
- Limits: enforce traffic bandwidth limit on the overall vDS set of dvUplinks
- Load-Based Teaming: efficiently use a vDS set of dvUplinks for networking capacity

NetIOC Traffic Classes

The NetIOC concept revolves around resource pools that are similar in many ways to the ones already existing for CPU and Memory.

NetIOC classifies traffic into six predefined resource pools as follows:

- vMotion
- iSCSI
- FT logging
- Management
- NFS
- Virtual machine traffic

The overall architecture is shown in Figure 1.

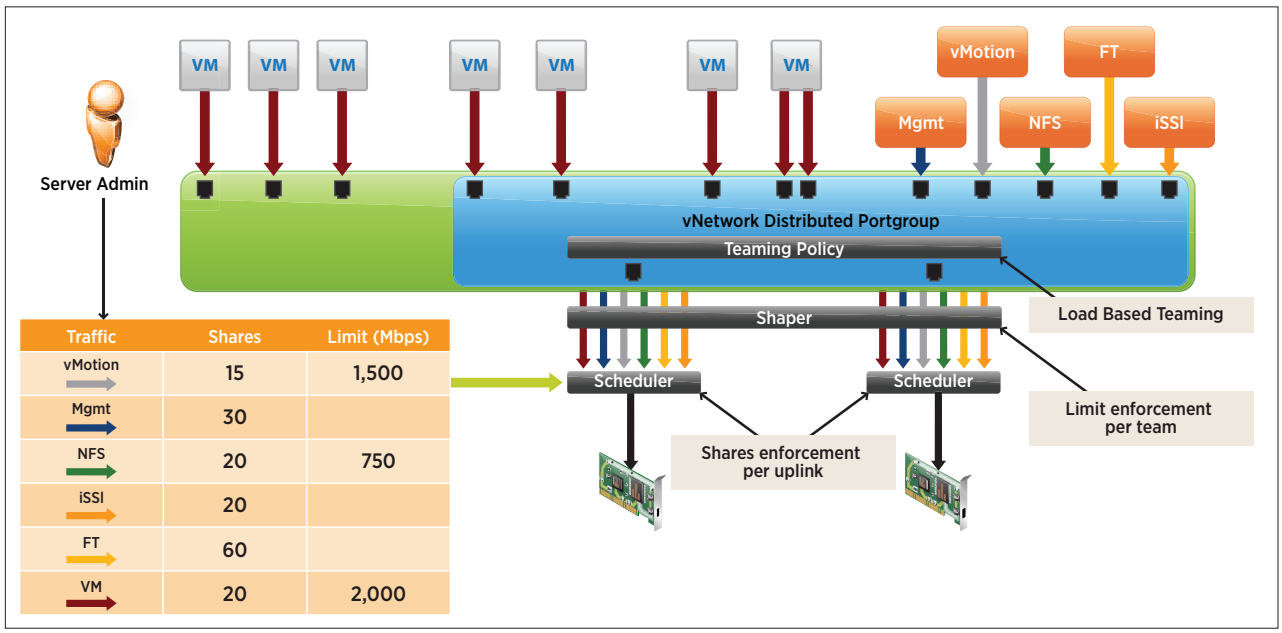

Figure 1. NetIOC Architecture

Shares

A user can specify the relative importance of a given resource-pool flow using shares that are enforced at the dvUplink level. The underlying dvUplink bandwidth is then divided among resource-pool flows based on their relative shares in a work-conserving way. This means that unused capacity will be redistributed to other contending flows and won't go to waste. As shown in Figure 1, the network flow scheduler is the entity responsible for enforcing shares and therefore is in charge of the overall arbitration under overcommitment. Each resource-pool flow has its own dedicated software queue inside the scheduler so that packets from a given resource pool won't be dropped due to high utilization by other flows.

Limits

A user can specify an absolute shaping limit for a given resource-pool flow using a bandwidth capacity limiter. As opposed to shares that are enforced at the dvUplink level, limits are enforced on the overall vDS set of dvUplinks, which means that a flow of a given resource pool will never exceed a given limit for a vDS out of a given vSphere host.

Load-Based Teaming (LBT)

vSphere 4.1 introduces a load-based teaming (LBT) policy that ensures vDS dvUplink capacity is optimized. LBT avoids the situation of other teaming policies in which some of the dvUplinks in a DV Port Group's team were idle while others were completely saturated just because the teaming policy used is statically determined. LBT reshuffles port binding dynamically based on load and dvUplinks usage to make an efficient use of the bandwidth available. LBT only moves ports to dvUplinks configured for the corresponding DV Port Group's team. Note that LBT does not use shares or limits to make its judgment while rebinding ports from one dvUplink to another. LBT is not the default teaming policy in a DV Port Group so it is up to the user to configure it as the active policy.

LBT will only move a flow when the mean send or receive utilization on an uplink exceeds 75 percent of capacity over a 30-second period. LBT will not move flows more often than every 30 seconds.

Configuring NetIOC

NetIOC is configured through the vSphere Client in the Resource Allocation tab of the vDS from within the "Home->Inventory->Networking" panel.

NetIOC is enabled by clicking on "Properties..." on the right side of the panel and then checking "Enable network I/O control on this vDS" in the pop up box.

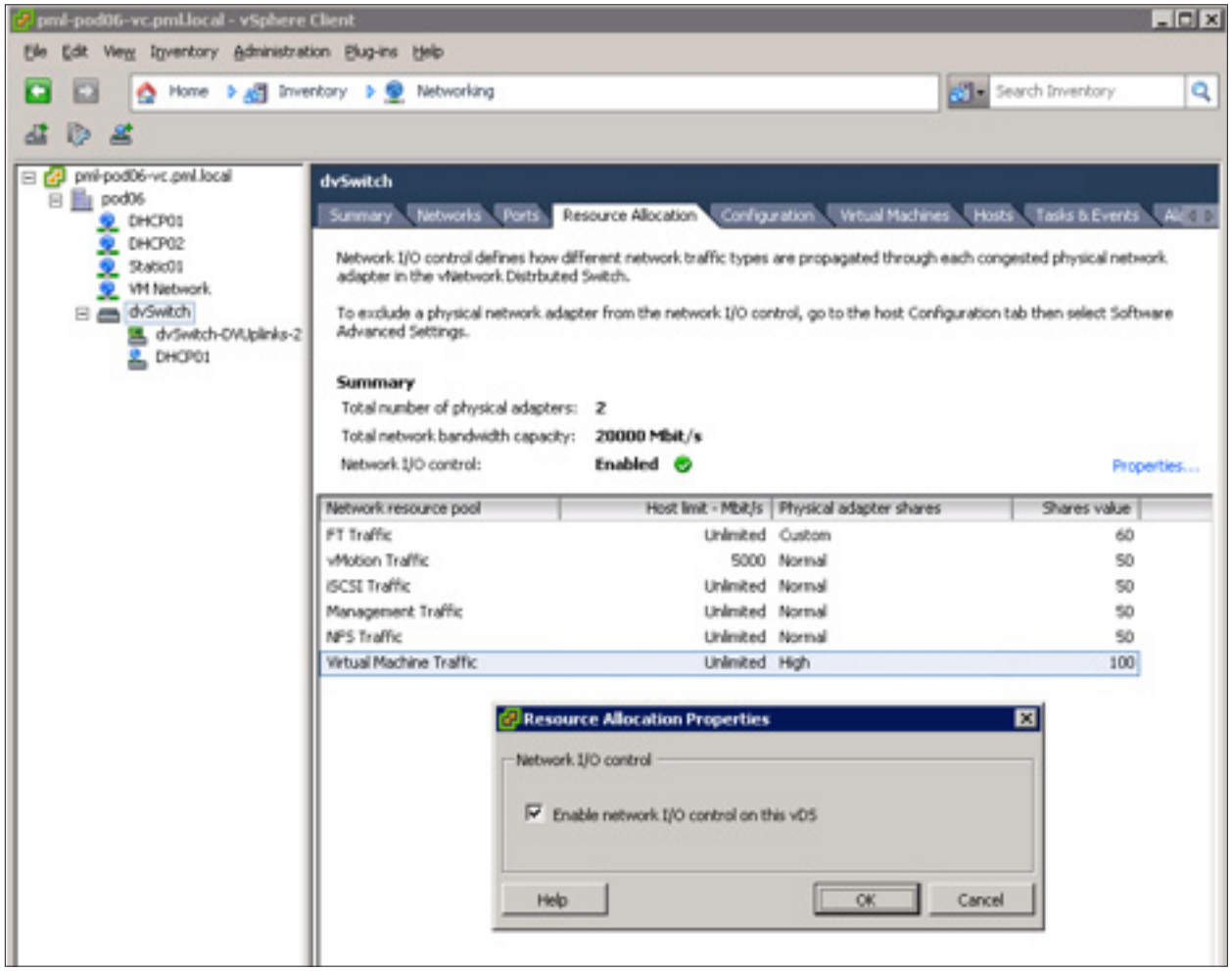

Figure 2. Enabling and Configuring NetIOC from the vSphere Client

The Limits and Shares for each traffic type is configured by right-clicking on the traffic type (for example, Virtual Machine Traffic) and selecting "Edit Settings..." This will bring up a Network Resource Pool Setting dialog box in which you can select the Limits and Shares values for that traffic type.

