

Managing Performance Variance of Applications Using Storage I/O Control

VMware vSphere 4.1

Application performance can be impacted when servers contend for I/O resources in a shared storage environment. There is a crucial need for isolating the performance of critical applications from other, less critical workloads by appropriately prioritizing access to shared I/O resources. Storage I/O Control (SIOC), a new feature offered in VMware vSphere 4.1, provides a dynamic control mechanism for managing I/O resources across VMs in a cluster. The experiments conducted in VMware performance labs show that:

- SIOC prioritizes VMs' access to shared I/O resources based on disk shares assigned to them. During the periods of I/O congestion, VMs are allowed to use only a fraction of the shared I/O resources in proportion to their relative priority, which is determined by the disk shares.
- If the VMs do not fully utilize their portion of the allocated I/O resources on a shared datastore, SIOC redistributes the unutilized resources to those VMs that need them in proportion to VMs' disk shares. This results in a fair allocation of storage resources without any loss in their utilization.
- SIOC minimizes the fluctuations in performance of a critical workload during periods of I/O congestion. For the test cases executed at VMware labs, limiting the fluctuations to a small range resulted in as much as a 26% performance benefit compared to that in an unmanaged scenario.

This paper is organized as follows:

- "Introduction" on page 1
- "Terminology" on page 2
- "Experimental Environment" on page 2
- "Test 1: Performance impact of Storage I/O Control" on page 5
- "Test 2: Intelligent prioritization of I/O resources" on page 7
- "Test 3: SIOC in a datacenter" on page 12
- "Conclusion" on page 15

Introduction

Datacenters based on VMware's virtualization products often employ a shared storage infrastructure to service clusters of vSphere hosts. Storage area networks (SANs) expose logical storage devices (LUNs) that can be shared across a cluster of ESX hosts. VMFS datastores are created on these shared logical devices, and virtual disks belonging to virtual machines (VMs) are created on them. Consolidating VMs' virtual disks onto a single VMFS datastore backed by a higher number of disks has several advantages—ease of management, better resource utilization, and higher performance (when storage is not bottlenecked). Active

VMs can take advantage of periods of inactivity in the shared environment to obtain a higher I/O performance compared to what they would have obtained had they used dedicated VMFS datastores backed by fewer disks.

However, there are instances when a higher than expected number of I/O-intensive VMs that share the same storage device become active at the same time. During this period of peak load, VMs contend with each other for storage resources. In such situations, lack of a control mechanism can lead to performance degradation of the VMs running critical workloads as they compete for storage resources with VMs running less critical workloads.

Storage I/O Control (SIOC), a new feature offered in VMware vSphere 4.1, provides a fine-grained storage control mechanism by dynamically allocating portions of hosts' I/O queues to VMs running on the vSphere hosts based on shares assigned to the VMs. **Using SIOC, vSphere administrators can mitigate the performance loss of critical workloads during peak load periods by setting higher I/O priority (by means of disk shares) to those VMs running them. Setting I/O priorities for VMs results in better performance during periods of congestion.**

The advantages of using SIOC when consolidating applications on a vSphere-based virtual infrastructure will be illustrated in this paper. For more information about using SIOC, see "Managing Storage I/O Resources" in the *vSphere 4.1 Resource Management Guide* at <http://pubs.vmware.com>.

Terminology

The following terms are used in this paper:

- **Disk shares** – represent the relative importance of a virtual machine with regard to the distribution of storage I/O resources. Under resource contention, virtual machines with higher share values have greater access to the storage array, which typically results in higher throughput and lower latency.
- **Host aggregate shares** – sum of disk shares of all VMs on a host sharing a datastore.
- **Datastore-wide normalized I/O latency** – weighted average of normalized I/O latency as seen by hosts sharing a datastore. The number of I/O requests completed per second by each host is used as the weight for averaging. Before computing the average, the device latencies are normalized based on the I/O request size. This is done because device latency depends on I/O size (for example, a large 256KB vs. a small 8KB). This way, high device latency due to a large I/O size is distinguished from high device latency due to I/O congestion at the device.
- **Congestion threshold** – threshold for datastore-wide normalized I/O latency at which Storage I/O Control begins to assign priority to the I/O requests of each VM according to its share.

Experimental Environment

The experiments we ran were based on workloads run in 4 VMs (2 hosts with 2 VMs each) and 5 VMs (2 hosts with 2 VMs each and a third host with 1 VM). In all test cases, one of the VMs was identified as a critical workload. In tests 1 and 2, a DVD Store version 2.0 (DS2) workload in VM2 was picked as the critical workload. Note that VM2 shared its host with VM1. In test 3, the DS2 workload in VM5, which runs alone on its host, was identified as critical.

Table 1 describes the vSphere host and system configuration for the experiments.

Table 1. vSphere host and storage system configuration information

Component	Details
Hypervisor	VMware vSphere 4.1
Processors	Two 3GHz Quad Core Intel Xeon 5160 Processors (hosts 1 and 2) Two 3GHz Quad Core Intel Xeon 5365 Processors (host 3)
Memory	8GB (hosts 1 and 2) 32GB (host 3)
Guest Operating System	Windows Server 2008 Enterprise x64 edition, SP1
Database	SQL Server 2008 Enterprise x64 edition
Virtual CPUs/Memory	VMs 1-4 each had 2 vCPUs with 4GB memory VM5 had 4 vCPUs and 8GB memory (32GB total)
Virtual Disk Size (OS)	25GB
Fibre Channel HBA	QLogic QLA2432
File System	VMFS for data and index files of DS2, RDM for log files of DS2
Storage Server/Array	EMC CLARiiON CX3-40 for Fibre Channel
Test Application	DVD Store Version 2

