TECHNICAL PAPER

Integrate Hitachi Virtual Storage Platform G1000 with VMware VAAI

Lab Validation Report

By Daniel Worden

April 16, 2014

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.

Table of Contents

Integrate Hitachi Virtual Storage Platform G1000 with VMware VAAI

Lab Validation Report

Data center administrators look for ways to improve scalability, performance, and efficiency to reduce administrative overhead and costs. One way to do this in a VMware environment is through the integration of Hitachi Virtual Storage Platform G1000 with VMware vStorage APIs for Array Integration (VAAI).

VAAI is a set of APIs, or primitives, that allow ESXi hosts to offload processing for certain data-related services to the Virtual Storage Platform G1000. This can enable significant improvements in virtual machine performance, virtual machine density, and availability in vSphere 5.5 environments. Moving these functions to the storage array offers many benefits, but requires the use of a highly available, scalable, high performance storage system like Virtual Storage Platform G1000.

Virtual Storage Platform G1000 works seamlessly with VMware vSphere 5.5 and ESXi 5.5 virtualization software. This enhances vSphere tasks such as Storage vMotion, cloning, and provisioning of new virtual machines. It also reduces SCSI reservation locking overhead, which can lead to decreased performance.

Hitachi Dynamic Provisioning enables the creation of a storage pool from which capacity can be used as needed to further improve performance, scalability, and utilization.

This white paper validates the benefits of using VAAI with Virtual Storage Platform G1000. It is written for storage administrators, vSphere administrators, and application administrators who manage large, dynamic environments. It assumes familiarity with SAN-based storage systems, VMware vSphere, and general IT storage practices.

VMware VAAI Overview

VMware VAAI enables execution of key data operations at the storage array level rather than at the ESXi server layer. This reduces resource utilization and potential bottlenecks on physical servers. It also enables more consistent server performance and higher virtual machine density.

When used with vSphere 5.5, Virtual Storage Platform G1000 supports the following API primitives:

XCOPY (Extended Copy) - Enables the storage system to make full copies of data within the storage system without having the ESXi host read and write the data. Read more about XCOPY (Extended Copy).

Write Same (Zero) Enables storage systems to zero out a large number of blocks to speed provisioning of virtual machines. Read more about Write Same (Zero).

Atomic Test & Set (ATS) Provides an alternative means to protect the metadata for VMFS cluster file systems, thereby improving the scalability of large ESXi host farms sharing a datastore. Read more about Atomic Test & Set (ATS).

Thin Provisioning-STUN (TP-STUN) - sends an alert to the VMware ESXi host when a thin provisioned pool is full so the ESXi host can temporarily pause the virtual machines. When additional space has been added to the thin provisioned pool or space is freed in the thin provisioned pool, the virtual machines can be resumed without failure or corruption. Read more about Thin Provisioning-STUN (TP-STUN).

Dead Space Reclamation (UNMAP) - Allows the ESXi host to inform the VSP G1000 that files or VMs have been moved or deleted from a Thin Provisioned VMFS datastore. This allows Virtual Storage Platform G1000 to reclaim the freed blocks for the storage pool. Read more about Dead Space Reclamation (UNMAP).

Table 1 shows a summary of the key findings for the testing of the VAAI primitives listed above.

Table 1. Key Findings

VAAI Primitive	Benefit
XCOPY (Extended Copy)	Speeds virtual machine deployment with reduction of host HBA utilization.
Write Same (Zero)	Enables storage systems to zero out a large number of blocks to speed provisioning of virtual machines. When used with Hitachi Dynamic Provisioning, all virtual disk types are thin provisioned, including eagerzeroedthick.
Atomic Test & Set (ATS)	Reduces locking conflicts and accelerates large-scale vMotion. This improves scalability of the vSphere infrastructure. Allows the creation of large VMFS volumes (up to a 2 TB single partition), thus simplifying storage configuration and sizing of LDEVs for VMFS volumes.
Thin Provisioning-STUN (TP-STUN)	Pauses thin provisioned virtual machine when the underlying storage system becomes full, and the VMDK for the virtual machine needs to grow.
Dead Space Reclamation (UNMAP)	Allows the storage array to reclaim disk space that was once used by a virtual machine that has been Storage vMotioned off of or deleted from a VMFS datastore.

XCOPY (Extended Copy)

For common ESXi administration tasks, use the full copy primitive on the ESXi host to offload the actual data copy to a Virtual Storage Platform. The full copy primitive helps with tasks such as provisioning virtual machines or migrating VMDK files between datastores within a storage system using Storage vMotion. The following operations are some examples of when the full copy primitive is used:

Virtual Machine Provisioning — The source and destination locations are within the same volume. Hitachi integrates with the full copy API to clone virtual machines or datastores from a golden image. This process dramatically reduces I/O between the ESXi nodes and Hitachi storage.

Storage vMotion — The source and destination locations are on different volumes within the same storage system. This feature enables VMDK files to be relocated between datastores within a storage system. Virtual machines can be migrated to facilitate load-balancing or planned maintenance without service interruption. By integrating with full copy, host I/O offload for VMware Storage vMotion operations accelerates virtual machine migration times considerably.

Figure 1 compares copy functions with and without VMware VAAI. As shown, the full copy primitive removes the ESXi host from the data path of the VMDK cloning operation. This reduces the number of disk I/Os from the ESXi host, saving host-side I/O bandwidth while copying virtual machines.