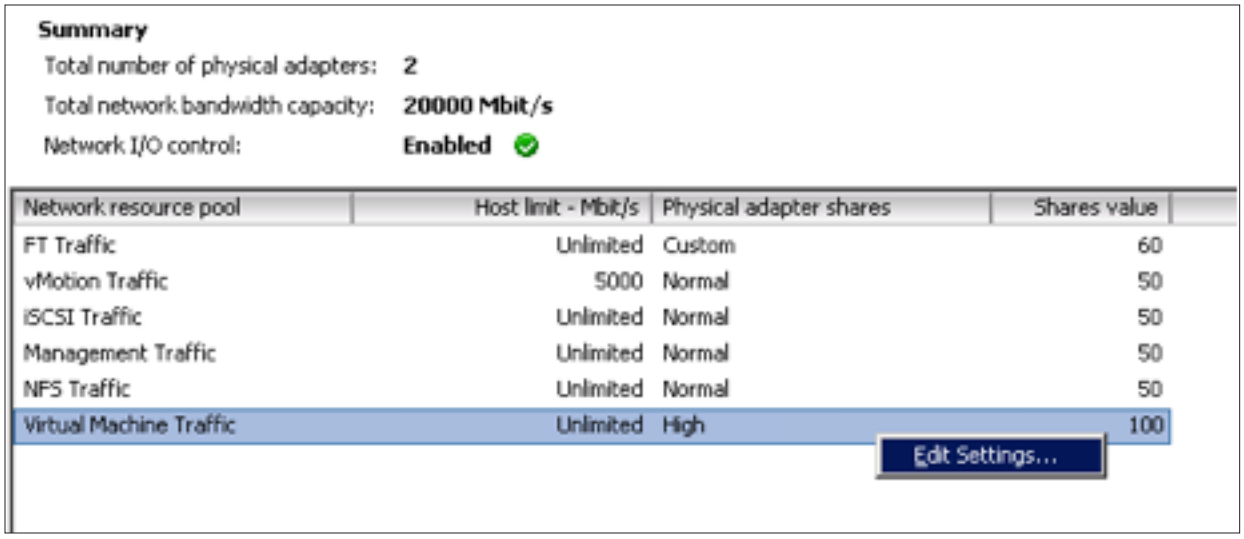

Figure 3. Editing the NetIOC Settings

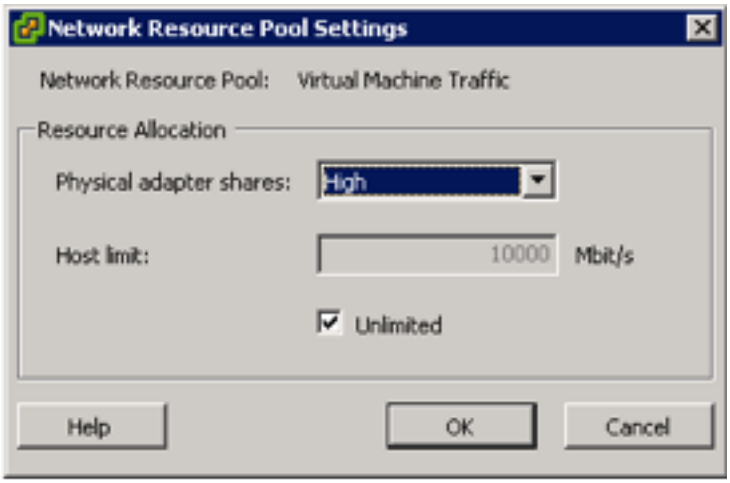

Figure 4. Resource Pool Settings Panel for NetIOC

LBT is configured from the Teaming and Failover panel on each DV Port Group.

Figure 5. Configuring Load-Based Teaming

NetIOC Usage

Unlike the limits that are specified in absolute units of Mbps, shares are used to specify the relative importance of the flows. Shares are specified in abstract units with a value ranging from 1 to 100. In this section, we provide an example that describes the usage of shares.

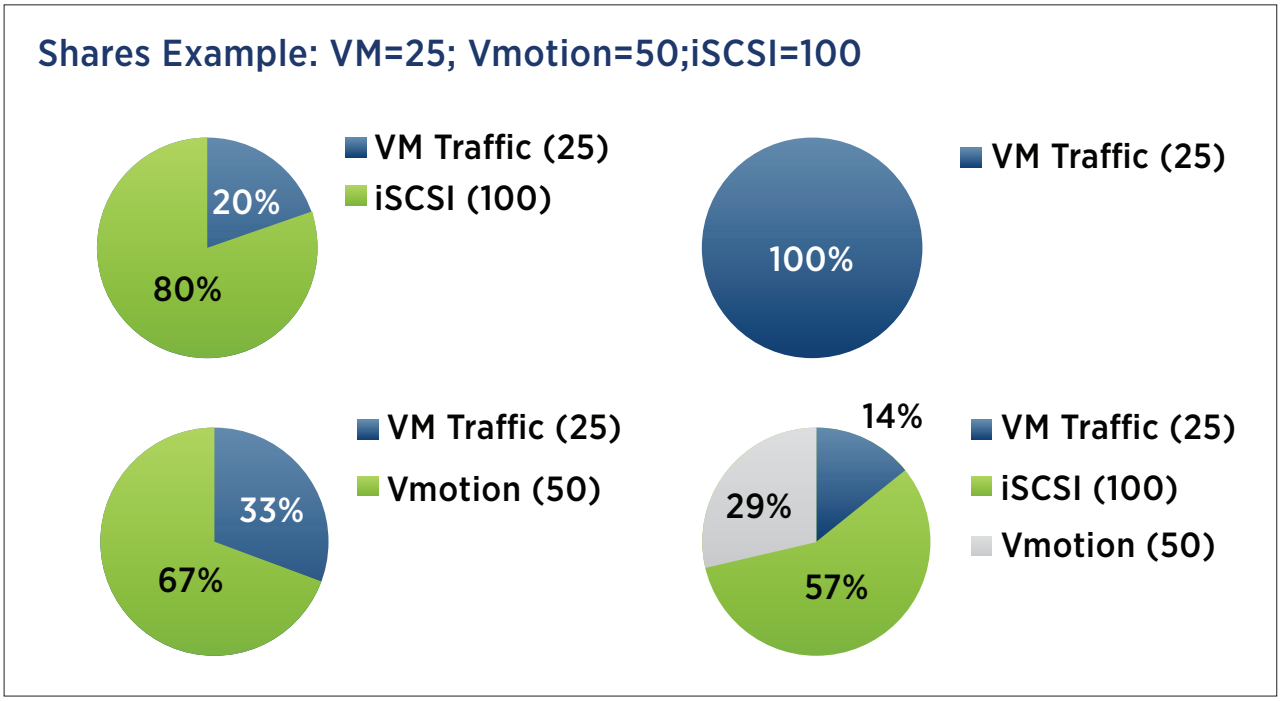

Figure 6. NetIOC shares usage example

Figure 6 highlights the following characteristics of the shares:

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

NetIOC Performance

In this section, we describe in detail the test-bed configuration, the workloads used to generate the network traffic flows, and the test results.

Test Configuration

In our test configuration, we used an ESX cluster that comprised two Dell PowerEdge R610 servers running the GA release of ESX 4.1. Each of the servers was configured with dual-socket, quad-core 2.27 GHz Intel Xeon L5520 processors, 96 GB of RAM, and a 10 GbE Intel Oplin NIC. The following figure depicts the hardware configuration used in our tests. The complete hardware details are provided in Appendix A.

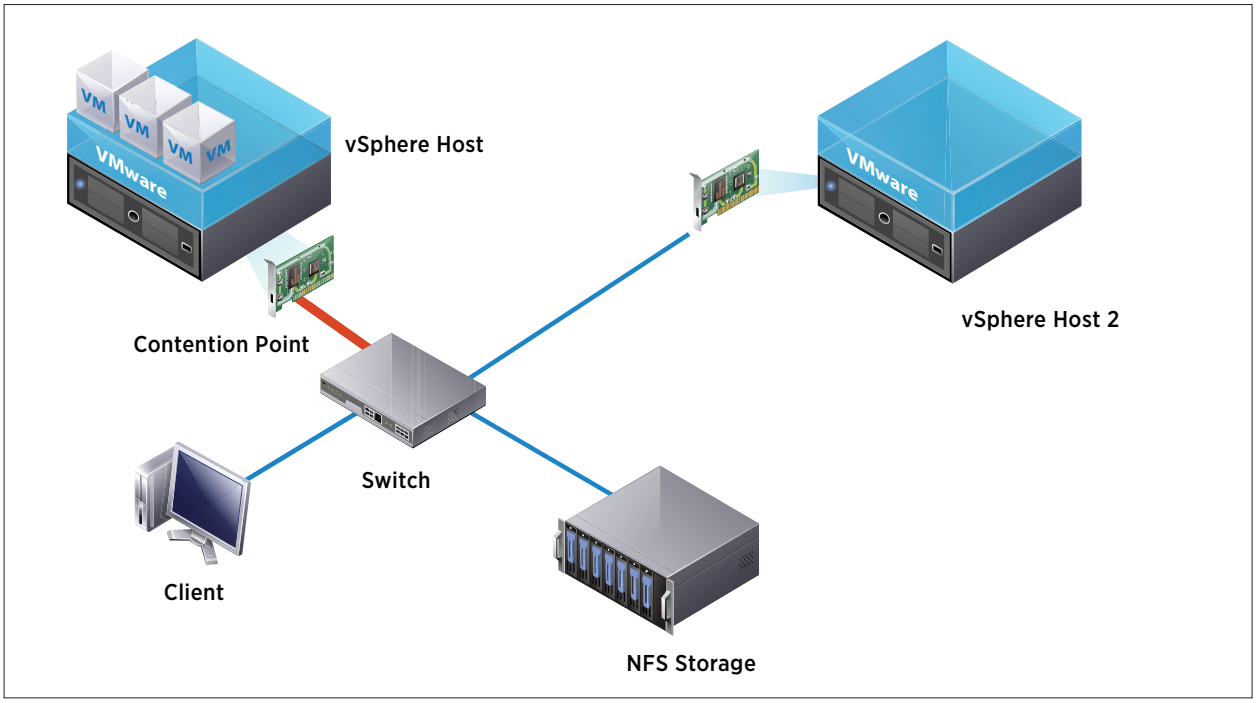

Figure 7. Physical Hardware Setup Used in the Tests

In our test configuration, we used a single vDS that spanned both vSphere hosts. We configured the vDS with a single dvUplink (dvUplink1). The 10GbE physical NIC port on each of two vSphere hosts was mapped to dvUplink1. We configured the vDS with four DV Port Groups as follows:

- dvPortGroup-FT for FT logging traffic
- dvPortGroup-NFS for NFS traffic
- dvPortGroup-VM for virtual machine traffic
- dvPortGroup-vMotion for vMotion traffic

Using four distinct DV Port Groups enabled us to easily track the network bandwidth usage of the different traffic flows. As shown in Figure 8, on both vSphere hosts, the virtual network adapters (vNICs) of all the virtual machines used for virtual machine traffic, and the VMkernel interfaces (vmknics) used for vMotion, NFS, and VMware FT logging were configured to use the same 10GbE physical network adapter through the vDS interface.

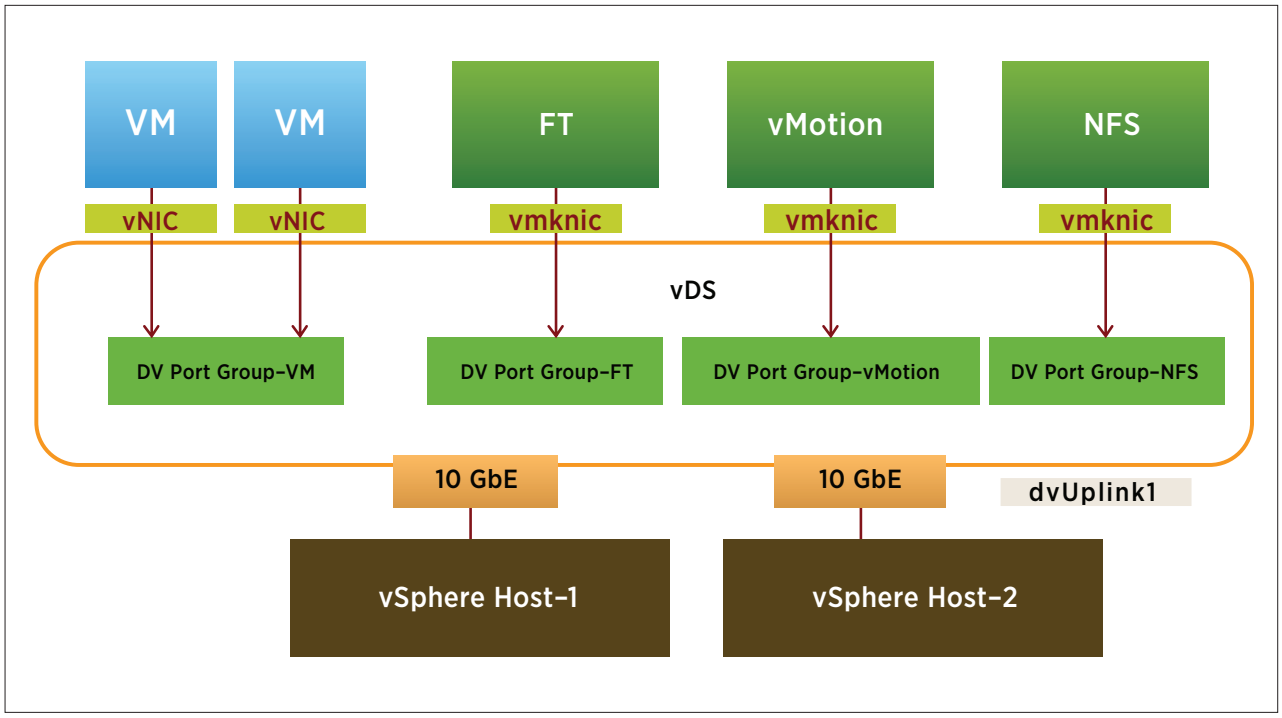

Figure 8. vDS Configuration Used in the Tests

Workloads Used for Performance Testing

To simulate realistic high network I/O load scenarios, we used the industry-standard workloads SPECweb2005 and SPECjbb2005, as they are representative of what most customers would run in their environments.