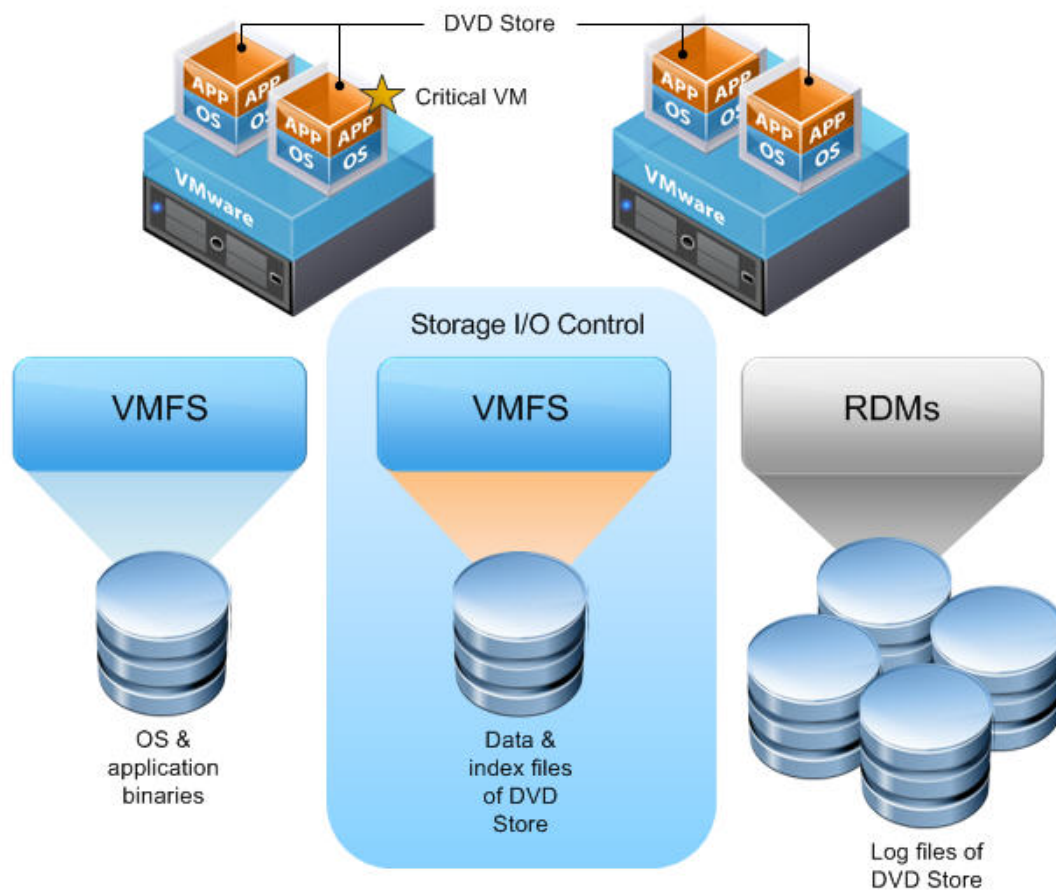
Setup for Test 1 with 4 VMs

Four identical VMs running Windows Server 2008 Enterprise x64 edition with SP1 were used for the first part of the experiments. The VMs ran on two identical vSphere hosts (2 VMs per host). A 25GB virtual disk created in a VMFS datastore was used to install the guest OS and application binaries in each VM. All the VMs shared a separate VMFS datastore created on a RAID 1/0 LUN (4+4 disks). Four identical virtual disks were created on the shared datastore and one was assigned to each VM. The virtual disks were used to store the data and index files of the DVD Store (DS2) workload running in each VM. A separate RDM¹ disk was used in each VM to store the log files of the DS2 workload.

DS2 in one of the VMs (VM2) was identified as a critical workload. Figure 1 shows the testbed layout.

¹ Following SQL server best practices, the log files were placed on separate virtual disks. A single physical disk was dedicated to each VM and mapped as RDM for placing the log files.

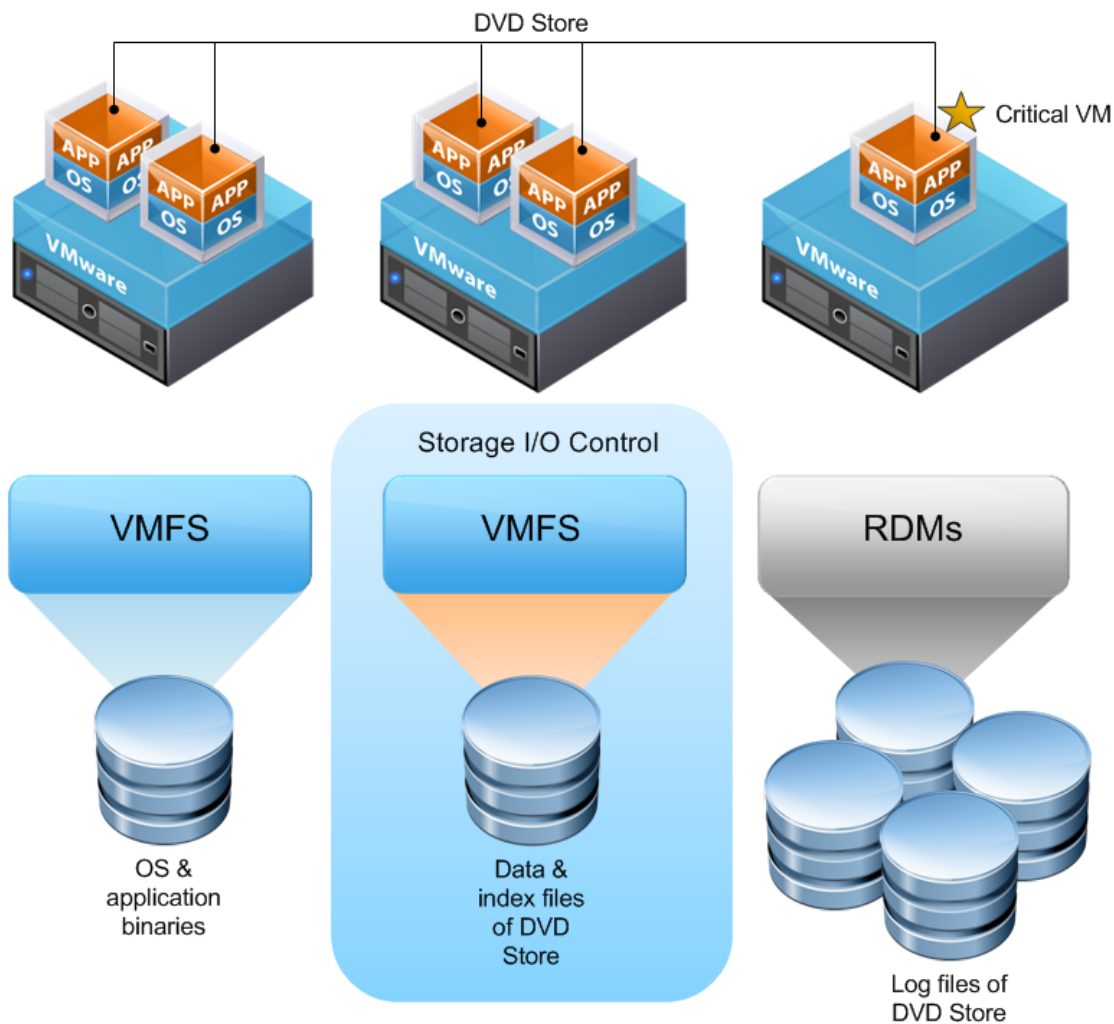
Figure 1. Testbed layout with 4 VM Configuration