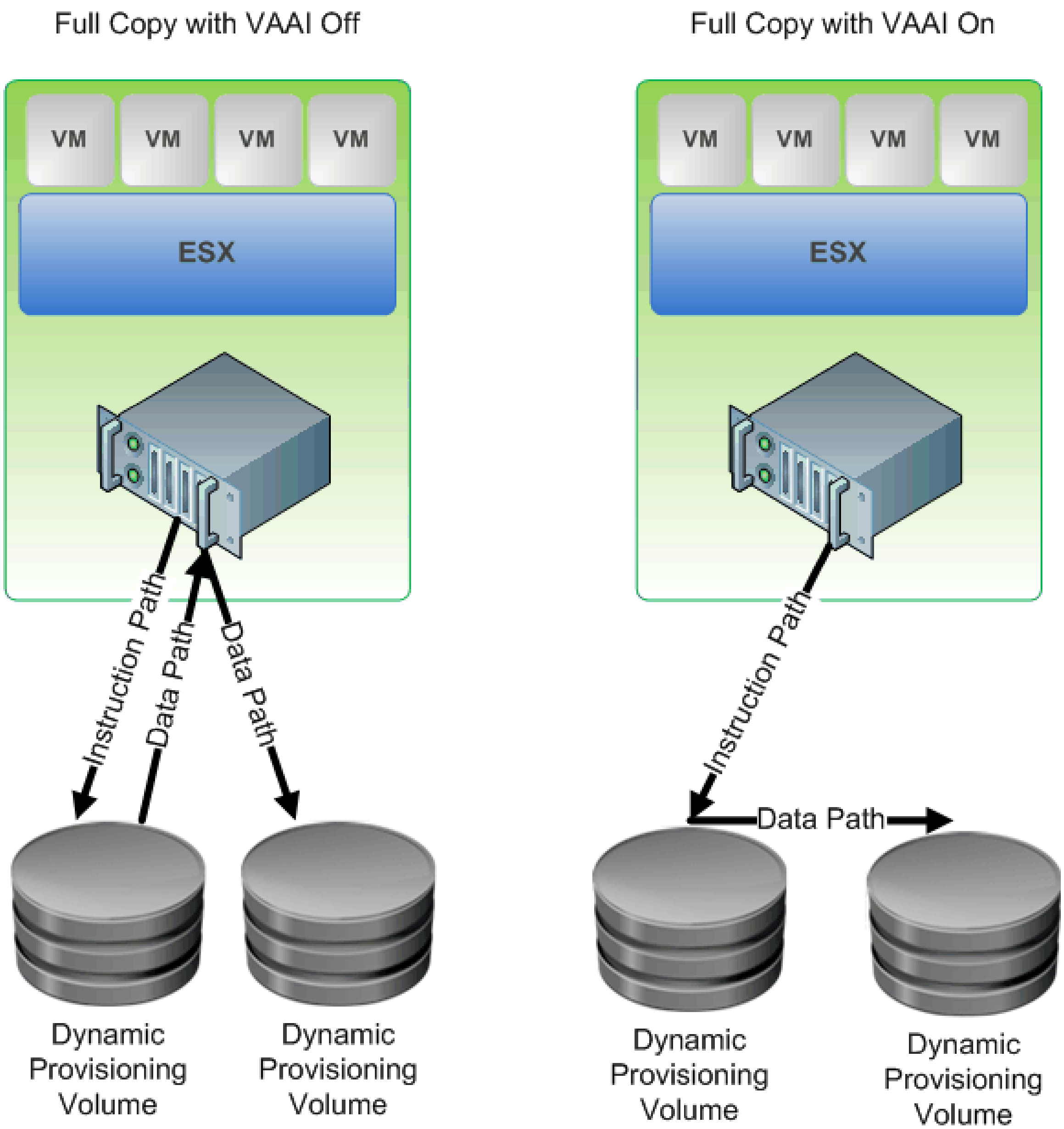

Figure 1

Write Same (Zero)

ESXi supports three different space allocation options when creating new virtual machines or virtual disks. Table 2 describes the three different options for VMDK provisioning.

Table 2. VMDK Provisioning Options

Hardware	*Description*
Thin	Virtual disk is allocated only the storage capacity required by the guest operating system. As write operations occur, additional space is allocated and zeroed. The virtual disk grows to the maximum allotted size.
Lazyzeroedthick (Zeroedthick)	The virtual disk storage capacity is pre-allocated at creation. The virtual disk does not grow in size. However, the space allocated is not "pre-zeroed." As the guest operating system writes to the virtual disk, the space is zeroed as needed.
Eagerzeroedthick	The virtual disk storage capacity is pre-allocated at creation. The virtual disk does not grow in size. The virtual disk is "pre- zeroed." As the guest operating system writes to the virtual disk, the space does not need to be zeroed.

Figure 2 compares block zeroing with and without VAAI. The block zeroing primitive allows Lazyzeroedthick or eagerzeroedthick to quickly provision VMDKs by writing zeros across hundreds or thousands of blocks on the VMFS datastores. This primitive is particularly useful when provisioning eagerzeroedthick VMDKs for VMware fault tolerant virtual machines due to the large number of blocks that need to be zeroed.

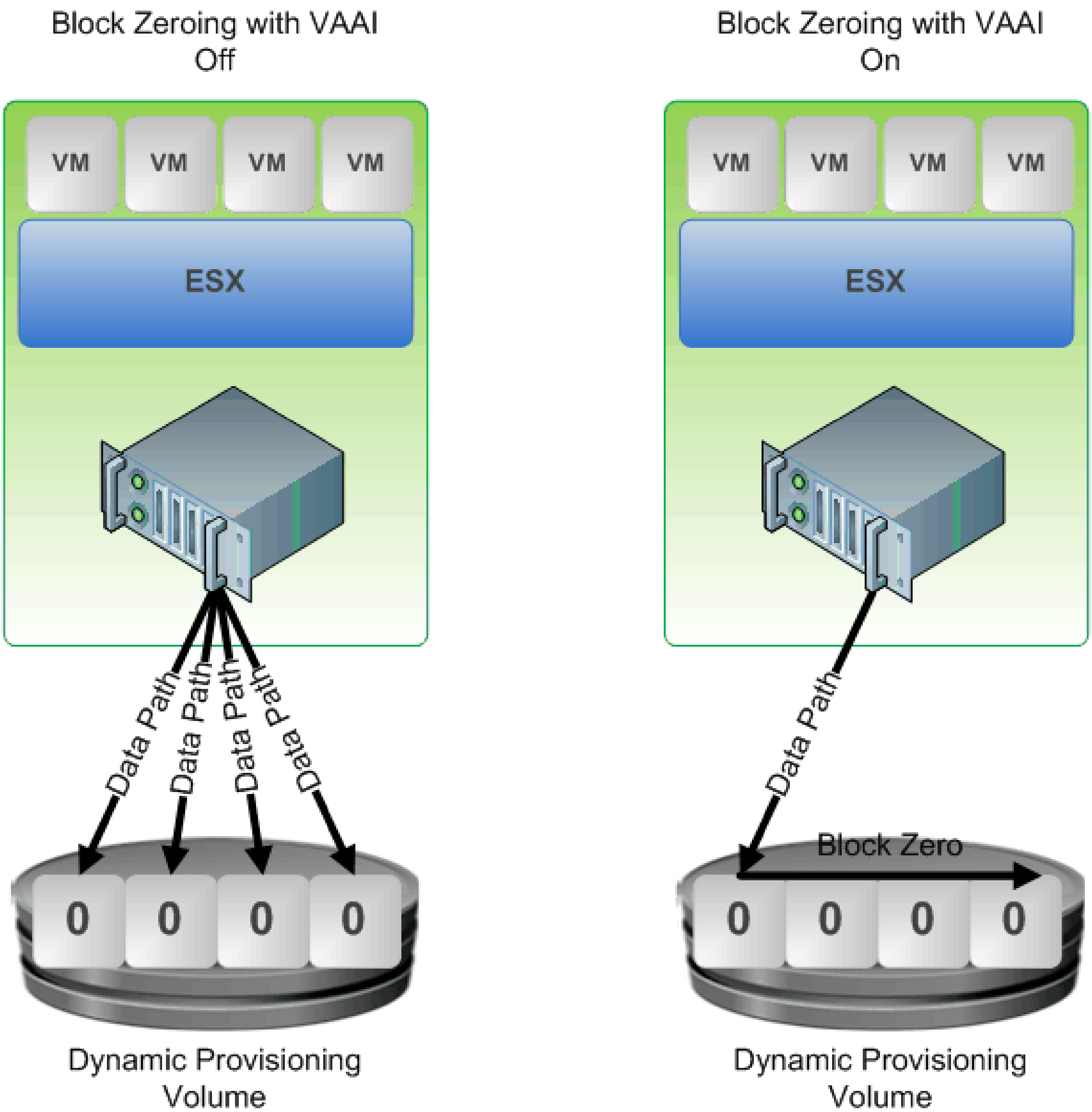

Figure 2

Atomic Test & Set (ATS)

ESXi 5.5 environments rely on locking mechanisms to protect VMFS metadata, particularly in clustered environments where multiple ESXi hosts access the same LUN. Hardware-assisted locking provides a granular LUN locking method to allow locking at the logical block address level without the use of SCSI reservations or the need to lock the entire LUN from other hosts.