SPECweb2005 workload: SPECweb2005 is an industry-standard web server workload that is comprised of three component workloads. The support workload emulates a vendor support site that provides downloads — such as driver updates and documentation — over HTTP. It is a highly intensive networking workload. The performance score of the workload is measured in terms of the number of simultaneous user sessions that meet the quality of service requirements specified by the benchmark.

SPECjbb2005 workload: SPECjbb2005 is an industry-standard server-side Java benchmark. It is a highly memory-intensive workload because of Java's usage of the heap and associated garbage collection. Due to these characteristics, when a virtual machine running a SPECjbb2005 workload is subject to vMotion, one could expect to generate heavy vMotion network traffic. This is because during vMotion the entire memory state of the virtual machine is transferred from the source ESX server to a destination ESX server through a high-speed network. During the process of migration, if the memory state of the virtual machine is actively changing, vMotion will need multiple iterations to transfer the active memory state that results in an increase in duration of vMotion and the associated network traffic.

IOmeter workload: IOmeter was used to generate NFS traffic.

NetIOC Performance Test Scenarios

Impact of the (or lack of the) network resource management controls is evident only when aggregate bandwidth requirements of the competing traffic flows exceed the available interface bandwidth. The impact is more apparent when one of the competing traffic flows is latency sensitive. Accordingly, we designed three different test scenarios with a mix of critical and noncritical traffic flows, with the aggregate bandwidth requirements of all the traffic flows under consideration exceeding the capacity of the network interface.

To evaluate and compare the performance and scalability of the virtualized environment with and without NetIOC controls, we used following different scenarios:

- Virtual machine and vMotion traffic flows contending on a vmnic
- NFS, VMware FT, virtual machine, and vMotion traffic flows contending on a vmnic
- Multiple vMotion traffic flows initiated from different vSphere hosts converging onto the same destination vSphere host

The goal was to determine if NetIOC provides good controls in achieving the QoS requirements in SPECweb2005 testing environments that otherwise would not have been met in absence of NetIOC.

Test Scenario 1: Using Two Traffic Flows—Virtual Machine Traffic and vMotion Traffic

We chose latency-sensitive SPECweb2005 traffic and vMotion traffic flows in our first set of tests. The goal was to evaluate the performance of a SPECweb2005 workload in a virtualized environment with and without NetIOC when latency-sensitive SPECweb2005 traffic and vMotion traffic contended for the same physical network resources.

As shown in Figure 9, our test-bed was configured such that both the traffic flows used the same 10GbE physical network adapter. This was done by mapping the virtual network adapters of the virtual machines (used for SPECweb2005 traffic) and the VMkernel interface (used for vMotion traffic) to the same 10GbE physical network adapter. The complete experimental setup details for these tests are provided in Appendix B.

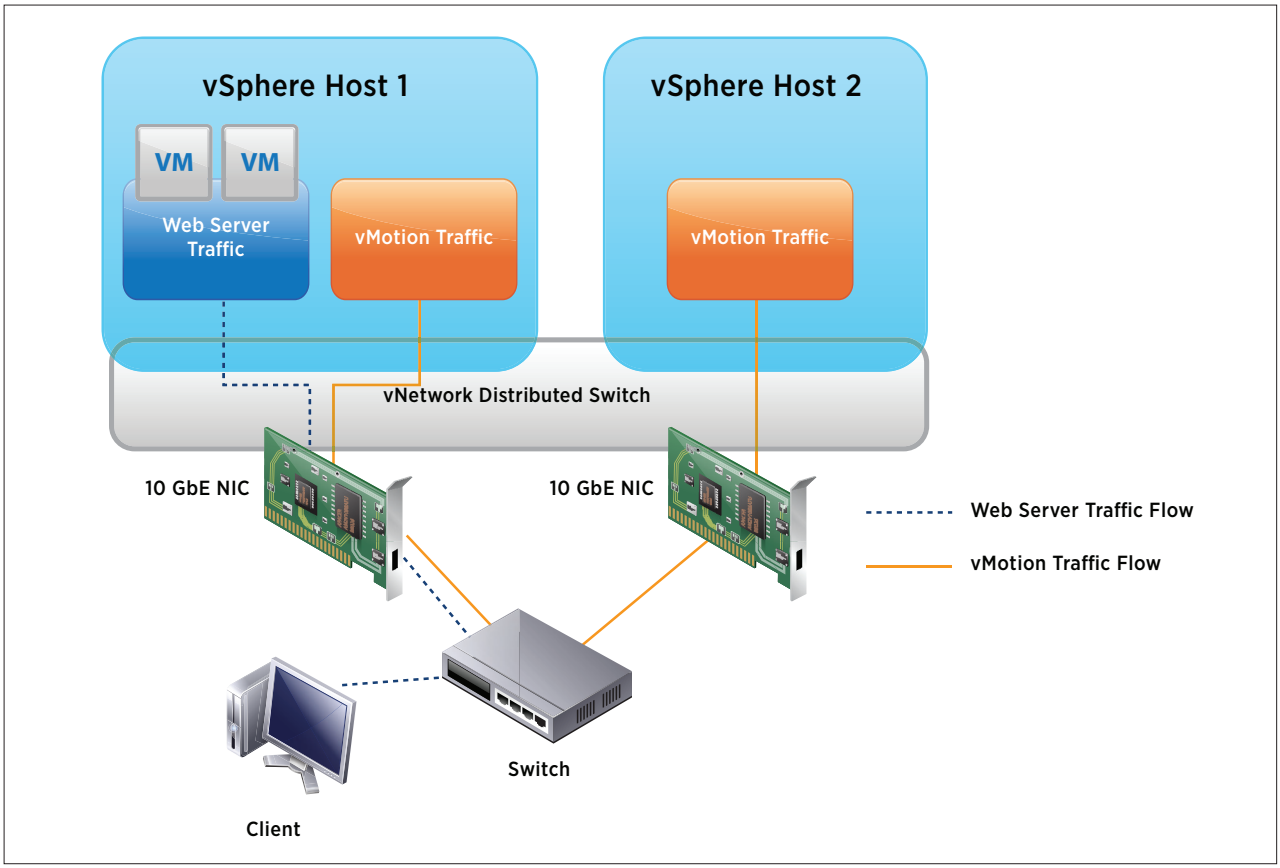

Figure 9. Setup for the Test Scenario 1

At first, we measured the bandwidth requirements of the SPECweb2005 virtual machine traffic and vMotion traffic flows in isolation. The bandwidth usage of the virtual machine traffic while running 17,000 SPECweb2005 user sessions was a little more than 7Gbps during the steady-state interval of the benchmark. The peak network bandwidth usage of the vMotion traffic flow used in our tests was measured to be more than 8Gbps. Thus, if both traffic flows used the same physical resources, the aggregate bandwidth requirements would certainly exceed the 10GbE interface capacity. In the test scenario, during the steady-state period of the SPECweb2005 benchmark, we initiated vMotion traffic flow, which resulted in both the vMotion traffic and the virtual machine traffic flows contending on the same physical 10GbE link.

Figure 10 shows the performance of the SPECweb2005 workload in a virtualized environment without NetIOC. The graph plots the number of SPECweb2005 user sessions that meet the QoS requirements ("Time Good") at a given time. In this graph, the first dip corresponds to the start of the steady-state interval of the SPECweb2005 benchmark when the statistics are cleared. The second dip corresponds to the loss of QoS due to vMotion traffic competing for the same physical network resources.

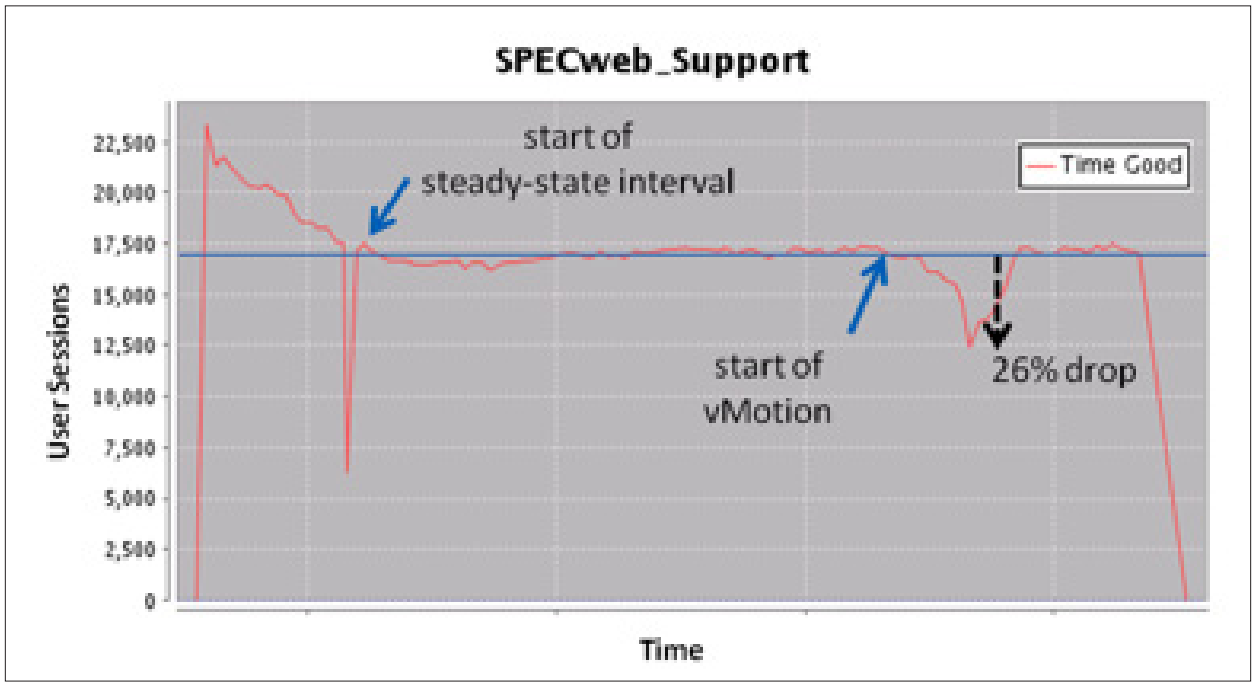

Figure 10. SPECweb2005 Performance without NetIOC

We note that when we repeated the same test scenario several times, the loss of performance shown in the graph varied, possibly due to the nondeterministic nature of vMotion traffic. Nevertheless, these tests clearly demonstrate that lack of any network resource management controls results both in loss of performance and predictability that is required to guarantee SLAs required by critical traffic flows.

