Advantages of using VMware Storage I/O Control with Hitachi Unified Compute Platform Select with VMware vSphere using Hitachi Unified Storage 150

Lab Validation Report

By Daniel Worden

March 29, 2013
Feedback

Hitachi Data Systems welcomes your feedback. Please share your thoughts by sending an email message to SolutionLab@hds.com. To assist the routing of this message, use the paper number in the subject and the title of this white paper in the text.
Advantages of using VMware Storage I/O Control with Hitachi Unified Compute Platform Select with VMware vSphere using Hitachi Unified Storage 150

Lab Validation Report

This lab validation report provides the validation of the advantages of using VMware Storage I/O Control (SIOC) with Hitachi Unified Compute Platform Select with VMware vSphere, using Hitachi Unified Storage 150.

Along with the advantages of virtualization using VMware in the data center, new challenges have also been discovered, such as the noisy neighbor problem.

One of the causes of the noisy neighbor problem is when the disk I/O of a virtual machine increases for short sporadic periods of time, from minutes to hours. This causes the performance of other virtual machines on the same VMware vCenter datastore to suffer as they lose I/O to the noisy neighbor.

Another cause of noisy neighbor problem is shifts or increases in workloads that span days or longer. With regular monitoring and analysis, workload changes can be predicted and adjustments made. These shifts may be caused from work that only occurs during certain cycles in the month, quarter, or year.

VMware developed SIOC to address the noisy neighbor problem. Use this tool for allocating available IOPS to virtual machines in the data center so you can meet service level agreements.

This testing environment was configured from Deploy Hitachi Unified Compute Platform Select for VMware vSphere using Hitachi Unified Storage 150 in a Scalable Environment Reference Architecture Guide (PDF).
Note — Testing of this configuration was in a lab environment. Many things affect production environments beyond prediction or duplication in a lab environment. Follow the recommended practice of conducting proof-of-concept testing for acceptable results in a non-production, isolated test environment that otherwise matches your production environment before your production implementation of this solution.
Product Features

These are the features of the products used in testing this solution.

Hitachi Unified Storage 150

**Hitachi Unified Storage** is a midrange storage platform for all data. It helps businesses meet their service level agreements for availability, performance, and data protection.

The performance provided by Hitachi Unified Storage is reliable, scalable, and available for block and file data. Unified Storage is simple to manage, optimized for critical business applications, and efficient.

Using Unified Storage requires a smaller capital investment. Deploy this storage, which grows to meet expanding requirements and service level agreements, for critical business applications. Simplify your operations with integrated set-up and management for a quicker time to value.

Unified Storage enables extensive cost savings through file and block consolidation. Build a cloud infrastructure at your own pace to deliver your services.

Hitachi Unified Storage 150 provides reliable, flexible, scalable, and cost-effective modular storage. Its symmetric active-active controllers provide input-output load balancing that is integrated, automated, and hardware-based.

Both controllers in Unified Storage 150 dynamically and automatically assign the access paths from the controller to a logical unit (LU). All LUs are accessible, regardless of the physical port or the server that requests access.

**Hitachi Dynamic Provisioning**

On Hitachi storage systems, **Hitachi Dynamic Provisioning** provides wide striping and thin provisioning functionalities.

Using Dynamic Provisioning is like using a host-based logical volume manager (LVM), but without incurring host processing overhead. It provides one or more wide-striping pools across many RAID groups. Each pool has one or more dynamic provisioning virtual volumes (DP-VOLs) of a logical size you specify of up to 60 TB created against it without allocating any physical space initially.

Deploying Dynamic Provisioning avoids the routine issue of hot spots that occur on logical devices (LDEVs). These occur within individual RAID groups when the host workload exceeds the IOPS or throughput capacity of that RAID group. Dynamic provisioning distributes the host workload across many RAID groups, which provides a smoothing effect that dramatically reduces hot spots.
When used with Hitachi Unified Storage, Hitachi Dynamic Provisioning has the benefit of thin provisioning. Physical space assignment from the pool to the dynamic provisioning volume happens as needed using 1 GB chunks, up to the logical size specified for each dynamic provisioning volume. There can be a dynamic expansion or reduction of pool capacity without disruption or downtime. You can rebalance an expanded pool across the current and newly added RAID groups for an even striping of the data and the workload.