Without hardware-assisted locking, ESXi uses SCSI reservations to prevent hosts from activating or sharing virtual disk content on more than one host at the same time. These SCSI locking algorithms lock an entire LUN and do not provide the granularity to lock a particular block on the LUN. In addition, this algorithm requires four separate commands to acquire a lock (simplified as *reserve*, *read*, *write*, and *release*).

In addition to lack of granularity, locking the entire LUN introduces SCSI reservation contention. Both of these affect scalability.

Figure 3 shows a comparison of hardware-assisted locking with and without VAAI.

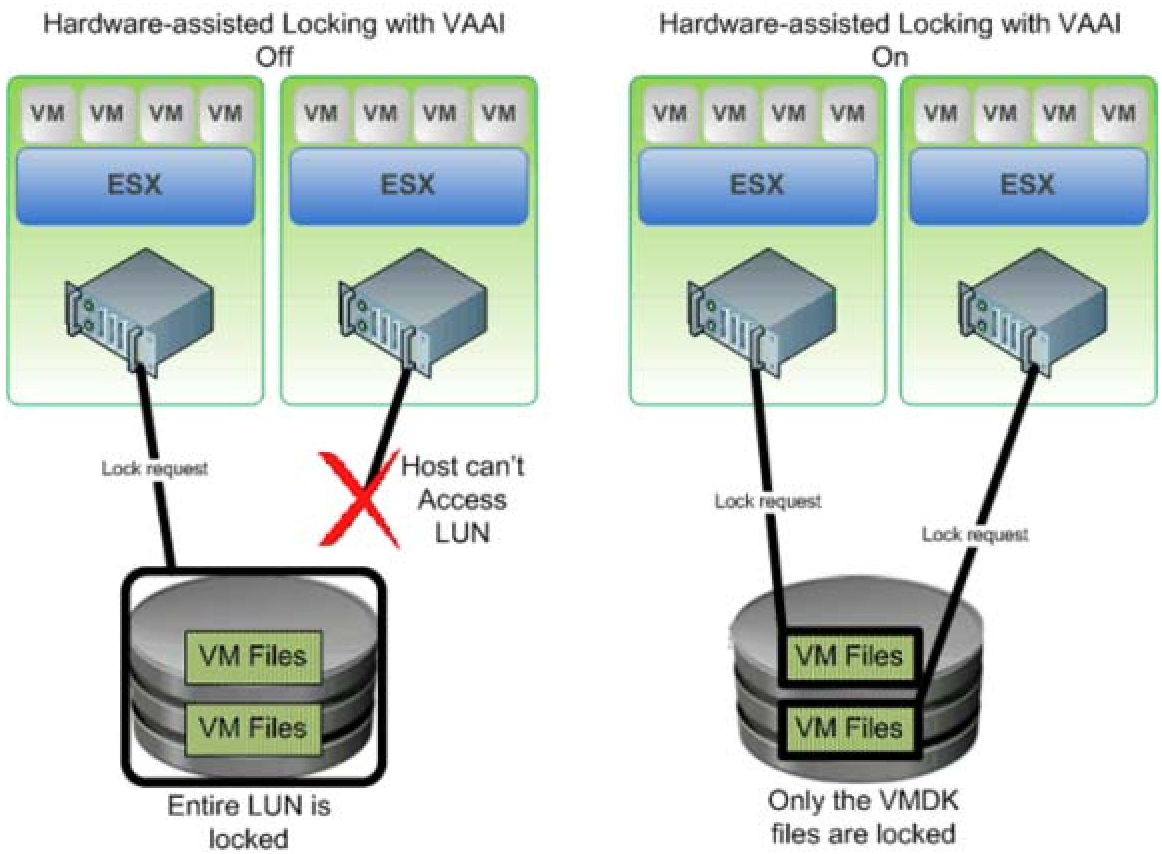

Figure 3

Transferring the LUN locking process to Virtual Storage Platform G1000 reduces the number of commands required to access a lock and allows more granular locking. This leads to better overall performance and increases the number of virtual machines per datastore as well as the number of hosts accessing the datastore.

The following are example use cases for hardware-assisted locking:

- Migrating a virtual machine with vMotion
- Creating a new virtual machine or template
- Deploying a virtual machine from a template
- Powering a virtual machine on or off
- Creating, deleting, or growing a file
- Creating, deleting, or growing a snapshot

Thin Provisioning-STUN (TP-STUN)

Thin Provision Stun (TP Stun) was introduced to address concerns regarding the impact on virtual machines when thin-provisioned datastore usage reaches 100 percent of capacity. Previously, this affected all virtual machines running on the datastore. With the release of the vSphere 5.0 VAAI Thin Provisioning primitives, if a thin-provisioned datastore becomes 100 percent utilized, only those virtual machines requiring extra blocks of storage space are paused; those not needing additional space continue to run. After additional space is allocated to the thin-provisioned datastore, the paused virtual machines can be resumed.

By pausing the virtual machine when the requested additional disk space is not available, virtual machine corruption can be avoided and resuming the virtual machine is as simple as either adding disk space to the thin provisioned datastore or moving virtual machines off of the at-capacity datastore to a datastore with remaining excess capacity.

Dead Space Reclamation (UNMAP)

A new VAAI primitive, using the SCSI UNMAP command, enables an ESXi host to inform the storage array that space can be reclaimed that previously had been occupied by a virtual machine that has been migrated to another datastore or deleted. This enables the storage array to accurately report space consumption of a thin-provisioned datastore.

The mechanism for conducting a space reclamation operation has changed since the primitive was introduced in vSphere 5.0. Originally, the operation was automatic. When a virtual machine was migrated from a datastore or deleted, the UNMAP was called immediately and space was reclaimed on the array. There were issues with this approach regarding performance on the ESXi host and the storage array. For this reason, the UNMAP operation is now a manual process that can be executed from an esxcli command. This allows the UNMAP process to be run during non-peak hours and maintenance windows.

Engineering Validation

To demonstrate the VMware VAAI capabilities of Virtual Storage Platform G1000, Hitachi Data Systems configured a vSphere environment and tested it using the following use cases:

- **Provisioning VMDK time Improvement** — Create eagerzeroedthick VMDK files on datastores using block zeroing.
- **UNMAP** — Recover space on the storage array using *UNMAP*
- **Warm-up time improvement** — Improve warm-up time of thin provisioned Virtual Machine using Block Zeroing
- **VM Cloning time improvement** — Clone Virtual Machines using Full Copy
- **Boot Storm —** Improve Virtual Machine boot times using Hardware-assisted Locking
- **Thin Provisioning-STUN (TP-STUN) -** Prevent virtual machine corruption and guest OS system crashes by pausing a virtual machine when the underlying storage system is depleted and growing of the virtual machine VMDKs are no longer possible.

The goal of these tests was to compare times and I/O performance with VAAI on and off. The test results report IOPS, response time, and total completion times. VAAI was enabled and disabled by logging into the vCenter client and enabling or disabling the VAAI primitives through advanced features.