Figure 11 shows the performance of a SPECweb2005 workload in a virtualized environment with NetIOC controls in place. We configured the virtual machine traffic with twice the number of shares than those configured for vMotion traffic. In other words, we ensured the virtual machine traffic had twice the priority over vMotion traffic when both the traffic flows competed for the same physical network resources. Our tests revealed that although the duration of the vMotion was doubled due to the controls enforced by NetIOC, as shown in Figure 11, the SPECweb2005 performance was unperturbed due to vMotion traffic.

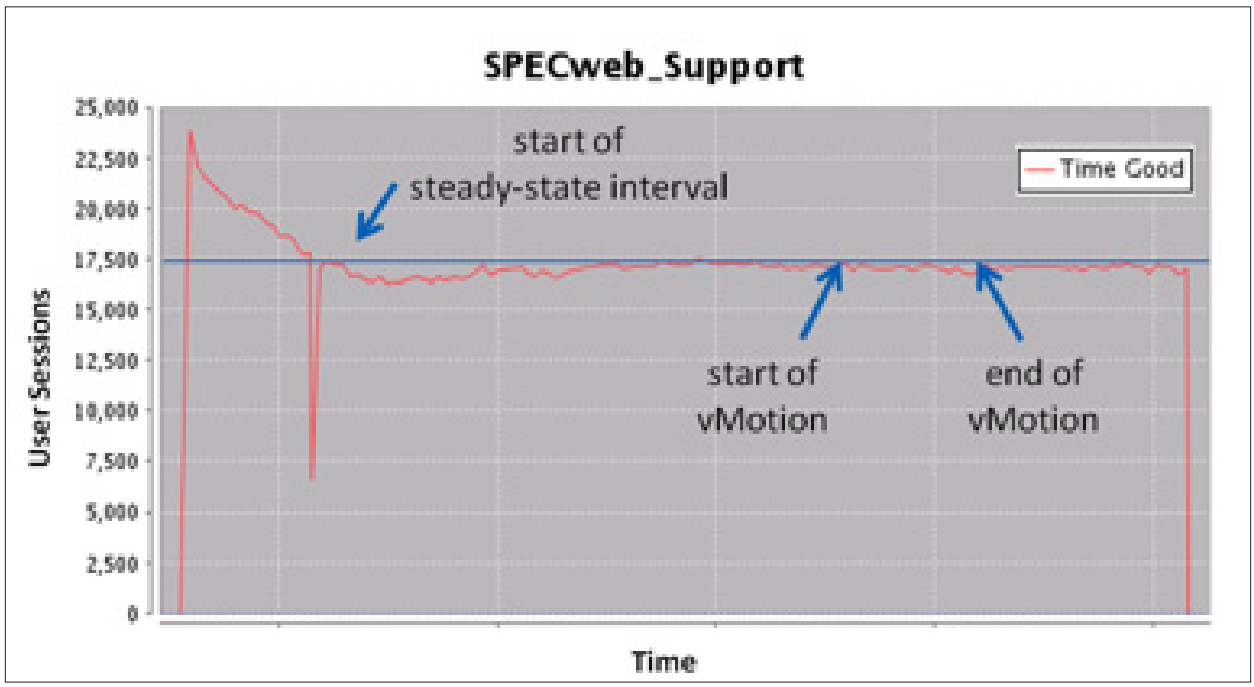

Figure 11. SPECweb2005 Performance with NetIOC

Test Scenario 2: Using Four Traffic Flows — NFS Traffic, Virtual Machine Traffic, VMware FT Traffic and vMotion Traffic

In this test scenario, we chose a very realistic customer deployment scenario that featured fault-tolerant Web servers.

A recent VMware customer survey found Web servers had the distinction of topping the high ranks among the popular applications used in conjunction with the VMware FT feature. This is no coincidence because fault-tolerant Web servers provide some compelling features that are not available with typical Web server-farm deployment scenarios using load balancers that redirect user requests when a Web server goes down. Such load–balancer based solutions may not be the most customer-friendly for Web sites that provide very large downloads, such as driver updates and documentation. As an example, consider a failure of a Web server while a user is downloading a large user manual. In a load-balancer based Web-farm deployment scenario, this will result in user request to fail (or timeout) and the user would need to resubmit the request. On the other hand, in a VMware FT–enabled Web server environment, the user will not experience such failure due to the presence of a secondary hypervisor that has full information on pending I/O operations from the failed primary virtual machine, and commits all the pending I/O. Refer to VMware vSphere 4 Fault Tolerance: Architecture and Performance for more information on VMware FT.

As shown in Figure 12, our test-bed was configured such that all the traffic flows used in the test mix contended for the same network resources. The complete experimental setup details for these tests are provided in Appendix B.

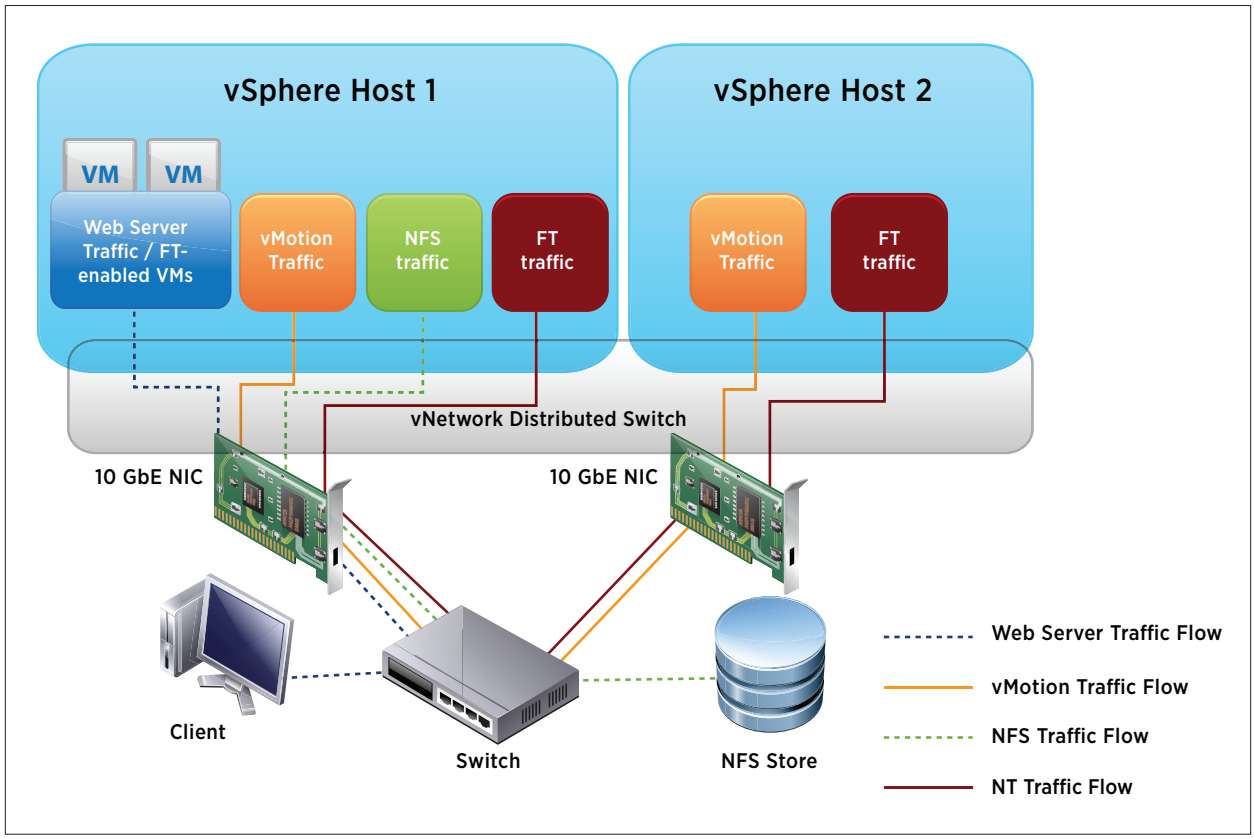

Figure 12. Setup for the Test Scenario 2

Our test-bed configuration featured four virtual machines that included:

- Two VMware FT–enabled Web server virtual machines serving SPECweb2005 benchmark requests (that generated virtual machine traffic and VMware FT logging traffic)
- One virtual machine (VM3) accessing an NFS store (that generated NFS traffic)
- One virtual machine (VM4) running a SPECjbb2005 workload (used to generate vMotion traffic)

At first we measured the network bandwidth usage of all the four traffic flows in isolation. Table 2 describes the network bandwidth usage.

NETWORK TRAFFIC	NETWORK BANDWITH USAGE
Virtual Machine Traffic	3Gbps
NFS Traffic	850Mbps
FT Traffic	900Mbps
vMotion Traffic	8.5Gbps (peak usage)

Table 2. Network Bandwidth Usage of the Four Traffic Flows used in the Test Environment

The goal was to evaluate the latencies of critical traffic flows including VMware FT and NFS traffic in a virtualized environment with and without NetIOC controls when four traffic flows contended for the same physical network resources. The test scenario had three phases:

Phase 1: The SPECweb2005 workload in the two VMware FT–enabled virtual machines was in the steady state.

Phase 2: The NFS workload in VM3 became active. SPECweb2005 workload in the other two virtual machines continued to be active.

Phase 3: The VM4 running the SPECjbb2005 workload was subject to vMotion while the NFS and SPECweb2005 workloads remained active in the other virtual machines.

The following figures depict the performance of different traffic flows in absence of NetIOC.

Let us first consider the performance of the VMware FT–enabled Web server virtual machines. The graph plots the number of SPECweb2005 user sessions that meet the QoS requirements ("Time Good") at a given time. In this graph, the first dip corresponds to the start of the steady-state interval of the SPECweb2005 benchmark when the statistics are cleared. The second dip corresponds to the loss of QoS due to multiple traffic flows competing for the same physical network resources. The number of SPECweb2005 users sessions that meet the QoS requirements dropped by about 67 percent during the period of contention. We note that the SPECweb2005 performance degradation in the VMware FT environment was much more severe in the absence of NetIOC than what we observed in the first test scenario. This is because in a VMware FT environment, the primary and secondary virtual machines run in vLockstep, and so the network link between the primary and secondary ESX hosts plays a critical role in performance. During the periods of heavy contention on the network link, the primary virtual machine will make little or no forward progress.