**Hitachi Compute Blade 500**

Hitachi Compute Blade 500 combines the high-end features with the high compute density and adaptable architecture you need to lower costs and protect investment. Safely mix a wide variety of application workloads on a highly reliable, scalable, and flexible platform. Add server management and system monitoring at no cost with Hitachi Compute Systems Manager, which can seamlessly integrate with Hitachi Command Suite in IT environments using Hitachi storage.

The Hitachi Compute Blade 500 chassis contains internal Fibre Channel and network switches for the high availability requirements of Hitachi Unified Compute Platform for VMware vSphere.

**Brocade Storage Area Network Switches**

Brocade and Hitachi Data Systems have collaborated to deliver storage networking and data center solutions. These solutions reduce complexity and cost, as well as enable virtualization and cloud computing to increase business agility.

This reference architecture uses the following Brocade products:

- **Brocade 6510 Switch**
- **Brocade VDX 6720 Data Center Switch**

The network design allows for the utilization of advanced features inherent in the Brocade VDX switch family such as Brocade’s VCS Fabric technology. This helps provide the following:

- Non-stop networking
- Simplified, automated networks
- An evolutionary approach that protects existing IT investments

See the Brocade website for more information about Brocade VCS Fabric Technology.
VMware vSphere 5

**VMware vSphere 5** is a virtualization platform that provides a datacenter infrastructure. It features vSphere Distributed Resource Scheduler (DRS), High Availability, and Fault Tolerance.

VMware vSphere 5 has the following components:

- **ESXi 5** — This is a hypervisor that loads directly on a physical server. It partitions one physical machine into many virtual machines that share hardware resources.

- **vCenter Server** — This allows management of the vSphere environment through a single user interface. With vCenter, there are features available such as vMotion, Storage vMotion, Storage Distributed Resource Scheduler, High Availability, and Fault Tolerance.
Test Environment Configuration

Testing of VMware Storage I/O Control (SIOC) on Hitachi Unified Compute Platform Select with VMware vSphere took place in the Hitachi Data Systems laboratory. This documented the performance differences when using and not using SIOC with various queue sizes on Hitachi Compute Blade 500 and Hitachi Unified Storage 150.

The testing environment was configured from Deploy Hitachi Unified Compute Platform Select for VMware vSphere using Hitachi Unified Storage 150 in a Scalable Environment Reference Architecture Guide (PDF).

Table 1 describes the details of the components used when testing SIOC.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Description</th>
<th>Version</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hitachi Unified Storage 150</td>
<td>- Dual controllers&lt;br&gt;- 16 × 8 Gb/sec Fibre Channel ports&lt;br&gt;- 32 GB cache memory&lt;br&gt;- 72 × 600 GB 10k RPM SAS disks, 2.5 inch SFF</td>
<td>0917/A-H</td>
<td>1</td>
</tr>
<tr>
<td>Hitachi Compute Blade 500 chassis</td>
<td>- 8-blade chassis&lt;br&gt;- 2 Brocade 5460 Fibre Channel switch modules, each with 6 × 8 Gb/sec uplink ports&lt;br&gt;- 2 Brocade VDX 6746 Ethernet switch modules, each with 8 × 10 Gb/sec uplink ports&lt;br&gt;- 2 management modules&lt;br&gt;- 6 cooling fan modules&lt;br&gt;- 4 power supply modules</td>
<td>SVP: A0108-B-5923, 5460: FOS 6.3.2d, VDX6746: NOS 2.0.1_kat4</td>
<td>1</td>
</tr>
</tbody>
</table>
Table 1. Infrastructure Components (Continued)