Note — All testing was done in a lab environment. In production environments, results can be affected by many factors that cannot be predicted or duplicated in a lab. Conduct proof-of-concept testing using your target applications in a non-production, isolated test environment that is identical to your production environment. Following this recommended practice allows you to obtain results closest to what you can expect to experience in your deployment. The test results included in this document are not intended to demonstrate actual performance capability of Virtual Storage Platform G1000.

Test Environment

The environment for these tests consisted of one or two VMware ESXi 5.5 hosts attached to a Virtual Storage Platform G1000 using internal disks. All of the ESXi hosts used redundant paths for both the HBAs and the NICs. The host configuration followed VMware recommended practices. For more information, see Optimizing the Hitachi Virtual Storage Platform in vSphere 4 Environments.

Table 3 lists the hardware used in the Hitachi Data Systems lab.

Table 3. Hardware Resources

Hardware	Description	Version
Hitachi Virtual Storage Platform G1000	■ 4 × 8 GB Fibre Channel ports ■ 1TB cache memory ■ 38 × 300 GB 10k SAS ■ 26 × 146 GB 15K SAS	Microcode: 80-01-20-00/01 SVP: 80-01-20/01 RIM server: 08_01_01
Hitachi Compute Blade 500 Chassis	■ 8-blade server chassis ■ 2 Brocade 5460 Fibre Channel switch modules, each with 6 × 8 Gb/sec uplink ports ■ 2 Pass-through Ethernet modules ■ 2 Management modules ■ 6 Cooling fan modules ■ 4 Power supply modules	SVP: A0165-E-8205 5460: FOS v7.0.2c
Hitachi 520HB2 Server Blades	■ Half blade ■ 2 × 12-core Intel Xeon E5-2697 v2 processor, 2.70 GHz ■ 192 GB RAM ■ 16 × 16 DIMMs	BMC/EFI: 04-15
Brocade 6510 switch	■ SAN switch with 48 x 8 Gb Fibre Channel ports	FOS: 7.2.0
Cisco Nexus 5548 Switch	■ Ethernet switch with 24 × 10 Gb/sec ports	5.2 (1) N1 (5)

Microsoft® Windows® 2008 R2 Enterprise was installed on the virtual machines used in the Hitachi Data Systems lab for this test environment. Each virtual machine was configured with one virtual CPU and 1 GB of RAM.

A standalone server running Windows 2008 R2 Enterprise with quad-core CPUs and 2 GB RAM was used to host the VMware vCenter server and VMware vCenter client.

Test Methodology

To measure the duration of each test case, the VMware vSphere client captured the requested start time and the end time of each task to determine how long each task required to complete. Each test was performed four times to validate the results, with the average value reported.

The tests that required disk I/O to be measured used a custom script to simultaneously launch VDbench 5.02 across all of the virtual machines. VDbench repeatedly created and deleted a 500 GB file on each virtual machine to generate a consistent level of VMFS metadata operations and disk I/O traffic.

To measure the number of host I/Os and SCSI conflicts being generated on each host, ESXTOP was used. At the end of each test, esxtop logs were collected from all the ESXi hosts and then averaged for each host.

Test Results

The following test results show the benefits of using the built in VAAI support in Virtual Storage Platform G1000. The tests show how Virtual Storage Platform G1000 improves performance and scalability when using the VAAI primitives. Each test consists of everyday administrative tasks that can put extra stress on a VMware vSphere environment that result in performance slowdowns.

Provisioning VMDK Time Improvement

This test is designed to test the block zeroing primitive of VAAI. For this test five new 100 GB eagerzeroedthick VMDK were created. The time it took to complete the provisioning was measured.

Table 4 shows the storage configuration used on Virtual Storage Platform G1000 for this test.

Table 4. Internal Virtual Storage Platform G1000 Configuration for Provisioning VMDK Time Improvement

DP-VOL (LU)	▪ Capacity (GB) - 501.86
PG (Parity Group)	▪ Drive Type - SAS ▪ Rotation - 10k ▪ Drive Size - 900 GB ▪ 1 Parity Group - RAID-5 (3D+1P)

Figure 4 shows the creation time of virtual machines was improved 96 percent with VAAI enabled versus VAAI disabled.

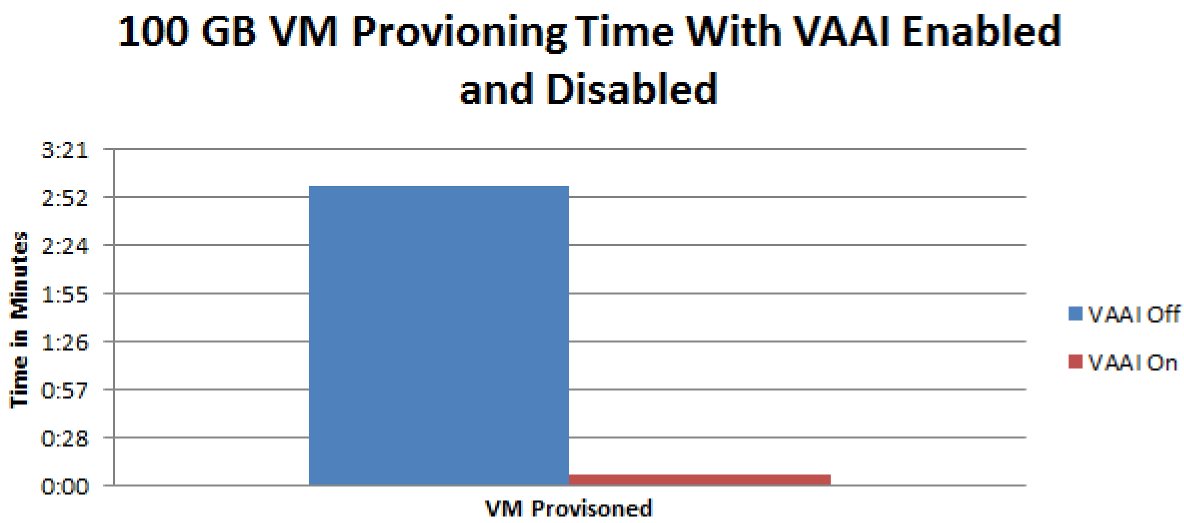

Figure 4

Key Finding — Because the full copy primitive uses the storage system to execute commands, the ESXi hosts are less burdened with respective copy commands. This allows the host more cycles for processing other tasks.

UNMAP With Eagerzerothick provisioning

For the first test of the Dead Space Reclamation (UNMAP) primitive of VAAI a virtual machine with a 100 GB eagerzeroedthick VMDK was created. The VM was then deleted from the datastore. The UNMAP ESXCLI command was run on the VMFS volume where the virtual machine was. Storage capacity on the Virtual Storage Platform G1000 virtual volume was measured before the virtual machine was created, after it was created, after it was deleted, and lastly after the esxcli UNMAP command was run. The virtual machine was provisioned with all other VAAI primitives disabled on the ESXi host.