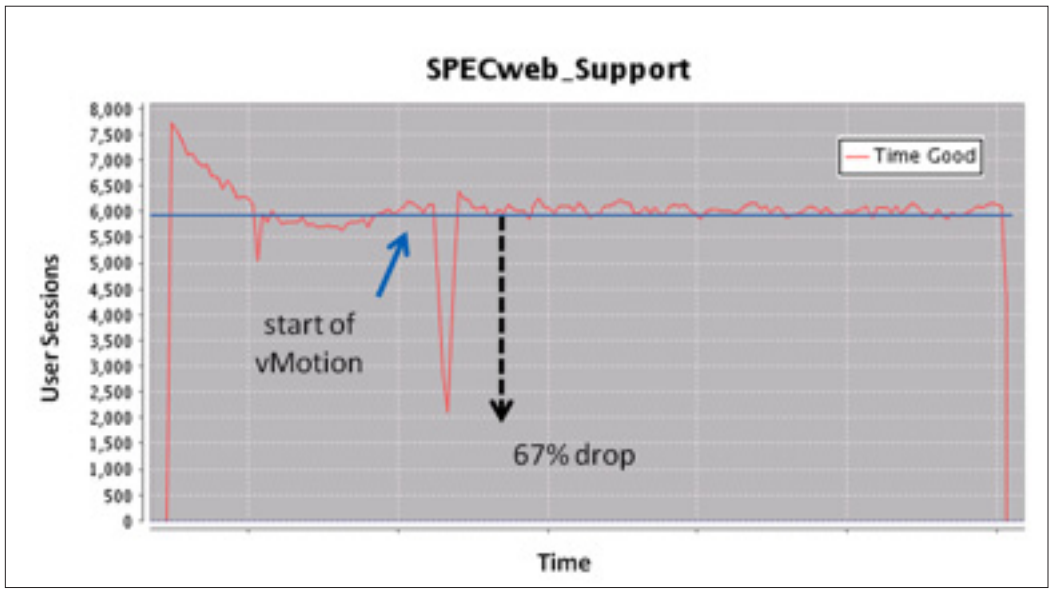

Figure 13. SPECweb2005 Performance in a VMware FT Environment without NetIOC

Similarly, we noticed a significant jump in the NFS store access latencies. As shown in Figure 14, the maximum I/O latency reported by the IOmeter increased from a mere 162 ms to 2166 ms (a factor of 13).

Figure 14. NFS Access Latency without NetIOC

Figure 15 shows the network bandwidth usage of all the competing traffic flows during different phases.

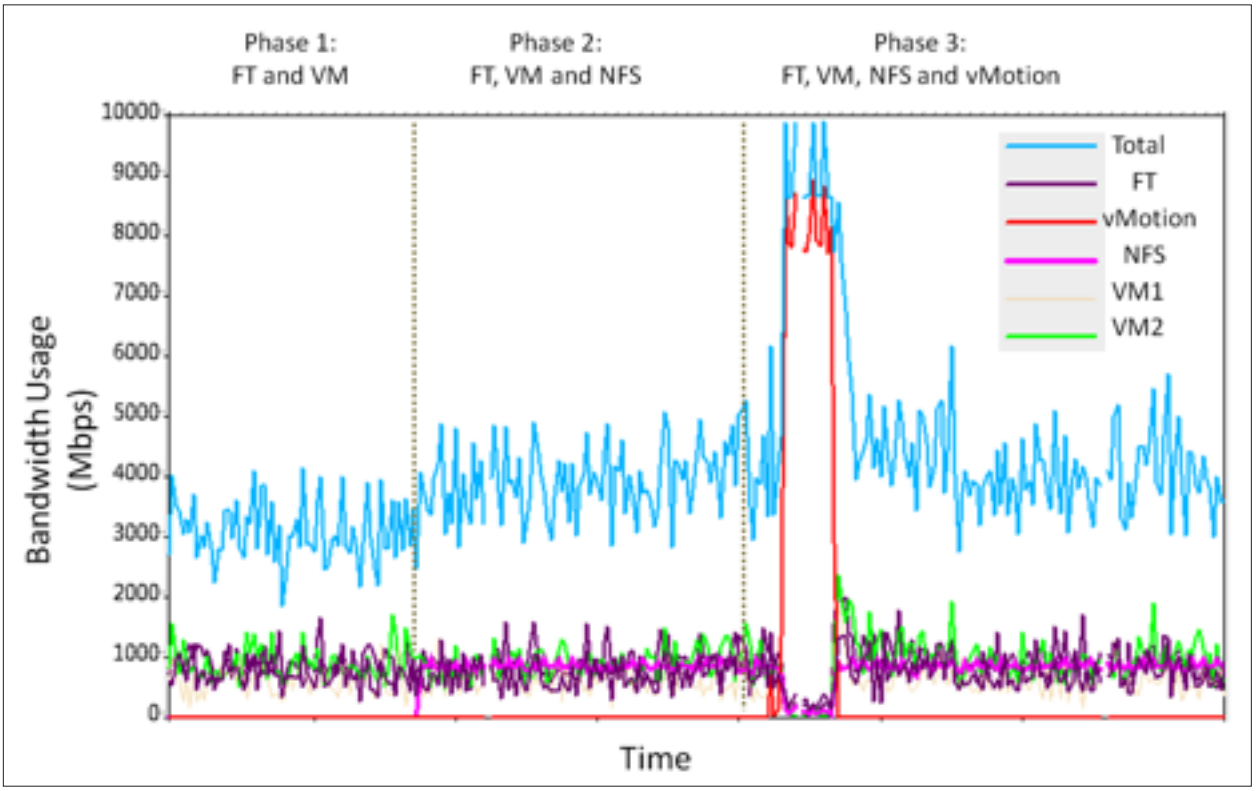

Figure 15. Network Bandwidth Usage of Traffic Flows in Different Phases without NetIOC

A detailed explanation of the bandwidth usage in each phase follows:

Phase 1: In this phase, the VMware FT–enabled VM1 and VM2 were active and the SPECweb2005 benchmark was in a steady-state interval. The aggregate network bandwidth usage of the virtual machine traffic flow and the VMware FT logging traffic flows was less than 4Gbps.

Phase 2: At the beginning of this phase, VM3 became active and added NFS traffic flow to the test mix. This resulted in three traffic flows competing for the network resources. Even so there was no difference in the QoS, as the aggregate bandwidth usage was still less than 5Gbps.

Phase 3: An addition of vMotion traffic flow to the test mix resulted in the aggregate bandwidth requirements of the four traffic flows exceeding the capacity of the physical 10GbE link. Lack of any control mechanism to manage access to the 10GbE bandwidth resulted in vSphere sharing the bandwidth among all the traffic flows. Critical traffic flows including VMware FT and NFS traffic flows got the same treatment as the vMotion traffic flow, which resulted in a significant drop in performance.

The performance requirements of the different traffic flows must be considered to put network I/O resource controls in place. In general, the bandwidth requirement of the VMware FT logging traffic is expected to be much smaller than the requirements of the other traffic flows. However, given its impact on performance, we configured VMware FT logging traffic with the highest priority over other traffic flows. We also ensured NFS traffic and virtual machine traffic flows had higher priority over vMotion traffic. Figure 16 shows shares assigned to the different traffic flows.

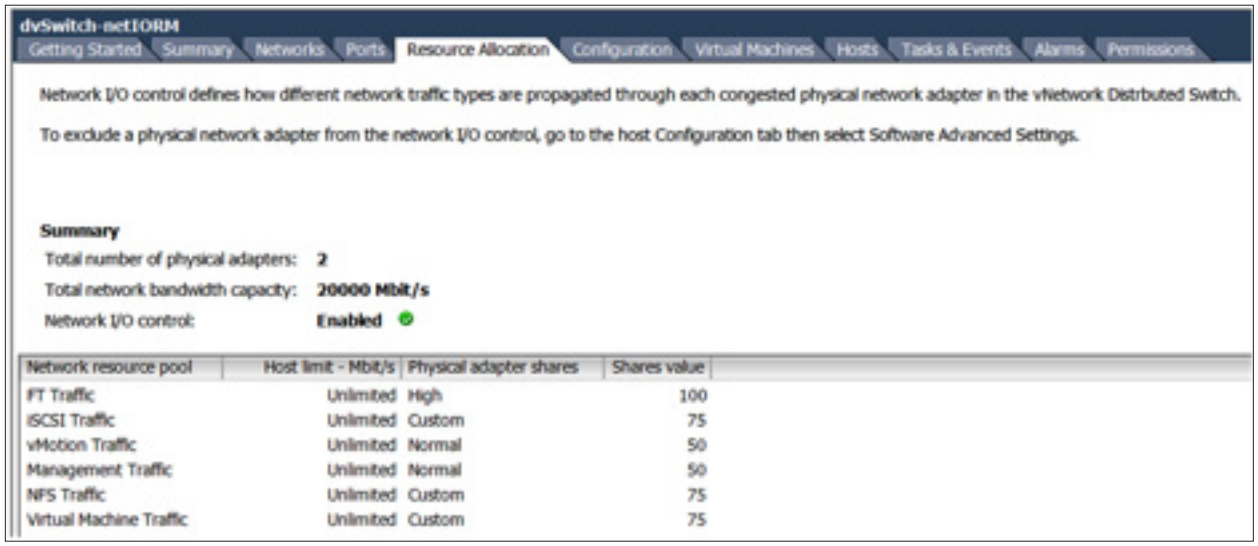

Figure 16. Share Allocation to Different Traffic Flows with NetIOC

Figure 17 shows the network bandwidth usage of the different traffic flows in different phases. As shown in the figure, thanks to the network I/O resource controls, vSphere was able to enforce priority among the traffic flows, and so the bandwidth usage of the critical traffic flows remained unperturbed during the period of contention.