Setup for Tests 2 and 3 with 5 VMs

An additional VM (VM5 in Figure 2) was added to the test mix. This VM ran on a separate vSphere host, but had its virtual disk for storing data and index files of the DS2 workload on the same shared datastore. The VM used 4 vCPUs and 8GB of memory. Otherwise, VM5 was identical to the remaining 4 VMs. The DS2 workload in this VM was identified as critical for tests 2 and 3. Figure 2 shows the testbed layout for tests 2 and 3.

Figure 2. Testbed layout with 5 VM configuration



Test and Measurement Tools

DS2 is an open source, online e-commerce test application, with a back-end database component, a Web application layer, and driver programs. For more information, see <http://www.delltechcenter.com/page/dvd+store>.

DS2 comes in three standard sizes: small, medium, and large. For our tests, we created a custom sized database that was between the medium and large sizes. Our workload used a database size of 20GB with 20,000,000 customers, 2,000,000 orders per month and 200,000 different products.

Test 1: Performance Impact of Storage I/O Control

In the first phase of this test case, the critical workload on VM2 was run in isolation (DS2 workloads in the other VMs were inactive) and its performance was recorded. Next, DS2 workloads in the rest of the VMs

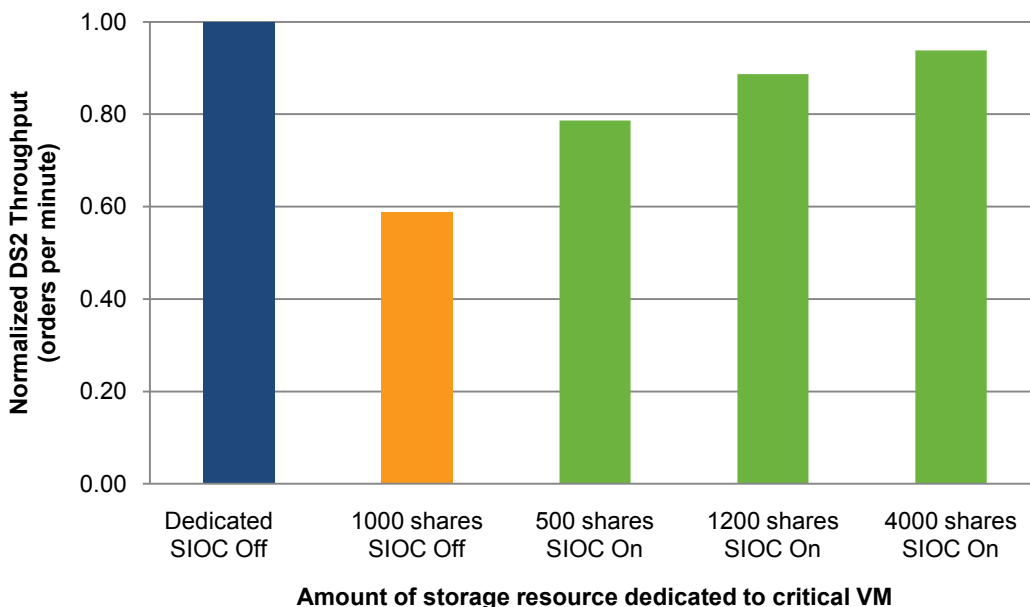
were made active with disk shares of each VM set as “normal” (default value of 1000). Performance of the critical workload was recorded when it ran at the same time as workloads in the other VMs. In the next phase, SIOC was enabled on the shared VMFS datastore. The congestion threshold value was set to 20 milliseconds. Each VM was assigned a disk share with the highest share allocated to VM2 as shown in Table 2. The number of disk shares assigned to VM2 was varied and performance of the critical workload was recorded in each case. Table 2 shows the placement of VMs, their disk shares and DS2 configuration in each VM.

Table 2. VM configuration and placement

VM ID	Host ID	Disk Shares		Number of DS2 Users
		SIOC OFF	SIOC ON	
1	1	1000	200	36
2	1	1000	500, 1200, 4000	50
3	2	1000	200	36
4	2	1000	200	36

Figure 3 compares the performance of the critical workload as the portion of the host’s I/O queue allocated to VM2 was varied.

Figure 3. Impact of Storage I/O Control on performance of DVD Store workload



The blue vertical bar, “Dedicated,” represents the performance of the critical DS2 workload when it ran in isolation with none of the other VMs sharing the datastore and SIOC not enabled on the datastore. In this case VM2 could use the entire I/O queue of the host.

The orange vertical bar, “1000 shares,” indicates the performance of the critical DS2 workload when DS2 workloads in all four VMs ran at the same time. In this case, all four VMs were assigned equal shares and SIOC was not enabled on the datastore. As a result, the performance of the critical DS2 workload dropped by 41%. The lack of a control mechanism to manage access to hosts’ I/O queues resulted in vSphere giving equal priority to each VM’s access to the I/O queue of the host on which the VM is running. VM2, though running a critical workload, got the same priority as other VMs running on that host and the other host. This resulted in a drop in performance.

The green vertical bars, “500 shares,” “1200 shares,” and “4000 shares,” represent the performance of the critical DS2 workload when SIOC was enabled on the shared datastore and DS2 workloads in all four VMs ran at the same time. In this case, vSphere dynamically managed the VMs’ access to each host’s I/O queue in proportion to their disk shares. VM2 was assigned a major chunk of its host’s I/O queue since it was assigned the highest number of shares. Performance of the critical workload improved as VM2’s disk shares were increased and reached 94% of the baseline when disk shares assigned to VM2 was maximum (4000).