The ESXCLI UNMAP command and syntax are described below.

esxcli storage vmfs unmap --volume-label=volume_label|--volume-uuid=volume_uuid --reclaim-unit=number

- **-l | --volume-label=volume_labelb -** The label of the VMFS volume to UNMAP. This is a mandatory argument. If you specify this argument, do not use -u|--volume-uuid=volume_uuid.
- **-u|--volume-uuid=volume_uuid -** The UUID of the VMFS volume to UNMAP. This is a mandatory argument. If you specify this argument, do not use -l|--volume-label=volume_label.
- -**n|--reclaim-unit=number -** The number of VMFS blocks to UNMAP per iteration. This is an optional argument. If it is not specified, the command uses a default value of 200.

Table 5 shows the configuration used on Virtual Storage Platform G1000 for this test.

Table 5. Virtual Storage Platform G1000 Configuration for UNMAP with Eagerzerothick Test

DP-VOL (LU)	▪ Capacity (GB) - 1019.89 GB
PG (Parity Group)	▪ Drive Type - SAS ▪ Rotation - 10k ▪ Drive Size - 900 GB ▪ 1 Parity group - RAID-5 (3D+1P)

UNMAP enables the storage system hosting both lazyzeroedthick and eagerzeroedthick VMDKs in VMFS datastores to reclaim unused blocks that were previously used by the VMFS file system.

Figure 5 contrasts the space used on the ESXi's VMFS file system and the Virtual Volume on Virtual Storage Platform G1000.

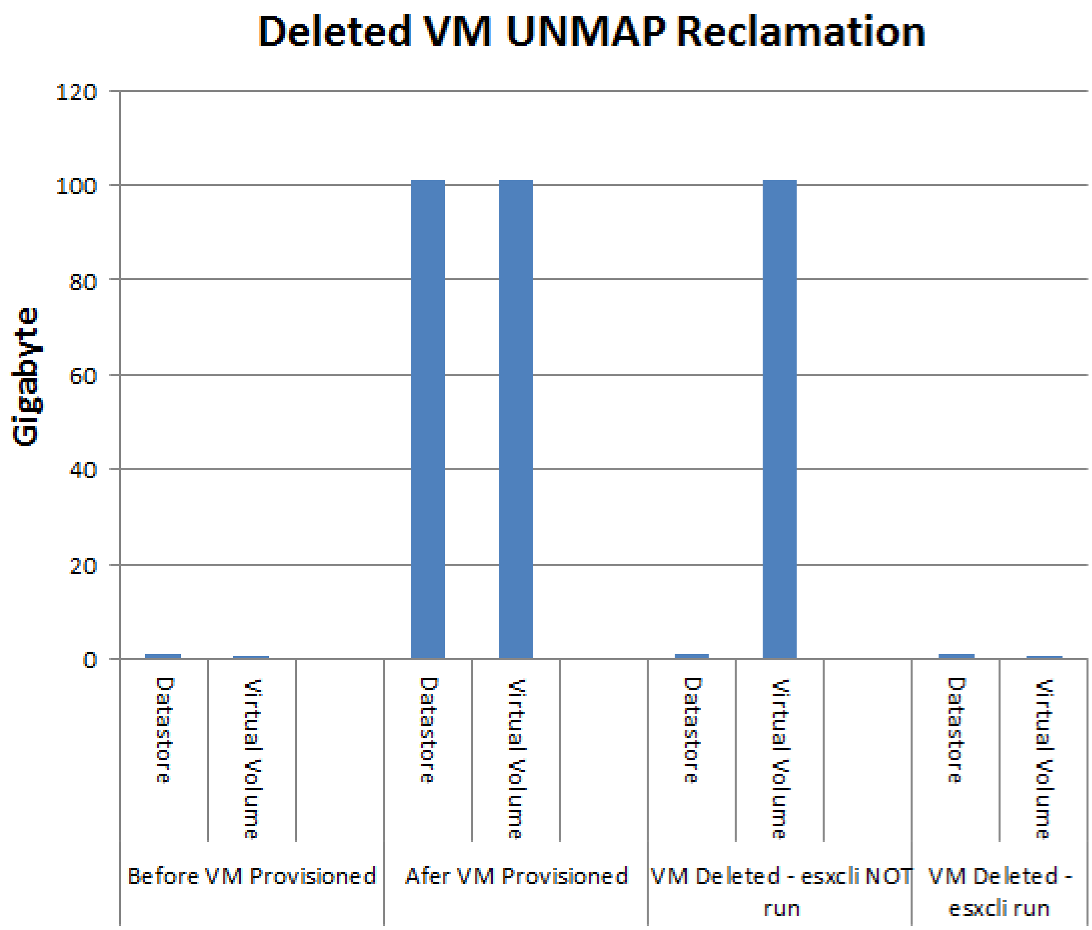

Figure 5

Testing of UNMAP where the virtual machine was storage vMotioned versus deleted were virtually identical.

Key Finding — The UNMAP primitive of VAAI works well on Virtual Storage Platform G1000 when working with eagerzerothick VMDKs. This will allow for better storage utilization as virtual machines are migrated within or deleted from the virtual environment..

Warm-up Time Improvement

This test analyzed the effect of VAAI on the warm-up time of thin provisioning eagerzeroedthick, lazyzerothick and thin provisioned VMDKs using Hitachi Dynamic Provisioning with Virtual Storage Platform G1000. A 100 GB virtual machine of each virtual disk type was booted and VDbench load generation tool was used to create a 400 IOPS workload. The time it took for the virtual machines to warm-up was captured.

A virtual machine was considered warmed-up when the IOPS on the VMFS file system reached a steady state. Because the eagerzeroedthick, lazyzeroedthick, and thin provisioned VMDKs were provisioned using the same storage configuration, the IOPS will be nearly identical when warm-up is complete.

Table 6 shows the storage configuration used on Virtual Storage Platform G1000 for these tests.

Table 6. Virtual Storage Platform G1000 Storage Configuration for the Warm-up Time Improvement Test

DP-VOL (LU)	▪ Capacity (GB) - 1019.89 GB
PG (Parity Group)	▪ Drive Type - SAS ▪ Rotation - 10k ▪ Drive Size - 900 GB ▪ 1 Parity group - RAID-5 (3D+1P)

Testing showed there is no difference in I/O or latency performance when using eagerzerothick VMDKs with VAAI enabled or disabled. This is due to the fact that the VMDK is full provisioned and zeroed. Therefore, there is no need to grow the VMDK or zero the blocks, and no work for the VAAI primitives to perform.

Figure 6 shows the IOPS for the VMFS file system during the testing of a lazyzerothick VMDK with VAAI enabled and disabled in a dynamic provisioning volume.