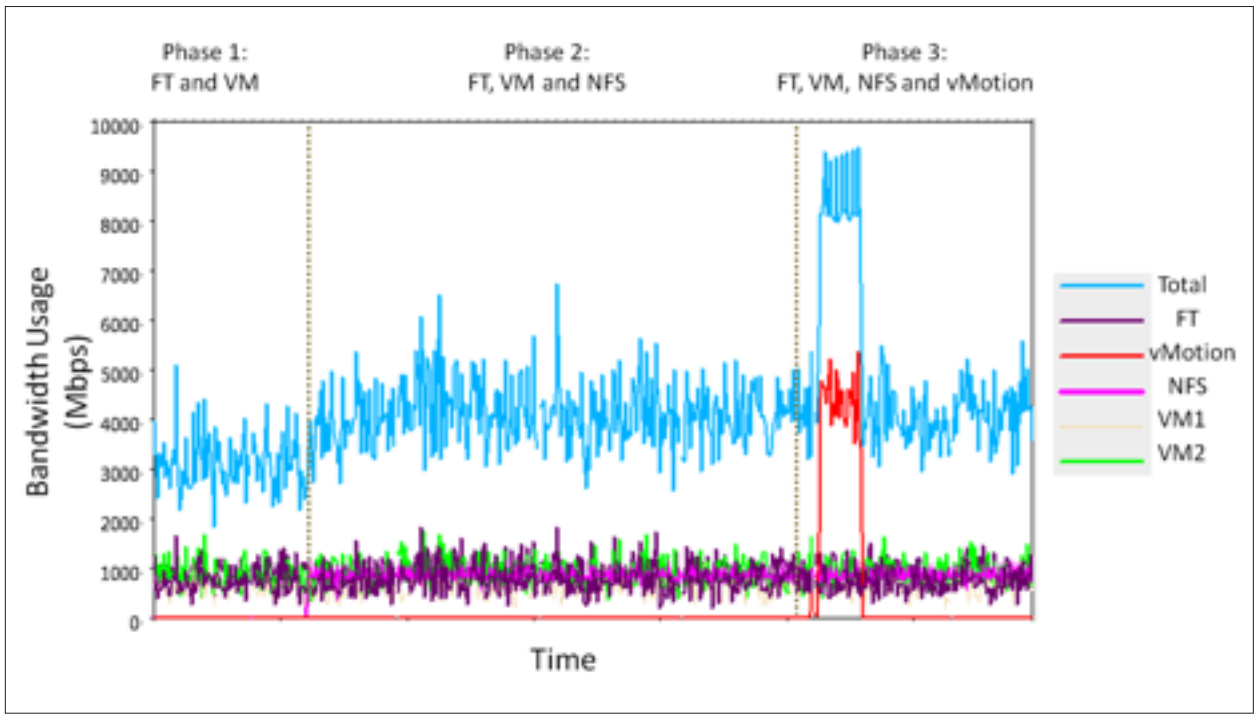

Figure 17. Network Bandwidth Usage of Traffic Flows in Different Phases with NetIOC

The following figures show the performance of SPECweb2005 and NFS workloads in a VMware FT–enabled virtualized environment with NetIOC in place. As shown in the figures, vSphere was able to ensure service level guarantees to both the workloads in all the phases.

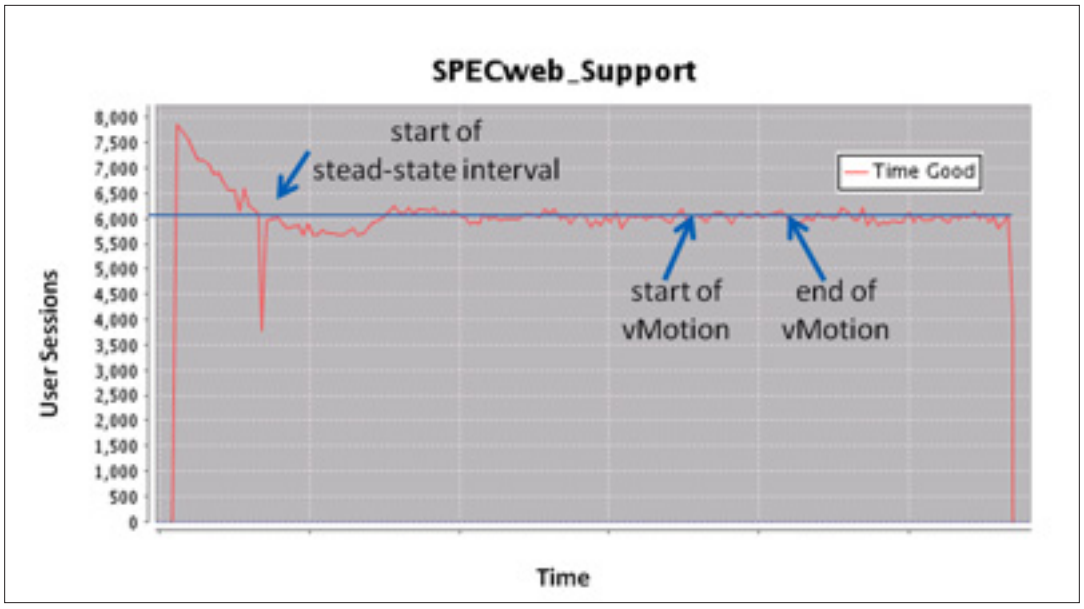

Figure 18. SPECweb2005 Performance in FT Environment with NetIOC

Figure 19. NFS Access Latency with NetIOC

The maximum I/O latency reported by the IOmeter remained unchanged at 162 ms in all the phases, and the SPECweb2005 performance remained unaffected by the network bandwidth usage spike caused by the vMotion traffic flow.

Test Scenario 3: Using Multiple vMotion Traffic Flows

In this final test scenario, we will show how NetIOC can be used in combination with Traffic Shaper to provide a comprehensive network convergence solution in a virtualized datacenter environment.

While NetIOC enables you to limit vMotion traffic initiated from a vSphere host, it fails to prevent performance loss when multiple vMotion traffic flows initiated on different vSphere hosts converge onto a single vSphere host and possibly overwhelm the latter. We will show how a solution based on NetIOC and Traffic Shaper can prevent such an unlikely event.

In vSphere 4.0, support for traffic shaping was introduced, providing some rudimentary controls on network bandwidth usage. For instance, it only provided bandwidth usage controls at the port level, and did not enforce prioritization among traffic flows. These controls were provided for both egress and ingress traffic. In vSphere deployment, the egress and ingress traffic are with respect to a vDS (or vSS). The traffic going into a vDS is ingress/input, and traffic leaving a vDS is egress/output. So, from the perspective of a vNIC port (or vmknic port), the network traffic from the physical network (or vmnic) will ingress into the vDS and egress from vDS to vNIC. Similarly, the traffic flow from vNIC will ingress into the vDS and egress to the physical network (or vmnic). In other words, the ingress and egress need to be interpreted as follows:

Ingress traffic: traffic from a vNIC (or vmknic) to vDS

Egress traffic: traffic from vDS to the vNIC (or vmknic)

In this final test scenario, we added a third vSphere host to the same cluster that we used in our previous tests. As shown in Figure 20, the cluster used for this test comprised three vSphere hosts.

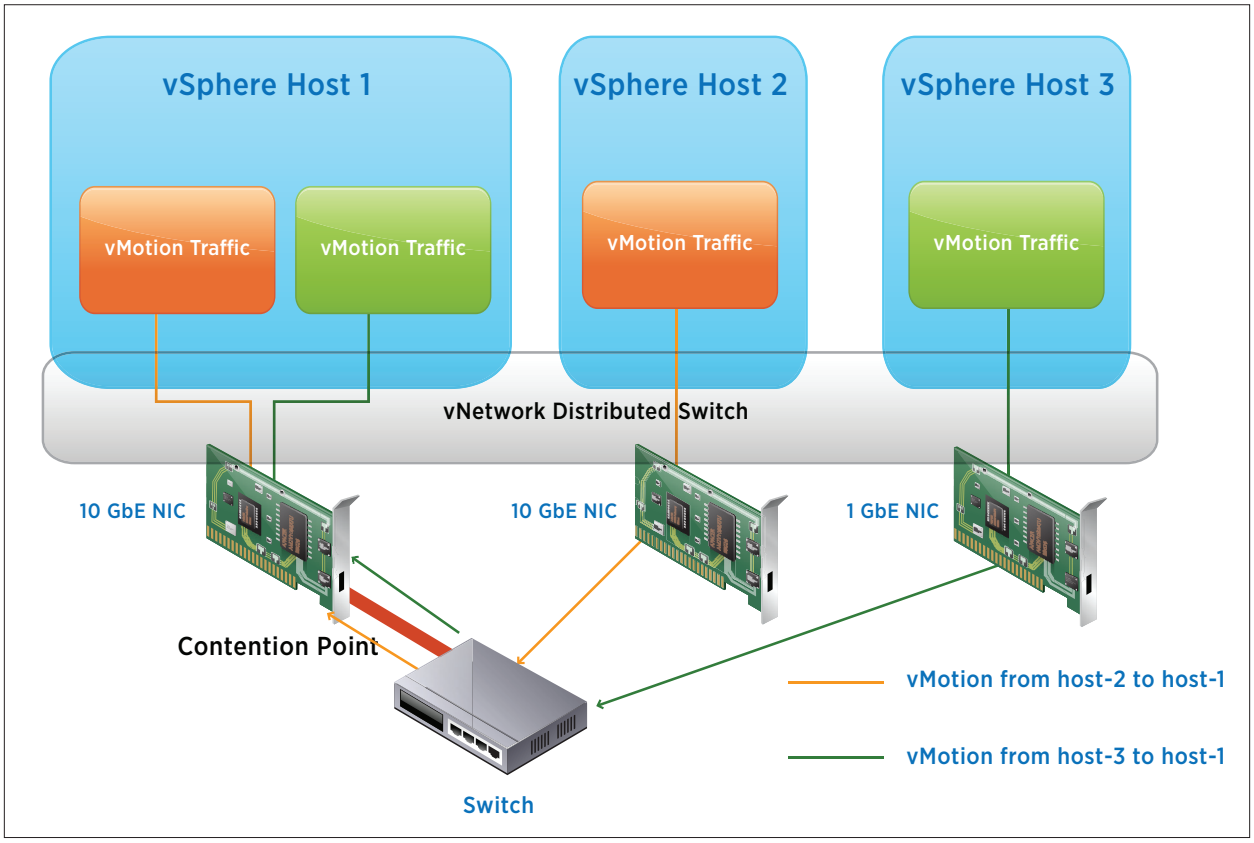

Figure 20.

We initiated vMotion traffic (peak network bandwidth usage of 9Gbps) from vSphere Host 2, and vMotion traffic (peak network bandwidth usage close to 1Gbps) from vSphere Host 3. Both of these traffic flows converged onto the same destination vSphere host (Host 1). Below, we describe the results of the three test configurations.

Without NetIOC

As a point of reference, we first disabled NetIOC in our test configuration. Our tests indicated that, without any controls, the receive link on Host 1 was fully saturated due to multiple vMotion traffic flows whose aggregate network bandwidth usage exceeded the link capacity.

With NetIOC

As shown in Figure 21, we used NetIOC to enforce limits on vMotion traffic.