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Description</th>
<th>Version</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>520BH1 server blade</td>
<td>Half blade</td>
<td>BMC/EFI: 01-27</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>2 × 8-core Intel Xeon E5-2680 processor, 2.70 GHz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>96 GB RAM</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 × 16 DIMMs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Brocade 6510</td>
<td>SAN switch with 48 × 8 Gb Fibre Channel ports</td>
<td>FOS 7.0.1a</td>
<td>2</td>
</tr>
<tr>
<td>Brocade VDX 6720</td>
<td>Ethernet switch with 24 × 10 Gb/sec ports</td>
<td>NOS 2.0.1b</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2 describes the software used in this reference architecture.

Table 2. Software Components

<table>
<thead>
<tr>
<th>Software</th>
<th>Version</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hitachi Storage Navigator Modular 2</td>
<td>Microcode Dependent</td>
</tr>
<tr>
<td>Hitachi Dynamic Provisioning</td>
<td>Microcode Dependent</td>
</tr>
<tr>
<td>VMware vCenter server</td>
<td>5.1.0</td>
</tr>
<tr>
<td>VMware Virtual Infrastructure Client</td>
<td>5.1.0</td>
</tr>
<tr>
<td>VMware ESXi</td>
<td>5.1.0</td>
</tr>
<tr>
<td>Microsoft® Windows Server® 2008</td>
<td>Enterprise edition, R2</td>
</tr>
<tr>
<td>Microsoft SQL Server 2008</td>
<td>Enterprise edition, R2</td>
</tr>
</tbody>
</table>

SAN Configuration

The Hitachi Unified Storage 150 controller used for this solution has 16 ports for connections to the Brocade 6510 enterprise Fabric switches.

The Hitachi Compute Blade 500 HBAs were zoned to four ports on the Hitachi Unified Storage 150 controller, two ports per controller.

Dedicating four ports to each Hitachi Compute Blade 500 chassis ensures bandwidth between the chassis and Hitachi Unified Storage 150.
Figure 1 illustrates the physical SAN architecture of the infrastructure cell for compute resources.

**Hitachi Compute Blade 500**

During the testing of SIOC, ten 10k RPM 600 GB hard disks were grouped into a single dynamic provisioning pool created by Hitachi Dynamic Provisioning. Then a single volume was created on the dynamic provisioning pool and presented to both ESXi hosts in the testing.
Network Configuration

The network design used in this solution provides ample bandwidth and redundancy for the following:

- A fully populated infrastructure cell for compute resources
- An infrastructure cell for storage resources
- Up to two expansion cells for compute resources

Figure 2 on page 10 illustrates the physical network architecture.
Figure 2
Virtualized Environment Configuration

The test environment had eight virtual machines on two ESXi hosts, as shown in Table 3.

Table 3. Virtual Machine to Host Placement

<table>
<thead>
<tr>
<th>ESXi Host</th>
<th>Virtual Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host_1</td>
<td>VM3, VM4, VM5, VM6, VM7</td>
</tr>
<tr>
<td>Host_2</td>
<td>VM1, VM2, VM8</td>
</tr>
</tbody>
</table>

All the virtual machines used during testing ran a database workload. Each virtual machine contained two virtual disks.

- **Virtual Disk 1** — Contained the NTFS-formatted virtual disk where the following was installed:
  - Microsoft® Windows Server® 2008 R2
  - SourceForge vdbench (I/O workload generator for verifying data integrity and measuring performance)

- **Virtual Disk 2** — Unformatted virtual disk where I/Os was issued to and from vdbench. This virtual disk was unformatted so the file system became a factor when determining raw I/O performance.
Test Methodology

The test methodology involved the following:

- Using and not using SIOC
- LUN queue depth of the HBAs set as follows for each test:
  - 32 (default)
  - 64
  - 128
  - 256
- Virtual disk shares set as following:
  - All virtual disks running equal share
  - Three of the eight virtual disks running with elevated disk shares

The test environment did not use VMware Distributed Resources Scheduler to control virtual machine host placement. The test environment did not use VMware Storage Distributed Resources Scheduler, as testing only used one datastore.