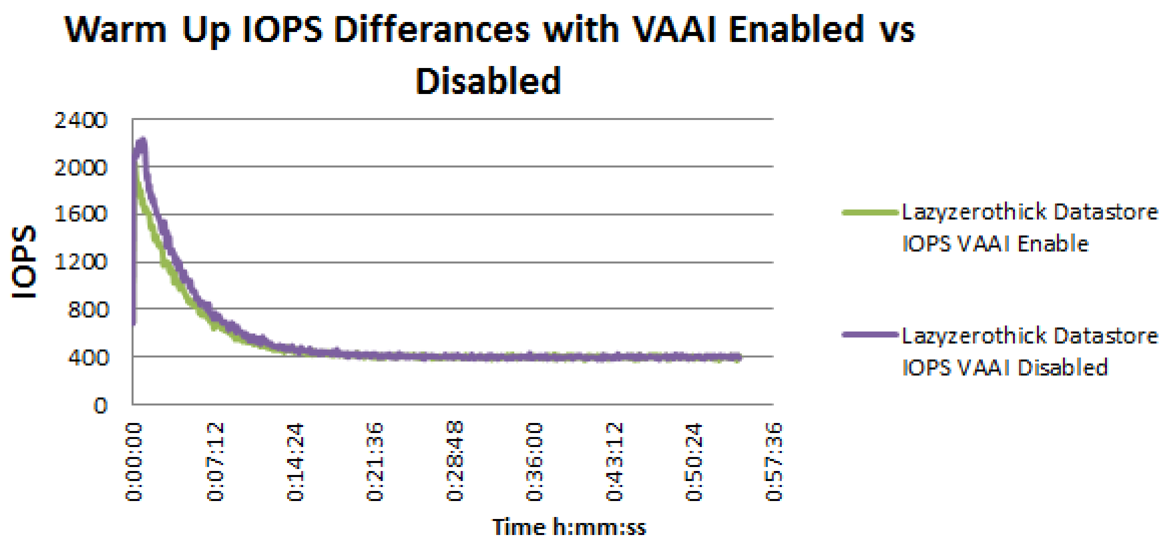

Figure 6

Figure 6 shows IOPS for lazyzerothick VMDK with VAAI enabled and disabled. It is shown that with VAAI disabled the IOPS start out higher, by about 200 IOPS, or 20%. This is due to blocks needing to be zeroed the first time they are written to when using lazyzeroedthick VMDKs. VAAI is able to off-load this workload to the storage array.

Figure 7 shows the IOPS for the VMFS file system during the testing of thin provisioned VMDK with VAAI enabled and disabled in a dynamic provisioning volume.

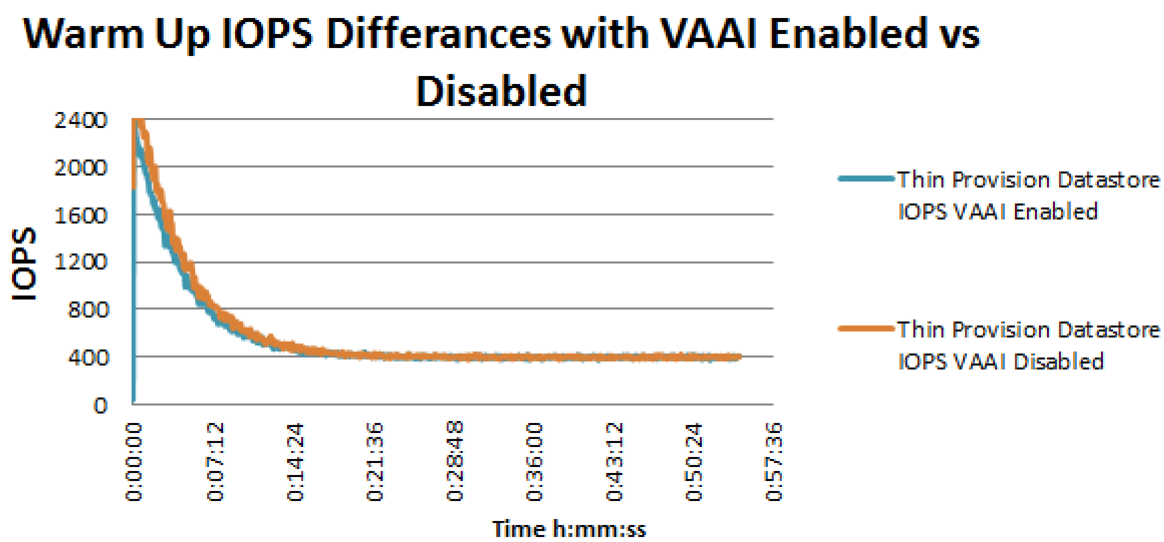

Figure 7

Figure 7 shows IOPS for thin provisioned VMDK with VAAI enabled and disabled. It is shown that with VAAI disabled the IOPS start out higher, by about 200 IOPS, or 20%. This is due to blocks needing to be zeroed the first time they are written to when using thin provisioned VMDKs. VAAI is able to offload this workload to the storage array.

When comparing Figure 6 IOPS on a lazyzerothick VMDK with VAAI enabled to Figure 7 IOPS for the thin provisioned VMDK with VAAI enabled, it shows that the thin provisioned VMDK created about 400 more IOPS in the beginning due to the need to grow the VMDK.

Key Finding — VAAI improved the performance of the virtual disk warm up time for lazyzerothick and thin provisioned VMDKs. VAAI helps to decrease the amount of I/O sent to the storage array from the ESXi host. Eagerzerothick VMDKs did not see a change in performance as the VMDKs were already zeroed and did not need to grow.

VM Cloning Time Improvement

This test is designed to test the block zeroing and full copy primitives of VAAI. For this test a 100 GB eagerzeroedthick virtual machine was cloned. During the testing, IOPS and the cloning time to completion were captured.

Table 7 shows the storage configuration used on Virtual Storage Platform G1000 for this test.

Table 7. Virtual Storage Platform G1000 Configuration for the VM Cloning Time Improvement Test

DP-VOL (LU)	▪ Capacity (GB) - 1019.89 GB
PG (Parity Group)	▪ Drive Type - SAS ▪ Rotation - 10k ▪ Drive Size - 900 GB ▪ 1 Parity group - RAID-5 (3D+1P)

Figure 8 contrasts the source and destination datastores IOPS when cloning an eagerzeroedthick virtual machine with VAAI enabled.

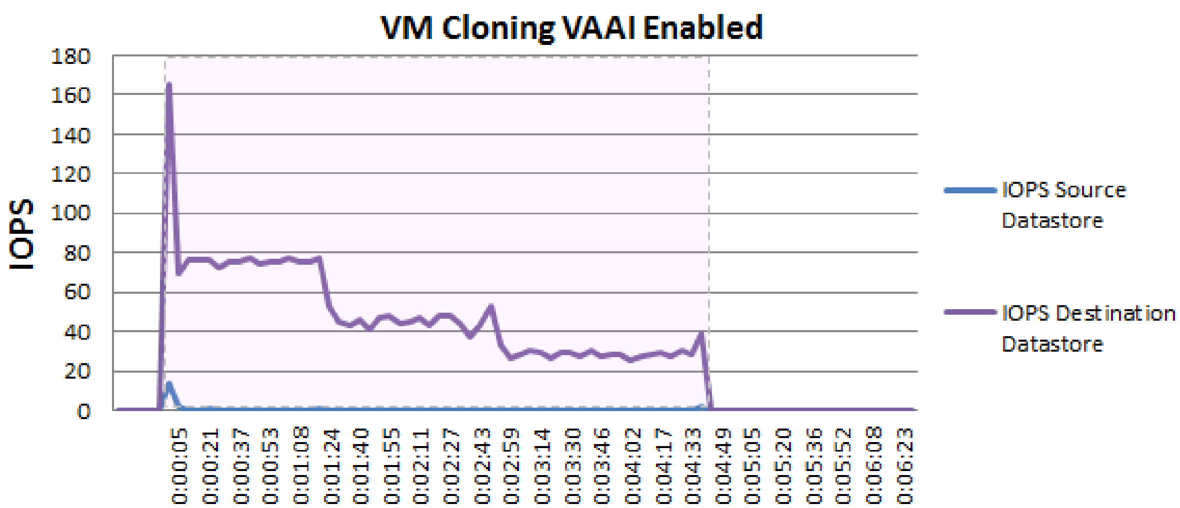

Figure 8

Figure 9 shows that with VAAI enabled, the cloning process produces minimal read or write I/O on the source or destination from the ESXi host as the cloning process was offloaded to the storage array.