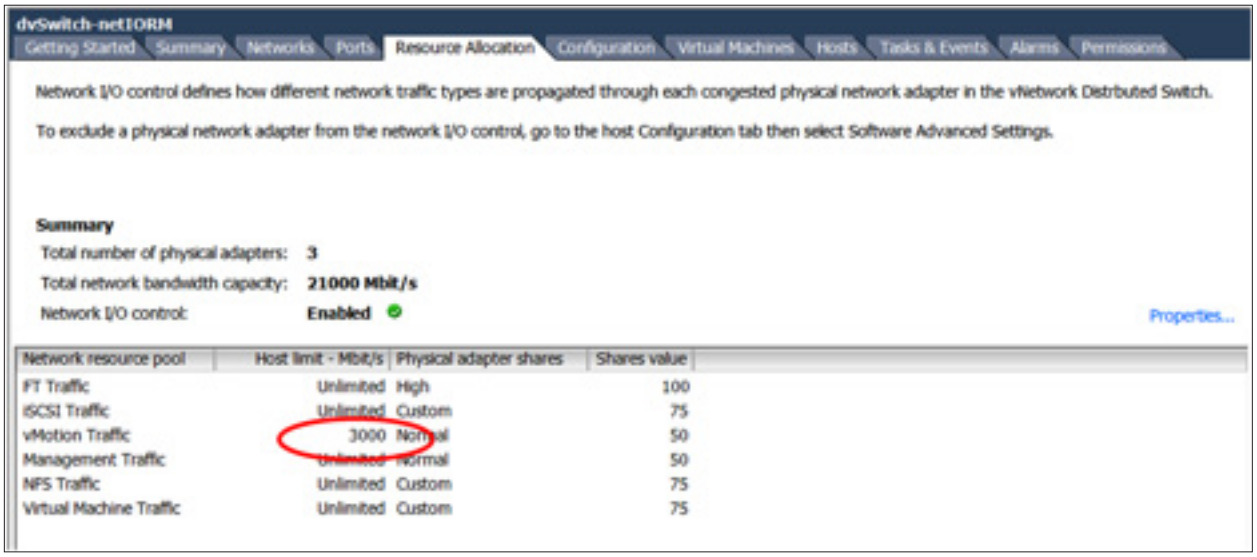

Figure 21. NetIOC Settings to Enforce Limits on vMotion Traffic Flow

Figure 22 shows the Rx network bandwidth usage on Host 1 (with NetIOC controls in place) as multiple vMotion traffic flows converge on it.

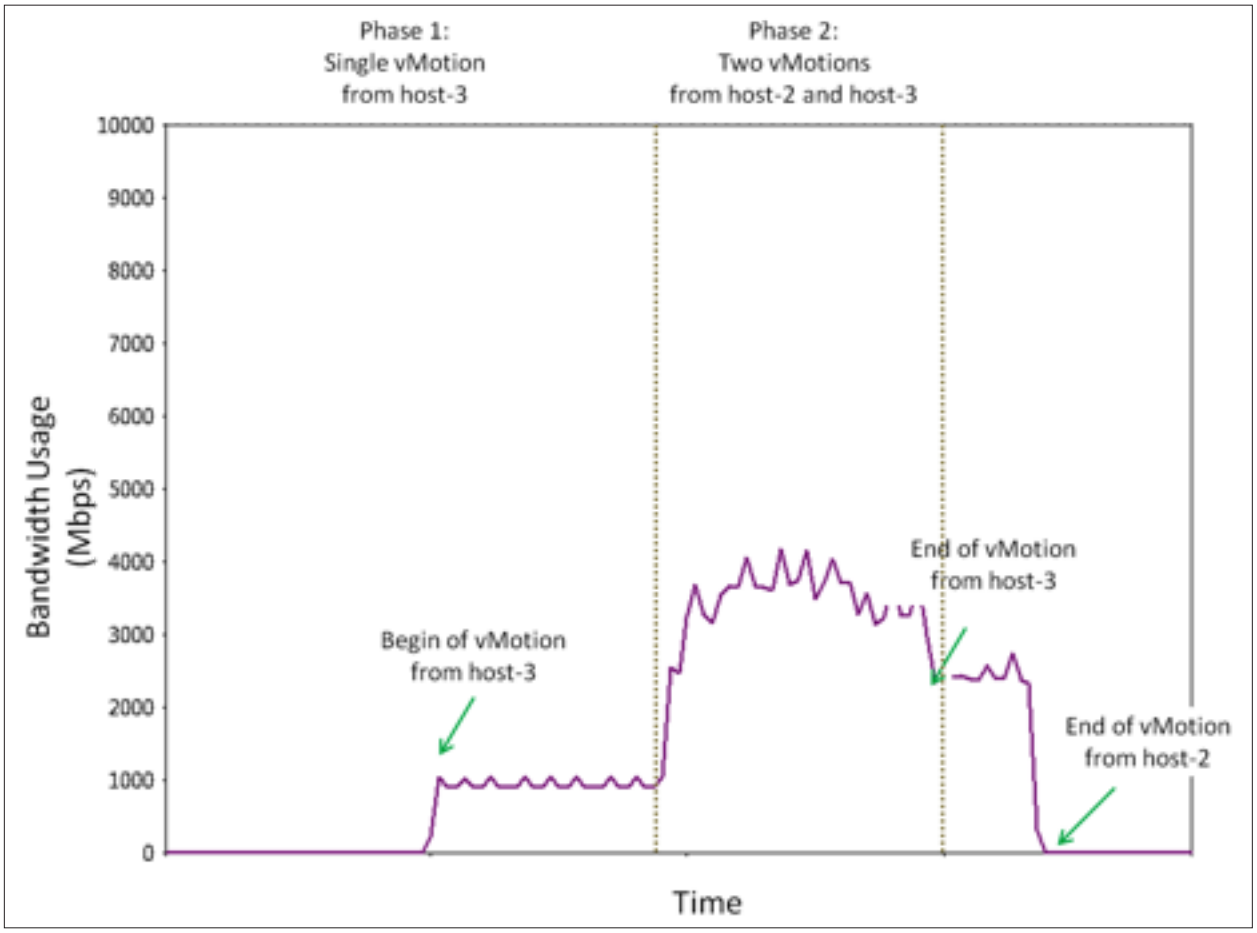

Figure 22. Rx Network Bandwidth Usage on Host 1 with Multiple vMotions (with NetIOC On)

A detailed explanation of the bandwidth usage in each phase follows:

Phase 1: In this phase, vMotion from Host 3 to Host 1 was active. Due to the 1GbE link capacity on Host 3, the bandwidth usage of the vMotion traffic flow was limited to 1Gbps.

Phase 2: At the beginning of this phase, vMotion from Host 2 to Host 1 became active, resulting in two active vMotion traffic flows converging onto the same destination vSphere host. Thanks to the NetIOC controls, the vMotion traffic flow from Host 2 was only limited to 3Gbps. The aggregate network bandwidth usage of both the active vMotion flows was close to 4Gbps.

NOTE: If there had been more concurrent vMotions (even if such an event is very unlikely), NetIOC would have failed to prevent these vMotions from saturating the receive link on the Host 1.

With NetIOC and Traffic Shaper

With NetIOC controls in place, we also used Traffic Shaper to enforce limits on the egress traffic. NetIOC controls obviate the need for traffic-shaping policies on ingress traffic. Accordingly, as shown in Figure 23, we used Traffic Shaper to enforce policies only on egress traffic. Also note that, each of the DV Port Groups can have its own traffic-shaping policy. In our example, we configured the dvPortGroup-vMotion with the traffic-shaping policies shown in Figure 21.

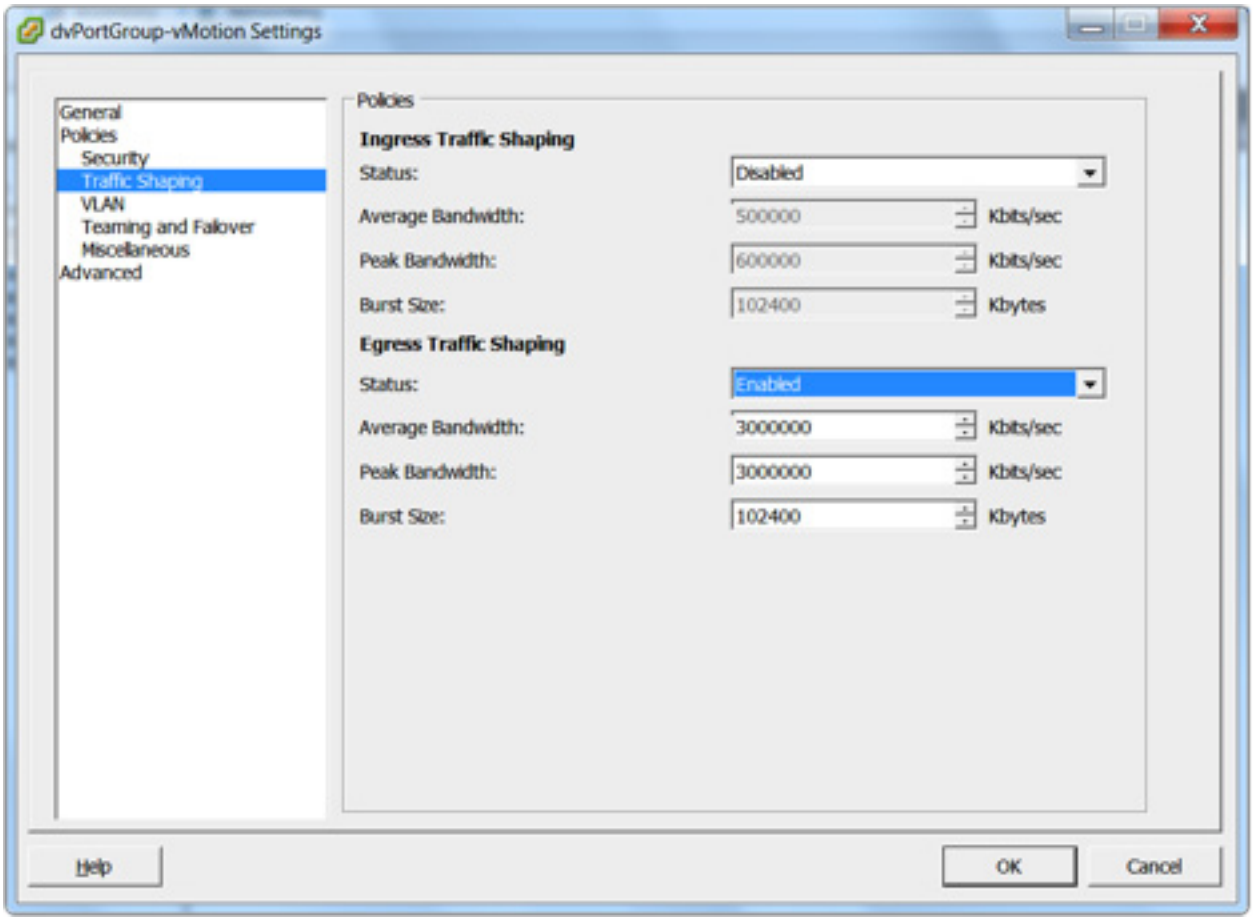

Figure 23. Traffic Shaping Policies on Egress Traffic

Figure 24 shows the Rx network bandwidth usage on Host 1 (with both NetIOC and Traffic Shaper controls in place) as multiple vMotion traffic flows converge on it.