SourceForge vdbench generated the workload in each virtual machine, simulating a database workload as shown in Table 4.

<table>
<thead>
<tr>
<th>Table 4. Database I/O Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>I/O Rate</strong></td>
</tr>
<tr>
<td><strong>Seek Percent (Percent Random)</strong></td>
</tr>
<tr>
<td><strong>Read Percent</strong></td>
</tr>
<tr>
<td><strong>I/O Size</strong></td>
</tr>
</tbody>
</table>

The configuration was changed in vdbench to test using this number of threads:

- 4
- 8
- 16
- 32
Test Case 1: Virtual Machines Running Workload with Equal Disk Shares

This case tested 32 different configurations, as shown in Table 5.

Table 5. Scenarios for Test Case 1

<table>
<thead>
<tr>
<th>Configuration</th>
<th>SIOC</th>
<th>HBA LUN Queue Size</th>
<th>I/O Rate</th>
<th>Seek %</th>
<th>Read %</th>
<th>Threads</th>
<th>I/O Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Off</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>2</td>
<td>Off</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>3</td>
<td>Off</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>4</td>
<td>Off</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>5</td>
<td>On</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>6</td>
<td>On</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>7</td>
<td>On</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>8</td>
<td>On</td>
<td>32</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>9</td>
<td>Off</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>10</td>
<td>Off</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>11</td>
<td>Off</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>12</td>
<td>Off</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>13</td>
<td>On</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>14</td>
<td>On</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>15</td>
<td>On</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>16</td>
<td>On</td>
<td>64</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>17</td>
<td>Off</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>18</td>
<td>Off</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>19</td>
<td>Off</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>20</td>
<td>Off</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>21</td>
<td>On</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>22</td>
<td>On</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>23</td>
<td>On</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>24</td>
<td>On</td>
<td>128</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>25</td>
<td>Off</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>26</td>
<td>Off</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>27</td>
<td>Off</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>28</td>
<td>Off</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
<tr>
<td>29</td>
<td>On</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>4</td>
<td>8K</td>
</tr>
<tr>
<td>30</td>
<td>On</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>8</td>
<td>8K</td>
</tr>
<tr>
<td>31</td>
<td>On</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>16</td>
<td>8K</td>
</tr>
<tr>
<td>32</td>
<td>On</td>
<td>256</td>
<td>Max</td>
<td>100</td>
<td>75</td>
<td>32</td>
<td>8K</td>
</tr>
</tbody>
</table>
Test Case 2: Virtual Machines Running Workload with Different Disk Shares

Table 6 has the disk share settings in Test Case 2 for each virtual machine. The purpose of this test case was to assess the impact of prioritization caused by virtual disk shares.

Table 6. Number of Disk Shares per Virtual Machine for Test Case 2

<table>
<thead>
<tr>
<th>Virtual Machine</th>
<th>Number of Disk Shares</th>
</tr>
</thead>
<tbody>
<tr>
<td>VM1</td>
<td>3,000 (Custom)</td>
</tr>
<tr>
<td>VM2</td>
<td>1,000 (Normal)</td>
</tr>
<tr>
<td>VM3</td>
<td>3,000 (Custom)</td>
</tr>
<tr>
<td>VM4</td>
<td>1,000 (Normal)</td>
</tr>
<tr>
<td>VM5</td>
<td>1,000 (Normal)</td>
</tr>
<tr>
<td>VM6</td>
<td>1,000 (Normal)</td>
</tr>
<tr>
<td>VM7</td>
<td>1,000 (Normal)</td>
</tr>
<tr>
<td>VM8</td>
<td>1,000 (Normal)</td>
</tr>
</tbody>
</table>

This case tested the same scenarios as used for Test Case 1, as found in Table 5, “Scenarios for Test Case 1,” on page 13.
Analysis

This is an analysis of the results of the testing of VMware Storage I/O Control (SIOC) in the Hitachi Data Systems laboratory to document the performance on Hitachi Compute Blade 500 and Hitachi Unified Storage 150.