Figure 10 contrasts the source and destination datastores read and write IOPS when cloning an eagerzerodthick virtual machine with VAAI disabled.

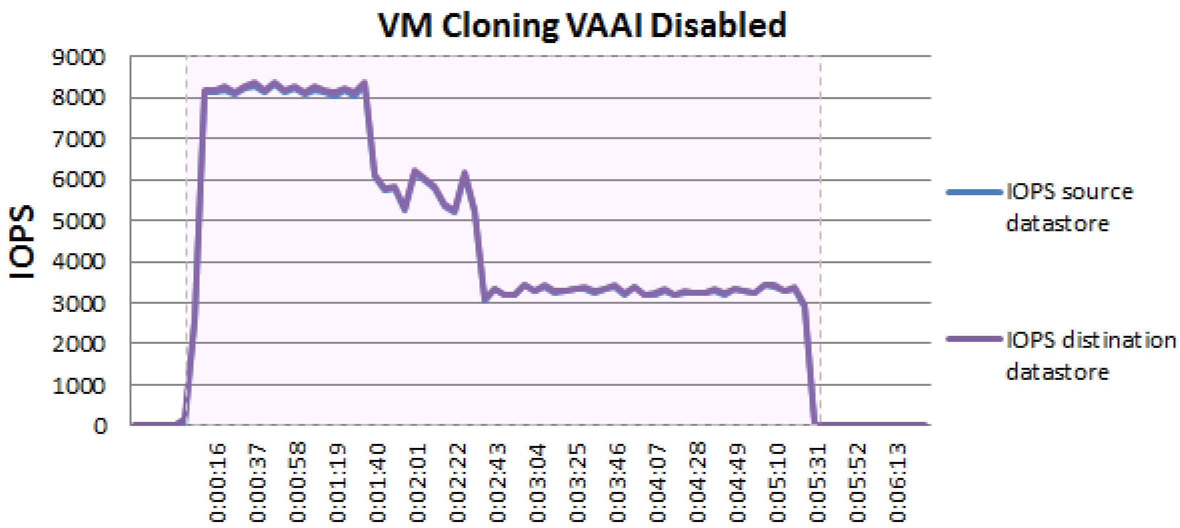

Figure 9

Figure 8 showed a maximum IOPS of about 160 with VAAI enabled. Figure 9 showed a maximum IOPS of about 8000 with VAAI disabled. There was a significant decrease in IOPS when VAAI was enabled due to the workload being offloaded to the storage array.

Figure 10 shows the time savings of deploying a 100 GB virtual machine with VAAI enabled vs. deploying a 100 GB virtual machine with VAAI disabled.

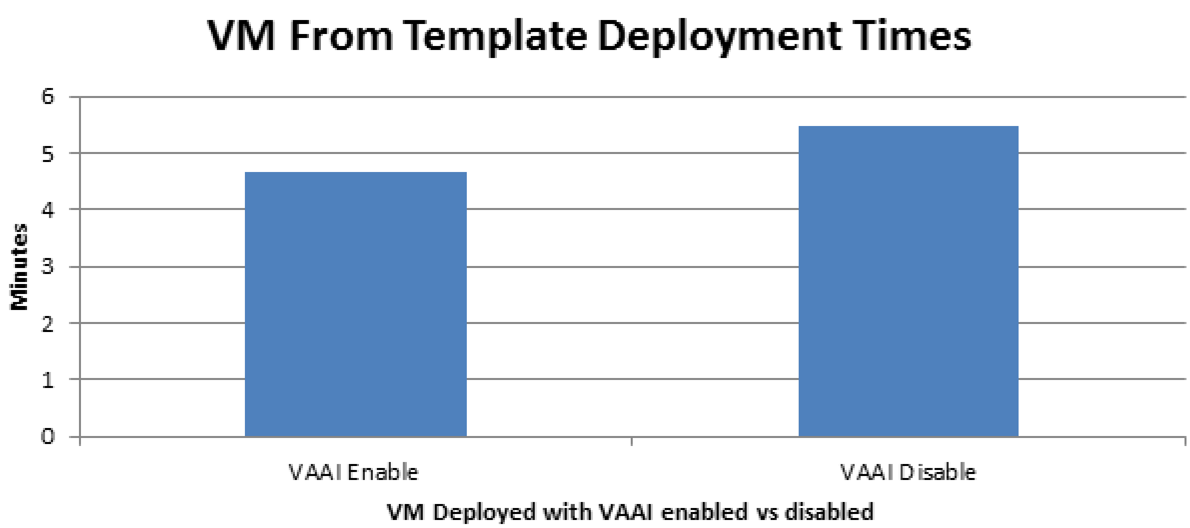

Figure 10

Figure 10 shows a 15% time savings when deploying a 100 GB virtual machine from a template with VAAI enabled vs disabled.

Key Finding — The VAAI XCOPY primitive is beneficial in off-loading virtual machine cloning operations to the storage array from the ESXi host. This frees resources from the ESXi host for other workloads. This also reduces traffic on the SAN fabric, again freeing resources for other workloads.

Boot Storm

This tested the hardware assisted locking primitive of VAAI. In this test case, linked clone virtual machines were evenly distributed across two ESXi 5.5 hosts using a single shared datastore.

This was created on an LDEV provisioned from a four spindle Dynamic Provisioning pool. For this testing all virtual machines were powered on simultaneously.

Table 8 shows the storage configuration used on Virtual Storage Platform G1000.

Table 8. Virtual Storage Platform G1000 Configuration for the Boot Storm Test

DP-VOL (LU)	▪ Capacity (GB) - 1019.89 GB
PG (Parity Group)	▪ Drive Type - SAS ▪ Rotation - 10k ▪ Drive Size - 900 GB ▪ 1 Parity group - RAID-5 (3D+1P)

With the storage using VAAI, boot times decreased by 49.75 percent. Figure 11 shows the boot times for all 128 virtual machines.