Test 2: Intelligent Prioritization of I/O Resources

In the second test case, an additional VM running the DS2 workload was added into the mix (VM5). This VM ran on a separate host, increasing the number of hosts in the test bed to three. The data and index files of the DS2 workload in this VM were placed in a virtual disk on the shared datastore. The DS2 workload in VM5 was the critical workload for this and the next test.

SIOC was enabled on the shared datastore and the congestion threshold was set to 20 milliseconds. Table 3 shows the disk shares assigned to each VM. Because the DS2 workload in VM5 was identified as critical, VM5 was assigned the highest number (4000) of disk shares.

Table 3. VM configuration and placement

VM ID	Host ID	vCPU	Memory (in GB)	Disk Shares	Number of DS2 Users
1	1	2	8	500	24
2	1	2	8	500	24
3	2	2	8	750	24
4	2	2	8	750	24
5	3	4	8	4000	50

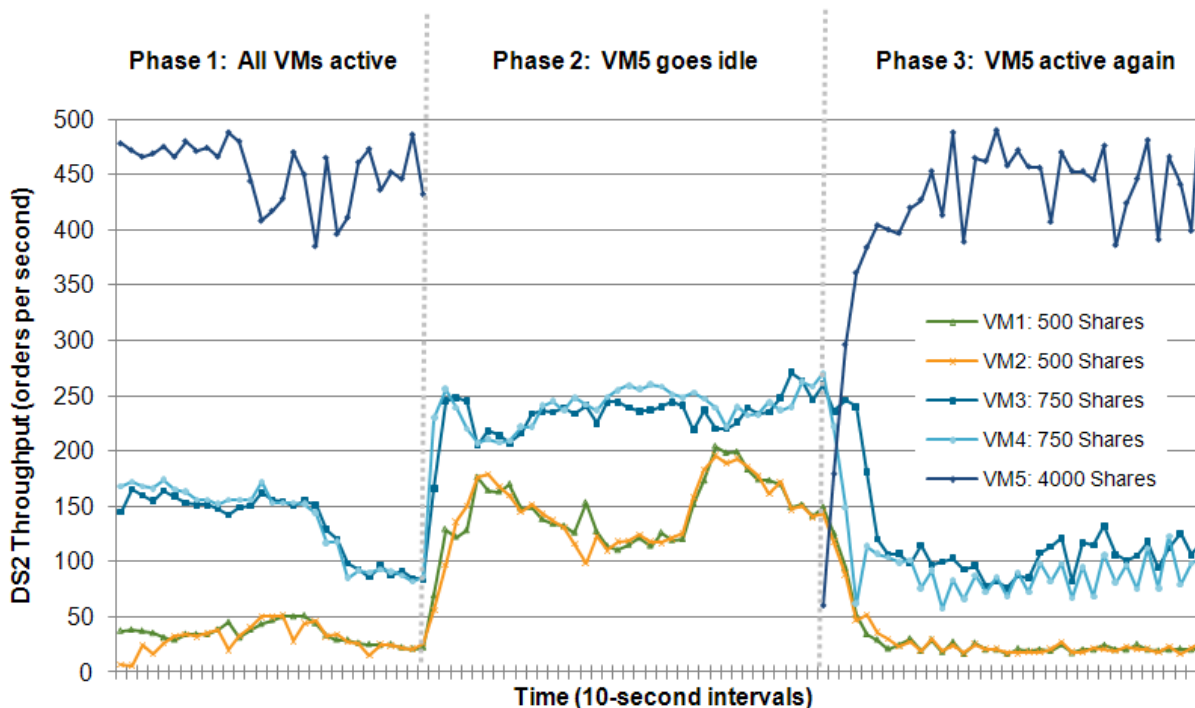
This test case had 3 phases:

- Phase 1: The DS2 workloads in all 5 VMs were active for 360 seconds.
- Phase 2: The DS2 workload in VM5 went idle for 345 seconds. DS2 in the remaining four VMs continued to be active.
- Phase 3: The DS2 workload in VM5 became active again for the next 360 seconds. DS2 in the remaining four VMs continued to be active.

Figure 4 shows the performance of the DS2 workload in each VM during different phases of the test. When all VMs were active, VM5 received the highest priority because it was assigned the highest number of disk shares. As a result, the DS2 workload in VM5 delivered the highest throughput (400 – 500 orders per

second) of all the VMs. When VM5 went idle, throughput of the DS2 workload in VMs 1 through 4 improved proportionally indicating that SIOC still maintained high utilization of the storage resources, but this time re-provisioned the I/O resources to the active VMs.

Figure 4. Application throughput of all VMs in different phases

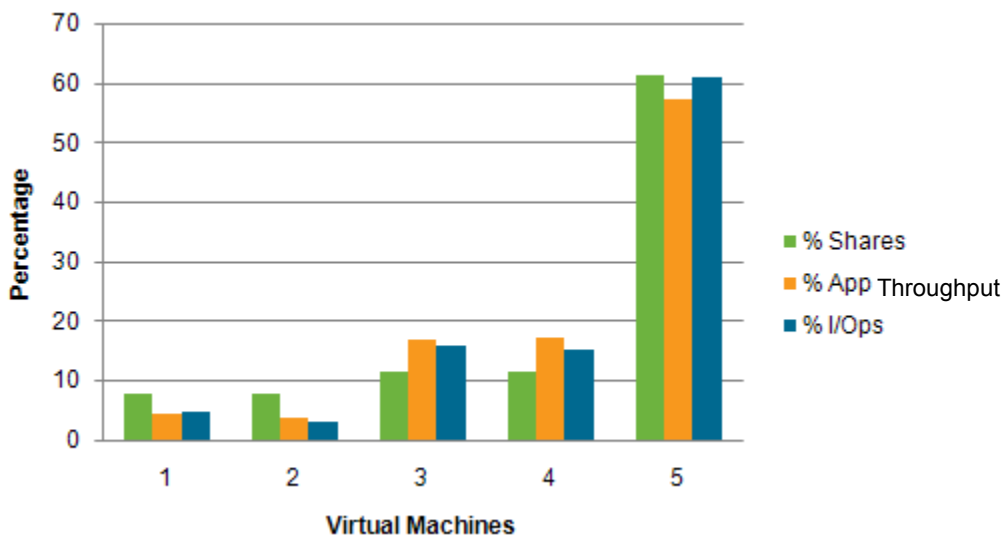


A more detailed look at what happened in each phase follows.