See “Test Results,” starting on page 17, for the results of each test case.

Test Case 1: Virtual Machines Running Workload with Equal Disk Shares

Testing showed that SIOC decreased the average application latency when increasing the default HBA per LUN queue depth. Testing also showed for heavier workloads, the greater SIOC improved the average application latency.

In this test case, the virtual machine disk configuration had an equal number of disk shares for each of the 32 different configurations. Analysis concludes the following from the test results:

- Enabling SIOC with the HBA LUN queue depth greater than the default value of 32 helps improve application latency. The results are at least modest under light workloads and significant under heavier workloads.
- Heavier workloads have the larger application latency.
- There was no degradation in storage system performance when increasing the HBA LUN queue depth, whether using or not using SIOC.
- In a scenario where two hosts access the same volume, the host with the smaller I/O load experiences lower application latency than the host with the larger I/O load. SIOC helped to reduce the application latency difference between the hosts. This behavior happens when the LUN queue depth is set above 32 under heavy I/O loads.
- When virtual machines run on the same datastore and on different hosts, SIOC did not totally equalize the virtual machine disk performance, but did bring the disk performance closer to equal.
- The virtual machine host placement has an impact on performance. When accessing the same volume, a virtual machine on a host with a heavier I/O load experiences more latency when compared to a host with lighter I/O load. SIOC helped to minimize the latency difference between the hosts.
Test Case 2: Virtual Machines Running Workload with Different Disk Shares

In this test case, two of the virtual machines had 3000 disk shares and six of the virtual machines had 1000 disk shares for each of the 32 different configurations. Analysis concludes the following from the test results:

- All the findings from Test Case 1 still applied with the disproportionate configuration of the disk shares on the virtual disks in Test Case 2.
- Virtual machines with more disk shares received better performance than virtual machines with fewer disk shares, even if the virtual machine with more disk shares was running on a host with a heavier workload.
- SIOC and disk shares have no significant impact on storage performance when the LUN queue depth of the HBA is left at the default settings, but do have a significant impact when increased to 64 and more.
Test Results

These are the test results used to validate the environment.

Test Case 1: Virtual Machines Running Workload with Equal Disk Shares

Figure 3 shows how SIOC impacted the average disk latency for all the eight virtual machines during all the test configurations, with and without SIOC.
Figure 3 on page 17 shows that SIOC had a significant impact on the disk performance of the virtual machines and lowers the average disk latency. The virtual disks with the highest latency drops from 180 milliseconds without SIOC to 140 milliseconds with SIOC.

The results were compared to different LUN queue depths for the LUNs and changing the workload by increasing the number of threads running on each virtual machine. These different results were tested with and without SIOC.

**LUN Queue Depth of 32 — 4 Threads per Virtual Machine**

The first test ran was with the default a LUN queue depth of 32. There were eight virtual machines with a workload of four threads on each virtual machine resulting in an approximate LUN queue usage of 32. The vdbench I/O rate of each virtual machine was set to max throughput.

SIOC was not effective in balancing the latency between the virtual machines on the host with the lighter workload (host2, with 3 virtual machines) and the host with the heavier workload (host 1 with 5 virtual machines) using the default LUN queue depth on the HBA.

Also with four threads per virtual machine, virtual machine latency was scattered. The virtual machines on the host with the heavier workload did not share common disk latency.

Figure 4 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 32
- 32 total disk threads

![VM Latency](image-url)
The entire HBA per LUN queue was being consumed, but was not oversubscribed.

**LUN Queue Depth of 32 — 8 Threads per Virtual Machine**

Changing the workload from four threads to eight threads per virtual machine divides the latency between the two hosts. The three virtual machines on the host with the lighter workload had a lower latency than the five virtual machines on the host with the heavier workload.