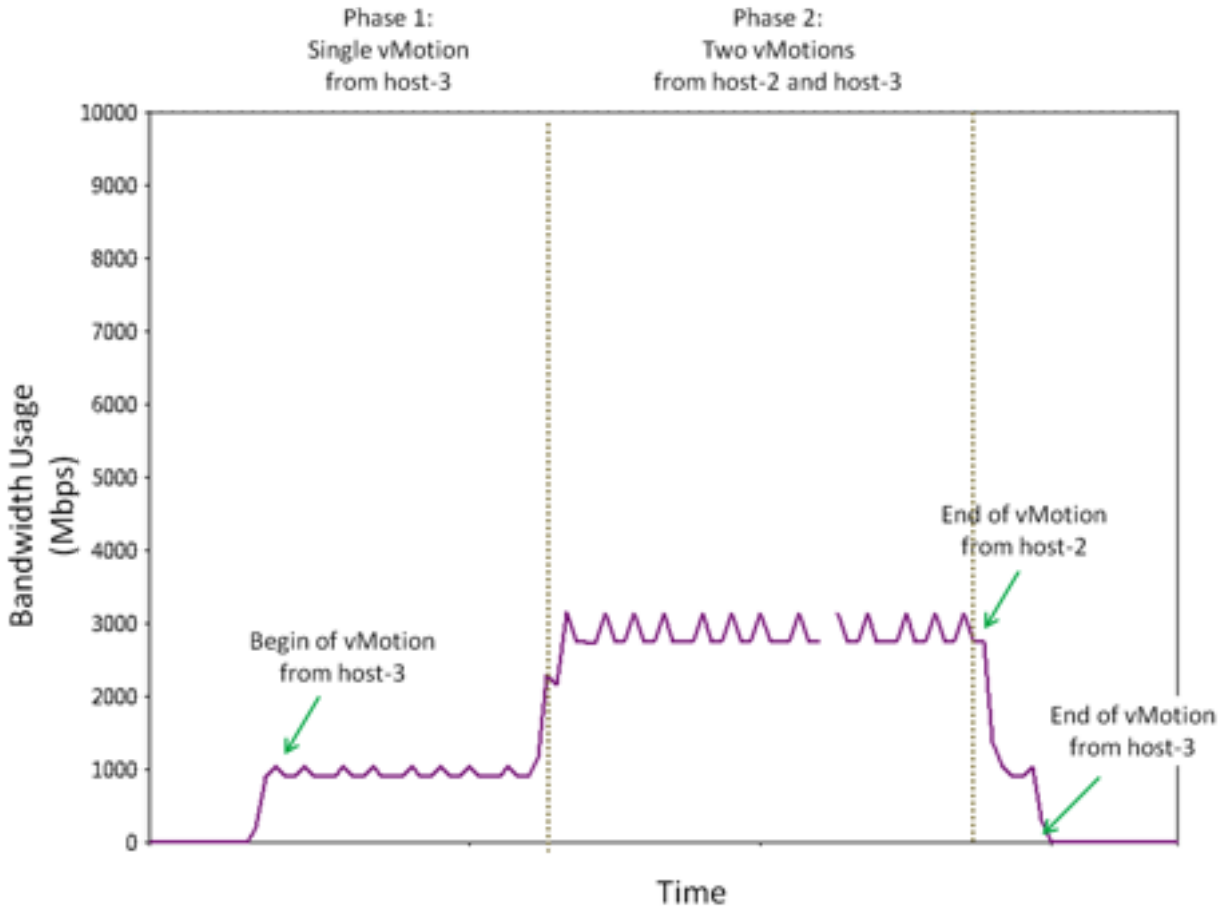

Figure 24. Rx Network Bandwidth Usage on Host 1 with Multiple vMotions (with NetIOC and TS On)

A detailed explanation of the bandwidth usage in each phase follows:

Phase 1: In this phase, vMotion from Host 3 to Host 1 was active. Due to the 1GbE link capacity on Host 3, the bandwidth usage of the vMotion traffic flow was limited to 1Gbps.

Phase 2: At the beginning of this phase, vMotion from Host 2 to Host 1 became active, resulting in two active vMotion traffic flows converging onto the same destination vSphere host. With both NetIOC and Traffic Shaper controls in place, the aggregate bandwidth usage on the receiver never exceeded 3Gbps.

These tests confirm that NetIOC in combination with Traffic Shaper can be a viable solution that provides effective controls on both receive and transmit traffic flows in a virtualized datacenter environment.

NetIOC Best Practices

NetIOC is a very powerful feature that will make your vSphere deployment even more suitable for your I/O-consolidated datacenter. However, follow these best practices to optimize the usage of this feature:

Best practice 1: When using bandwidth allocation, use "shares" instead of "limits," as the former has greater flexibility for unused capacity redistribution. Partitioning the available network bandwidth among different types of network traffic flows using limits has shortcomings. For instance, allocating 2Gbps bandwidth by using a limit for the virtual machine resource pool provides a maximum of 2Gbps bandwidth for all the virtual machine traffic even if the team is not saturated. In other words, limits impose hard limits on the amount of the bandwidth usage by a traffic flow even when there is network bandwidth available.

Best practice 2: If you are concerned about physical switch and/or physical network capacity, consider imposing limits on a given resource pool. For instance, you might want to put a limit on vMotion traffic flow to help in situations where multiple vMotion traffic flows initiated on different ESX hosts at the same time could possibly oversubscribe the physical network. By limiting the vMotion traffic bandwidth usage at the ESX host level, we can prevent the possibility of jeopardizing performance for other flows going through the same points of contention.

Best practice 3: Fault tolerance is a latency-sensitive traffic flow, so it is recommended to always set the corresponding resource-pool shares to a reasonably high relative value in the case of custom shares. However, in the case where you are using the predefined default shares value for VMware FT, leaving it set to high is recommended.

Best practice 4: We recommend that you use LBT as your vDS teaming policy while using NetIOC in order to maximize the networking capacity utilization.

NOTE: As LBT moves flows among uplinks it may occasionally cause reordering of packets at the receiver.

Best practice 5: Use the DV Port Group and Traffic Shaper features offered by the vDS to maximum effect when configuring the vDS. Configure each of the traffic flow types with a dedicated DV Port Group. Use DV Port Groups as a means to apply configuration policies to different traffic flow types, and more important, to provide additional Rx bandwidth controls through the use of Traffic Shaper. For instance, you might want to enable Traffic Shaper for the egress traffic on the DV Port Group used for vMotion. This can help in situations when multiple vMotions initiated on different vSphere hosts converge to the same destination vSphere server.

Conclusions

Consolidating the legacy GbE networks in a virtualized datacenter environment with 10GbE offers many benefits — ease of management, lower capital costs and better utilization of network resources. However, during the peak periods of contention, the lack of control mechanisms to share the network I/O resources among the traffic flows can result in significant performance drop of critical traffic flows. Such performance loss is unpredictable and uncontrollable if the access to the network I/O resources is unmanaged. NetIOC available in vSphere 4.1 provides a mechanism to manage the access to the network I/O resources when multiple traffic flows compete. The experiments conducted in VMware performance labs using industry standard workloads show that:

- Lack of NetIOC can result in unpredictable loss in performance of critical traffic flows during periods of contention.
- NetIOC can effectively provide service level guarantees to the critical traffic flows. Our test results showed that NetIOC eliminated a performance drop of as much as 67 percent observed in an unmanaged scenario.
- NetIOC in combination with Traffic Shaper provides a comprehensive network convergence solution enabling features that are not available with the any of the hardware solutions in the market today.

Appendix A: Hardware Setup

vSphere Host

- Number of hosts: 2
- System: Dell PowerEdge R610 servers
- Processor: Intel Xeon L5520 processors @ 2.27 GHz
- Cores: 8 cores, 2 chips, 4 cores/chip (Hyper-Threading enabled)
- Memory: 96 GB
- NIC: Intel 10 Gigabit XF SR Server Adapter
- Hypervisor: ESX 4.1

NFS store

- Model: EMC Celerra NS960

Network Switch

- Model: Summit X450a-24t switches
- Slots: 2*10G XFP slots (XGM2-2xf)
- Tranceivers: 4*XFP tranceivers (10G-SR XFP)

Client Machine

- Number of clients: 22
- System: Dell PowerEdge R200
- Processor: Intel Xeon @ 2400 MHz
- Cores: 4
- Memory: 8192 MB SDRAM
- Network Controller: Broadcom NetXtreme BCM5721 Gigabit Ethernet PCI Express
- Operating System: RHEL4 x86_64 (2.6.9-42.ELsmp)
- JVM Version: Java™ SE Runtime Environment (build 1.6.0_01-b06)

Besim Machine

- Number of Simulators: 1
- System: HP ProLiant DL380 G5
- Processor: Intel Xeon @ 2333 MHz
- Cores: 8
- Memory: 32GB
- Network Controller: Intel 82571EB GbE
- Operating System: RedHat Enterprise Linux 5 Update 1 (x86_64)
- Web Server: Rock Web Server v1.4.2
- Server Scripts: ISAPI

Appendix B: Workload Details

Test Scenario 1

SPECweb2005

- Number of virtual machines: 2
- Virtual machine configuration: 3 VCPUs, 16GB RAM, vmxnet3 virtual NIC, LSI Logic virtual SCSI adapter
- OS version: RHEL5.3, x64
- Web server: Rock Web Server v1.4.7 (x86_64), Rock JSP/Servlet Container v1.3.2 (x86_64)
- Benchmark parameters: 17000 SPECweb2005 support sessions

SPECjbb2005

- Virtual machine configuration: 1 VCPU, 32GB RAM
- OS version: RHEL5.3, x64
- Java version: JRockit R27.4.0, Java 1.6.0_22
- Benchmark parameters: 2 warehouses
- JVM parameters: -Xms24GB -Xmx24GB -Xgc:parallel -XXcompactratio8 -XXlargepages -XXaggressive -XXminblocksize16k

Test Scenario 2

SPECweb2005

- Number of virtual machines: 2
- Virtual machine configuration: 1 VCPU, 12GB RAM, vmxnet3 virtual NIC, LSI Logic virtual SCSI adapter
- OS version: RHEL5.3, x64
- Web server: Rock Web Server v1.4.7 (x86_64), Rock JSP/Servlet Container v1.3.2 (x86_64)
- Benchmark parameters: 6000 SPECweb2005 support sessions

SPECjbb2005

- Virtual machine configuration: 1 VCPU, 32GB RAM
- OS version: RHEL5.3, x64
- Java version: JRockit R27.4.0, Java 1.6.0_22
- Benchmark parameters: 2 warehouses
- JVM parameters: -Xms24GB -Xmx24GB -Xgc:parallel -XXcompactratio8 -XXlargepages -XXaggressive -XXminblocksize16k

IOmeter (NFS workload)

- Virtual machine configuration: 1 VCPU, 1GB RAM
- OS version: Windows Server 2003, x64
- Benchmark parameters:
  - Outstanding I/Os: 12
  - Number of workers: 1
  - Access pattern: 32KB, 0% read, 0% random

About the Author

Sreekanth Setty is a staff member of Performance Engineering at VMware. His work focuses on performance-related topics with an emphasis on networking and virtualization. He has published his findings in a number of white papers, and presented them at various technical conferences. He has a Masters degree in Computer Science from the University of Texas, Austin.

Acknowledgements

The author would like to sincerely thank Jean-Pascal Billaud for reviews and contributions to the paper. He also would like extend thanks to a number of reviewers including Boon Seong Ang, Guy Brunsdon and several other colleagues on his team.

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