Phase 1: In this phase, DS2 workloads in all five VMs were active. This led to contention among all VMs for accessing the shared datastore, resulting in an increase in the I/O latency. As the datastore-wide I/O latency increased beyond the congestion threshold limit, SIOC detected the I/O congestion. At this stage, SIOC started limiting each VM's access to its host's I/O queue based on the I/O priority of the VM (disk shares). As a result, each VM's I/O requests per second became proportional² to its disk shares, which resulted in the DS2 performance of each VM being proportional to its disk shares as shown in Figure 5.

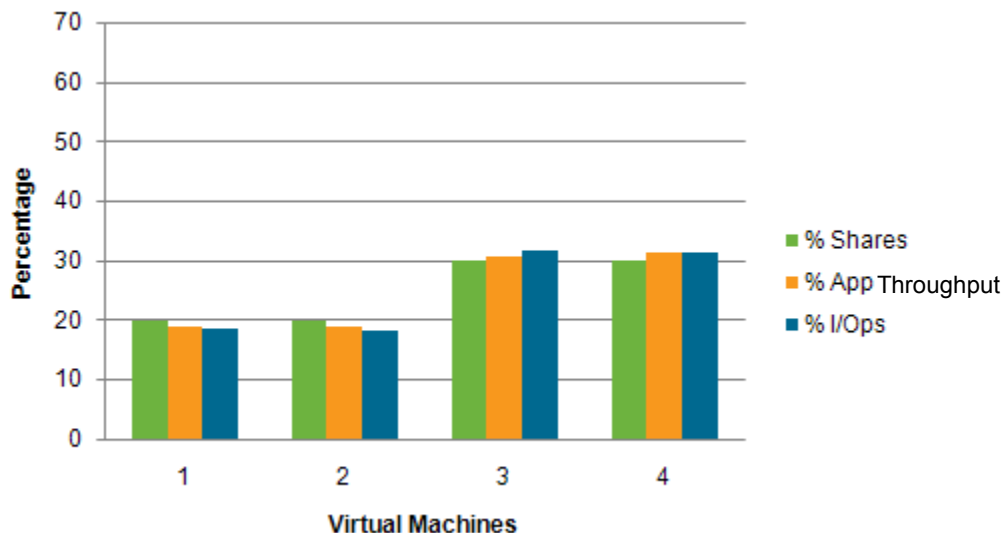
² SIOC's ability to accurately throttle hosts' device queues is also dependent on factors such as the read/write I/O access profile of all the VMs sharing the datastore and the max queue depth set in the HBA driver of all the hosts. These two factors can cause the computation of a host's device queue length to be non-trivial, which was the case in this test. A detailed explanation of how they influenced SIOC is beyond the scope of this paper.

Figure 5. Ratios of DS2 throughput, I/O requests per second, and VM disk shares in phase 1



Phase 2: At the end of phase 1, the critical DS2 workload in VM5 went idle. However, the DS2 workloads in the remaining four VMs continued to be active. VMs 1 through 4 still contended for access to the shared datastore. SIOC monitored the I/O contention, but this time detected that there were no active I/O requests from VM5. Since SIOC is designed to be dynamic, it reprioritized each active VM’s access to the shared datastore in less than a minute based on their disk shares. At this stage, VMs 3 and 4 had higher priority over VMs 1 and 2. SIOC adjusted the hosts’ queue depth and VMs’ access to the queues accordingly. This resulted in an increase of DS2 throughput in all active VMs in proportion to their shares as shown in Figure 6.

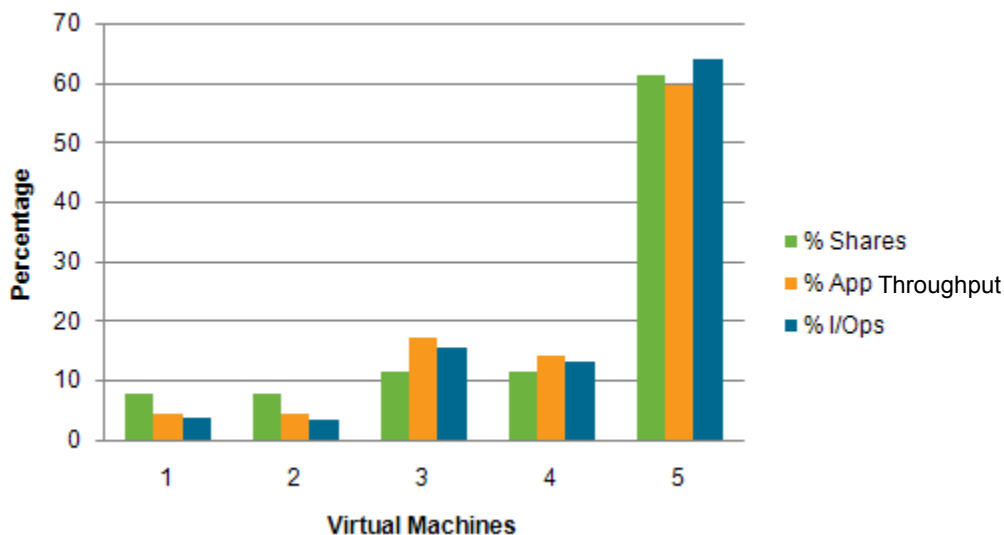
Figure 6. Ratios of DS2 throughput, I/O requests per second, and VM disk shares in phase 2



Phase 3: In phase 3, the critical DS2 workload in VM5 became active again and started issuing I/O requests. In less than a minute, SIOC detected that VM5 was active and once again reprioritized each VM’s access to the shared datastore. The access priority of each VM returned to similar levels as that seen in phase 1. DS2 performance in VMs 1 through 4 dropped to their phase 1 values, whereas DS2 performance of VM5

increased to its phase 1 value. Figure 7 shows that VM5 gains the most throughput because it has the most shares. Throughput of the other VMs is reduced to accommodate VM5.