Figure 5 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 32
- 64 total disk threads

![VM Latency](image)

**Figure 5**

This means that the HBA per LUN queue was oversubscribed by 32 threads or 200%. The disk I/O that could not be processed through the HBA per LUN queue were pending in the VMkernel of the ESXi host.

**LUN Queue Depth of 32 — 32 Threads per Virtual Machine**

When the HBA LUN queue depth remained constant but the workload increased, the following also increased:

- Number of threads
- Applications disk I/O latency

The disk latency of the virtual machines on the host with heavier workloads, Host1, remained higher than the disk latency of the virtual machines on the host with the lighter workload, Host2.
Figure 6 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 32
- 256 total disk threads

The HBA per LUN queue was oversubscribed by 224 threads or 800%. The disk I/O that could not be processed through the HBA per LUN queue were pending in VMkernel of the ESXi hosts.

**LUN Queue Depth of 256 — 4 Threads per Virtual Machine**

When the workload was restricted to 4 threads per virtual machine and the LUN queue depth on the LUN was increased to 256, SIOC did not have much influence over the workload latency. Mostly, the latency division was based on the ESXi host on which the virtual machine was running.

Figure 7 on page 21 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 256
- 32 total disk threads
Figure 7

This means that the HBA per LUN queue was under subscribed by 224 threads or 12.5% utilized. There were no I/O requests pending in the VMkernel.

LUN Queue Depth of 256 — 32 Threads per Virtual Machine

When SIOC was enabled, the LUN queue depth of the HBA was set to 256 and the workload set to 32 threads per virtual machine, there is a difference in latency between the virtual machines on the host with the heavier and lighter workloads.
Figure 8 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 256
- 256 total disk threads

**Figure 8**

The HBA per LUN queue was fully subscribed. There were insignificant numbers of I/O requests pending in the VMkernel.

When comparing workloads with the use of SIOC to the same workloads without the use of SIOC, the results show that SIOC has a significant impact on the performance of the virtual machines.

Figure 9 on page 23 shows the results for the workload with the following:

- Without SIOC
- HBA per LUN queue depth configured to 256
- 256 total disk threads
The HBA per LUN queue was fully subscribed. There were insignificant numbers of I/O requests pending in the VMkernel.

**Test Case 2: Virtual Machines Running Workload with Different Disk Shares**

- The Test Case 2 used the same configurations as Test Case 1, with the following exceptions:

- Each of these virtual machines were configured with 3000 disk shares:
  - VM1
  - VM3

- Each of these virtual machines were configured with 1000 disk shares:
  - VM2
  - VM4
  - VM5
  - VM6
  - VM7
  - VM8
LUN Queue Depth of 32 — 4 Threads per Virtual Machine

This used the same workload as in Test Case 1 in “LUN Queue Depth of 32 — 4 Threads per Virtual Machine” on page 18, except for the changed configuration for Test Case 2.

Figure 10 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 32
- 32 total disk threads

![VM Latency](image)

**Figure 10**

The HBA per LUN queue was fully subscribed. There were insignificant numbers of I/O requests pending in the VMkernel.

When comparing the results of Test Case 1 in Figure 4 on page 18 to the results of Test Case 2 in Figure 10, there is no change. Virtual machine latency was random and fully scattered between 15 and 25 milliseconds in both test cases. The elevated disk shares did not have much effect on the latency.

LUN Queue Depth of 32 — 8 Threads per Virtual Machine

This used the same workload as in Test Case 1 in “LUN Queue Depth of 32 — 8 Threads per Virtual Machine” on page 19, except for the changed configuration for Test Case 2.

Figure 11 on page 25 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 32
- 64 total disk threads
Figure 11

The HBA per LUN queue was oversubscribed by 32 threads or 200%. The disk I/O that could not be processed through the HBA per LUN queue were pending in Vmkernel in the ESXi hosts.

When comparing the results of Test Case 1 in Figure 5 on page 19 to the results of Test Case 2 in Figure 11, there is little change. The disk shares have less effect than virtual machine placement, as VM1 (3,000 disk shares), VM2 (1,000 disk shares), and VM8 (1,000 disk shares) have lower disk latency than VM3 (3,000 disk shares). The placement of VM1, VM2, and VM8 on Host2 (lighter workload) was more significant in this configuration than was disk shares.

LUN Queue Depth of 256 — 4 Threads per Virtual Machine

This used the same workload as in Test Case 1 in Figure on page 20, except for the changed configuration for Test Case 2.

Figure 12 on page 26 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 256
- 32 total disk threads
Figure 12

The HBA per LUN queue was undersubscribed by 224 threads. There were no I/O requests pending in the VMkernel.

When comparing the results of Test Case 1 in Figure 5 on page 19 to the results in Test Case 2 in Figure 12, there is little change. The disk shares have little effect. Increasing the HBA per LUN queue depth while holding the work load low has little effect on how SIOC works, when testing four threads per virtual machine.

Disk shares do not have a strong impact until the number of threads per virtual machine goes from 8 to 16, even with a default queue depth set.

LUN Queue Depth of 32 — 16 Threads per Virtual Machine

This used the same workload as in Test Case 1 in “LUN Queue Depth of 32 — 32 Threads per Virtual Machine” on page 19, except for the changed configuration for Test Case 2.
Figure 13 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 32
- 128 total disk threads

Figure 13

The HBA per LUN queue was oversubscribed by 96 threads or 400%. The disk I/O that could not be processed through the HBA per LUN queue was pending in the VMkernel of the ESXi hosts.

The two virtual machines with the highest disk shares have the lowest latency, VM1 (Host2) and VM3 (Host1).

The virtual machines with the next lowest latency are VM2 and VM8. These are the remaining two virtual machines on the host with the lightest workload, Host2.

The virtual machines with the highest latency are VM4, VM5, VM6, and VM7. These are the remaining four virtual machines on the host with the heaviest workloads.

LUN Queue Depth of 256 — 32 Threads per Virtual Machine

This used the same workload as in Test Case 1 in “LUN Queue Depth of 256 — 32 Threads per Virtual Machine” on page 21, except for the changed configuration for Test Case 2.
Figure 14 shows the results for the workload with the following:

- Using SIOC
- HBA per LUN queue depth configured to 256
- 256 total disk threads

The HBA per LUN queue was not oversubscribed. There was no disk I/O that could not be processed through the HBA per LUN queue and were pending in the VMkernel of the ESXi hosts.

As the queue depth and workload increase, the effects of the disk share continue. VM1 and VM3 continue to get lower latency, due to having the higher 3,000 disk shares, although the virtual machines with the lower 1,000 disk shares have lower latency. Host2, with its lower I/O loads, continues to provide lower latency than Host1, with its higher I/O loads. This behavior continues, even as the workload and queue depth increase.
For More Information

Hitachi Data Systems Global Services offers experienced storage consultants, proven methodologies and a comprehensive services portfolio to assist you in implementing Hitachi products and solutions in your environment. For more information, see the Hitachi Data Systems Global Services website.

Live and recorded product demonstrations are available for many Hitachi products. To schedule a live demonstration, contact a sales representative. To view a recorded demonstration, see the Hitachi Data Systems Corporate Resources website. Click the Product Demos tab for a list of available recorded demonstrations.

Hitachi Data Systems Academy provides best-in-class training on Hitachi products, technology, solutions and certifications. Hitachi Data Systems Academy delivers on-demand web-based training (WBT), classroom-based instructor-led training (ILT) and virtual instructor-led training (vILT) courses. For more information, see the Hitachi Data Systems Services Education website.

For more information about Hitachi products and services, contact your sales representative or channel partner or visit the Hitachi Data Systems website.