Figure 7. Ratios of DS2 throughput, I/O requests per second, and VM disk shares in Phase 3



Device Queue Depth at the Host

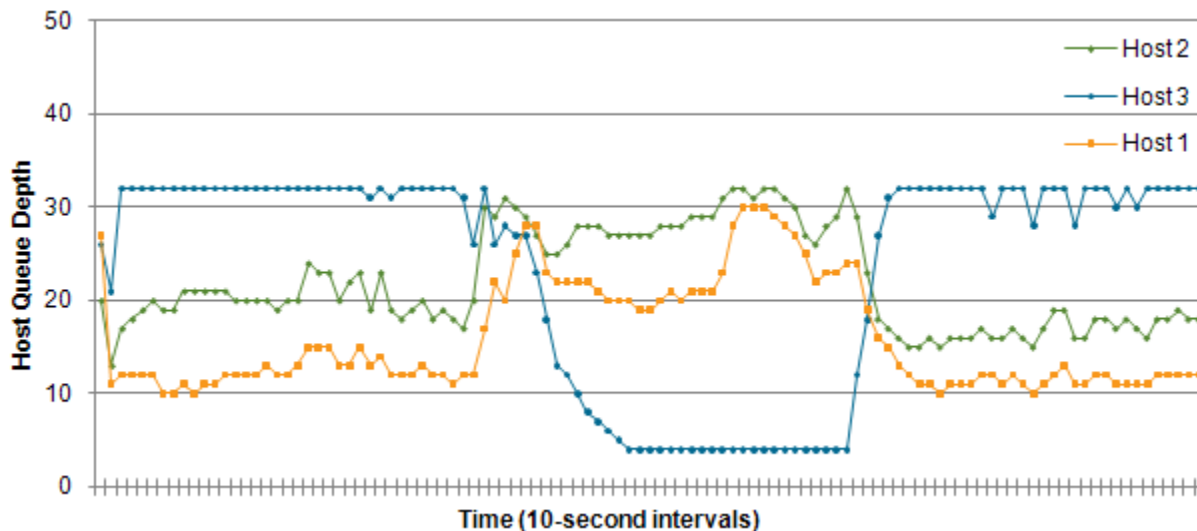
When a datastore is shared by more than one vSphere host, the array-level I/O queue of the logical device on which the datastore resides is shared among the hosts. SIOC, if enabled, manages the distribution of queue slots from the array-level I/O queue among the hosts by throttling the device queue in each host to keep the datastore latency close to the congestion threshold. The lower and upper bound on the current device queue in each host is as follows:

- Lower limit: 4
- Upper limit: minimum of (queue depth set by SIOC, queue depth set in the HBA driver)

Further, SIOC also manages the distribution of the queue slots from a host's device queue among the individual VMs running on the host. The distribution of the slots is based on several factors—relative priority (disk shares) of individual VMs, congestion threshold, and the current usage of the slots by the individual VMs. By allocating queue slots proportionally to each VM, SIOC regulates VMs' access to their host's device queue.

SIOC monitors the usage of the device queue in each host, aggregate I/O requests per second from each host, and datastore-wide I/O latency every four seconds and throttles the device queue in each host, if required. The actual device queue depth of each host in all 3 phases of this test is shown in Figure 8.

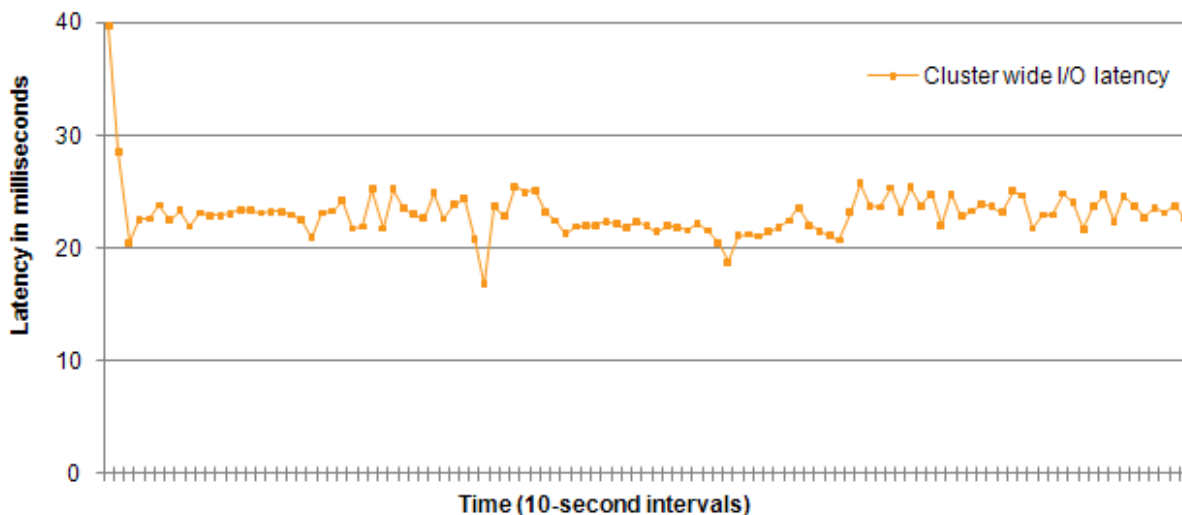
Figure 8. Host device queue depth



In the first and third phase of the test, host 3 was allowed to use a device queue of depth 32 (default maximum value). The device queue depth of hosts 1 and 2 was throttled down to 10-15 and 15-20 respectively from the default value of 32. When the critical DS2 workload in VM5 went idle, host 3’s queue depth was throttled down to 4 (the lowest limit for queue depth). The device queue depth of hosts 1 and 2 was increased proportionally according to their aggregate disk shares.

The datastore-wide I/O latency of the shared datastore during the entire test run is shown in Figure 9. As seen in the graph, SIOC tried to maintain the datastore-wide latency close to the threshold value (20 milliseconds in our case).

Figure 9. Datastore-wide I/O latency



Test 3: SIOC in a Datacenter

In the final test case, we show SIOC's effectiveness in preventing performance fluctuations of a critical workload, caused by contention for storage resource. The set up used for this test case represents a centralized datacenter servicing various offices on different coasts.

The test setup used for this test case was the same as that used in Test 2. Imagine VMs 1 through 4 to be order processing servers, and VM 5 to be a product catalogue server, with the VMs operating in different time zones. The order entry servers were active for different time periods based on the time zone in which they operated. The product catalogue server serviced regions in all time zones; hence, it remained active throughout the test duration.

- VM1 - East coast (order processing server)
- VM2 & VM3 - Central (order processing server)
- VM4 - West coast (order processing server)
- VM5 - Servicing all the offices (product catalogue server)

The order processing servers (VMs 1 through 4) were identified as less critical VMs and were assigned lower priority (500 disk shares). The product catalogue server (VM5) was identified as a critical VM and assigned the highest priority (4000 disk shares). VM configuration and placement are described in Table 5.

Table 5. VM configuration and placement

Virtual Machines	Host	vCPU	Memory (in GB)	Disk Shares	Number of DS2 Users
1	1	2	8	500	24
2	1	2	8	500	24
3	2	2	8	500	24
4	2	2	8	500	24
5	3	4	8	4,000	50

Load Period: The workloads in all VMs were started and stopped according to the schedule shown in Table 6.

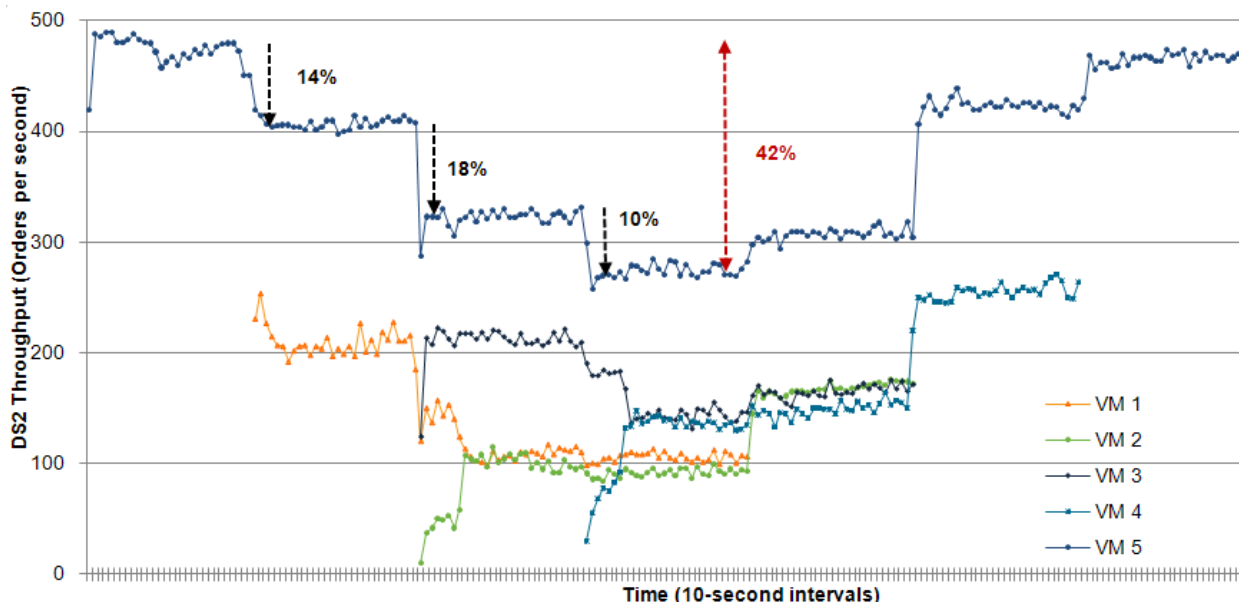
Table 6. Load period schedule

Time in seconds	Activity
0	VM5 starts
300	VM1 starts
600	VM2 and VM3 start
900	VM4 starts

1200	VM1 stops
1500	VM2 and VM3 stop
1800	VM4 stops
2100	VM5 stops

Figure 10 shows performance of the DS2 workload in all five VMs with SIOC turned off (default setting).

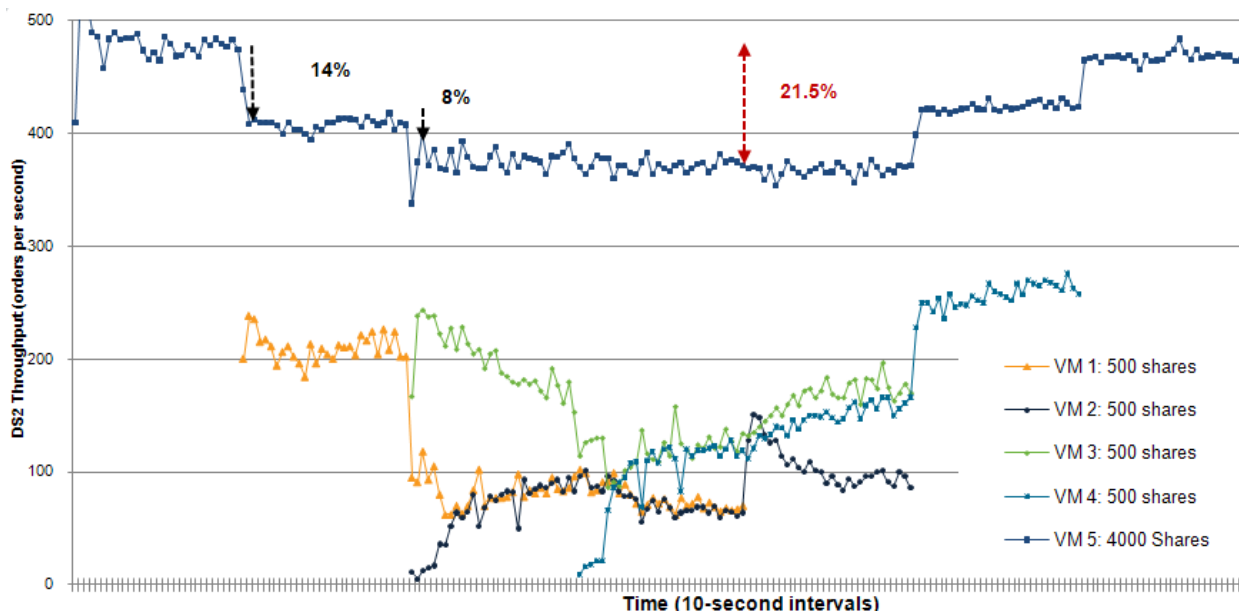
Figure 10. Test results when SIOC is turned off



As seen in Figure 10, every time one of the low priority order processing server VMs was active, performance of the catalogue server VM was affected. When order processing server 1 (VM1) was active, performance of the catalogue server VM dropped by 14%. When order processing servers 2 and 3 became active, performance dropped by an additional 18%. Ultimately, when order processing server VM4 also became active, performance of the catalogue server VM degraded by another 10%. The bottom line is that performance of the catalogue server VM suffered from 10% to 18% every time an order processing server became active, and eventually degraded by as much as 42% from its peak performance because there was no mechanism to prioritize the I/O access to the shared datastore.

The same experiment was repeated after turning on SIOC for the VMFS datastore that was shared between the VMs. This time the congestion threshold was set to 25 milliseconds. Figure 11 shows the results.

Figure 11. Test results when SIOC is turned on

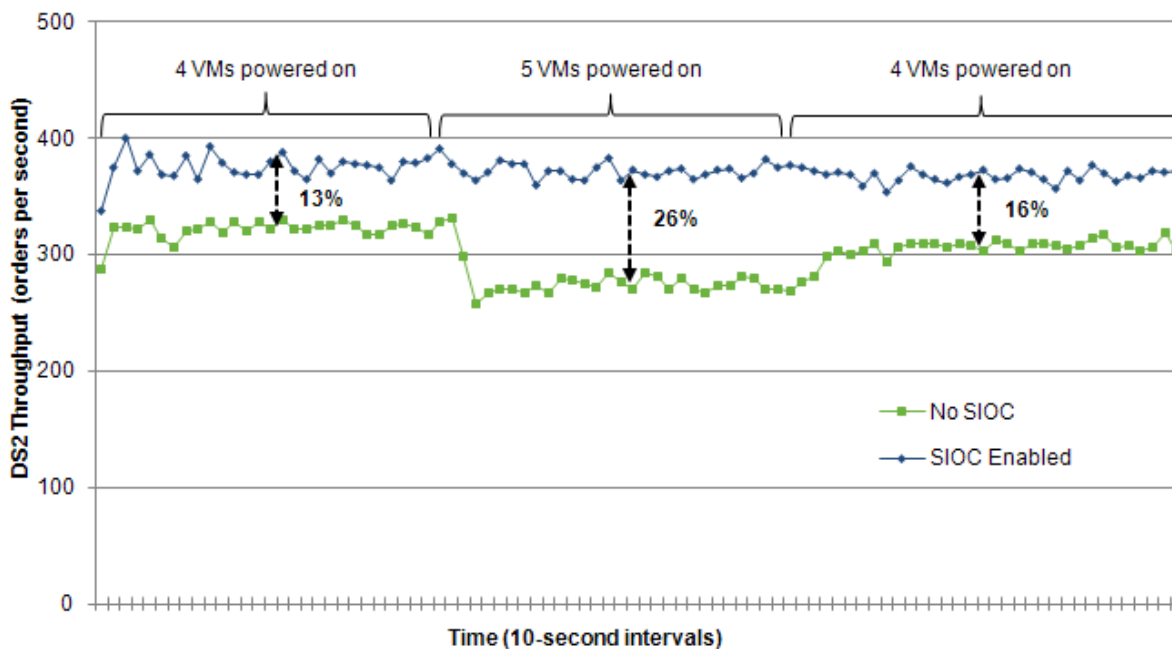


As one of the order processing server VMs became active and started issuing I/O requests to the shared datastore, latency of the I/O accesses increased noticeably. But the datastore-wide I/O latency was still under the congestion threshold limit. Therefore, vSphere did not assign any priority to I/O requests from either of the VMs; the VMs were allowed to access the shared datastore equally. The increase in I/O latency resulted in a 14% decrease in performance of the catalogue server VM.

When more than one of the order processing server VMs became active, I/O latency increased above the congestion threshold limit. At this stage, SIOC started prioritizing each VM's access to the shared datastore in proportion to the VM's disk shares. The catalogue server VM received the highest priority because it had the highest number of disk shares. Hence its performance remained unaffected irrespective of the number of active order processing server VMs. The performance of the catalogue server VM was at a constant 78.5% of its peak value throughout the congestion period.

Benefits of Storage I/O Control

Figure 12 compares the application throughput of the catalogue server VM with and without SIOC enabled on the shared datastore when order processing server VMs 2, 3, and 4 were active.

Figure 12. DS2 throughput of the critical workload with and without SIOC enabled

From the graph, we can infer that:

- When SIOC was enabled, loss of application performance in the catalogue server VM was predictable and manageable. Without SIOC, loss in performance was unmanageable. Performance deteriorated every time an order processing server VM became active.
- When SIOC was enabled, performance of the catalogue server VM recovered by 13% to 26% during the congestion period compared to that when SIOC was not used.

Conclusion

Sharing storage infrastructure to service I/O demands of applications in a virtual infrastructure offers many benefits—better utilization of resources, reduced cost of ownership, higher performance, and ease of management. However, peak demands for I/O resources can lead to contention for I/O resources among the virtual machines. This results in performance reduction for applications running in those VMs. Storage I/O Control provides a dynamic control mechanism for managing VMs' access to I/O resources in a cluster. The experiments conducted in VMware performance labs show that SIOC effectively prioritizes and redistributes I/O resources while maintaining a high resource utilization. This allows high priority VMs to maintain a higher level of performance even during peak loads when there is contention for I/O resources.

References

- “PARDA: Proportional Allocation of Resources for Distributed Storage Access,” paper accepted in USENIX Annual Technical Conference
Paper: http://www.usenix.org/events/fast09/tech/full_papers/gulati/gulati.pdf
Slides: <http://www.usenix.org/events/fast09/tech/slides/gulati.pdf>
- “Storage I/O Control Technical Overview and Considerations for Deployment”
<http://www.vmware.com/go/tp-storage-io-control>
- “TA3461 – Tech Preview IO DRS: VM Performance Isolation in Shared Storage Environments,” VMworld 2009 talk
- *vSphere Resource Management Guide* for ESX & ESXi 4.1
<http://pubs.vmware.com>

About the Author

Chethan Kumar is a senior member of Performance Engineering at VMware, where his work focuses on performance-related topics concerning database and storage. He has presented his findings in white papers and blog articles, and technical papers in academic conferences and at VMworld.

If you have comments about this documentation, submit your feedback to: docfeedback@vmware.com

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

Copyright © 2010 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

Item: EN-000415-00