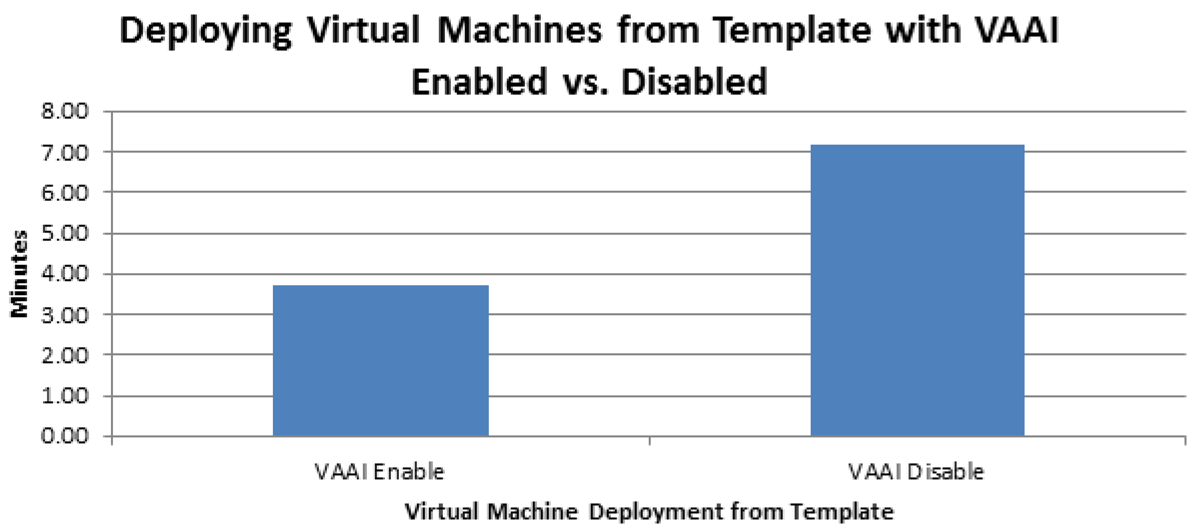

Figure 11

Figure 11 shows the decrease in the average number of SCSI conflicts per virtual machine when powering on 128 virtual machines during a boot storm with VAAI enabled vs. powering on 128 virtual machines during a boot storm with VAAI disabled.

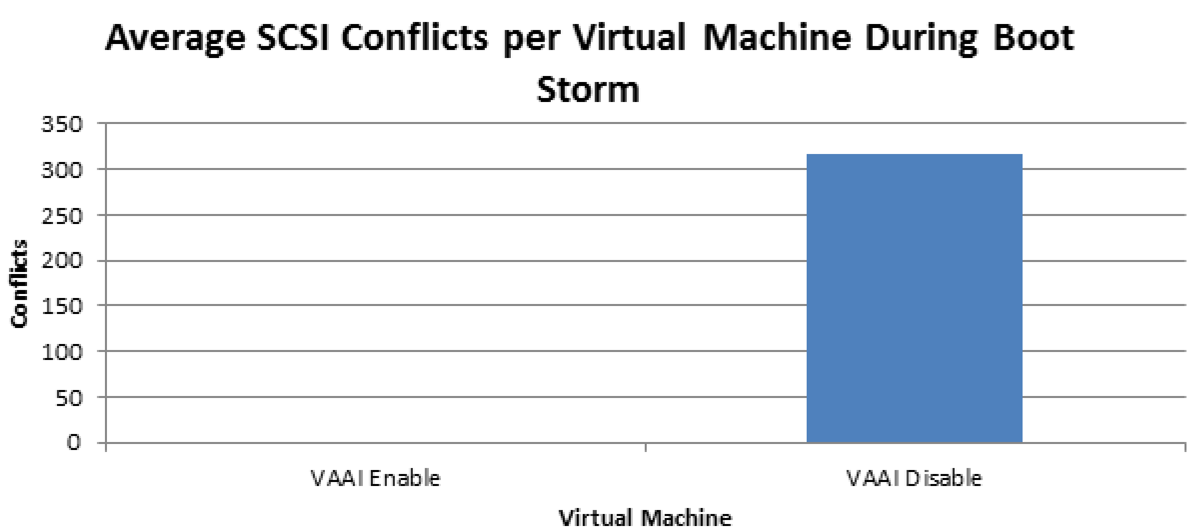

Figure 12

Figure 12 shows that during a boot storm, with VAAI enabled there were no SCSI conflicts as the SCSI locking was offloaded to the storage array. During the boot storm, with VAAI disabled there were an average of 317 SCSI conflicts per virtual machine, which increased the boot time for the 128 virtual machines during the boot storm.

Key Finding — The hardware-assisted locking primitive improved virtual machine boot times significantly by eliminating SCSI lock conflicts from the ESXi hosts. This benefit will be useful in both a server environment and especially in a VDI environment where boot storms are more common.

Thin Provision Stun (TP Stun)

With VAAI on and with storage becoming depleted on the SAN, alerts from the vSphere web client, seen in Figure 13, can be sent to the administrator using email or console logging.

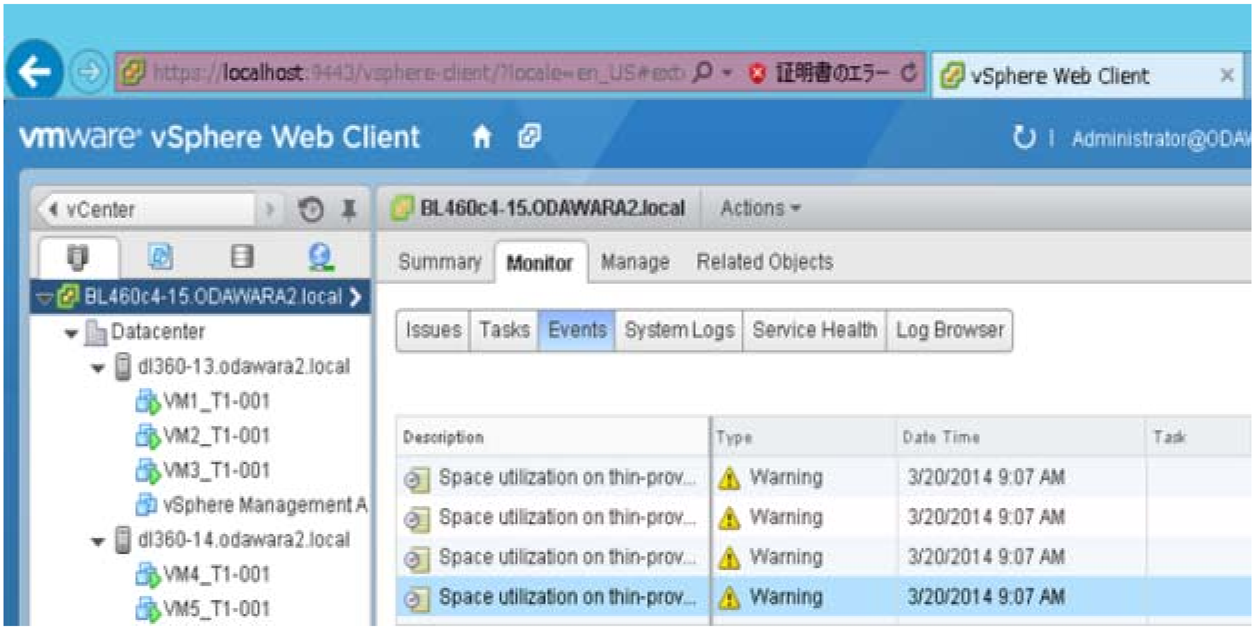

Figure 13

When conducting the thin provisioning stun API primitive testing, an over provisioned Hitachi Dynamic Provisioning pool filled to capacity sends an alert first to the administrative console. When the consumed capacity overtakes the alert threshold, the virtual machine displays the dialog box in Figure 14 suggesting that you free disk space and retry.

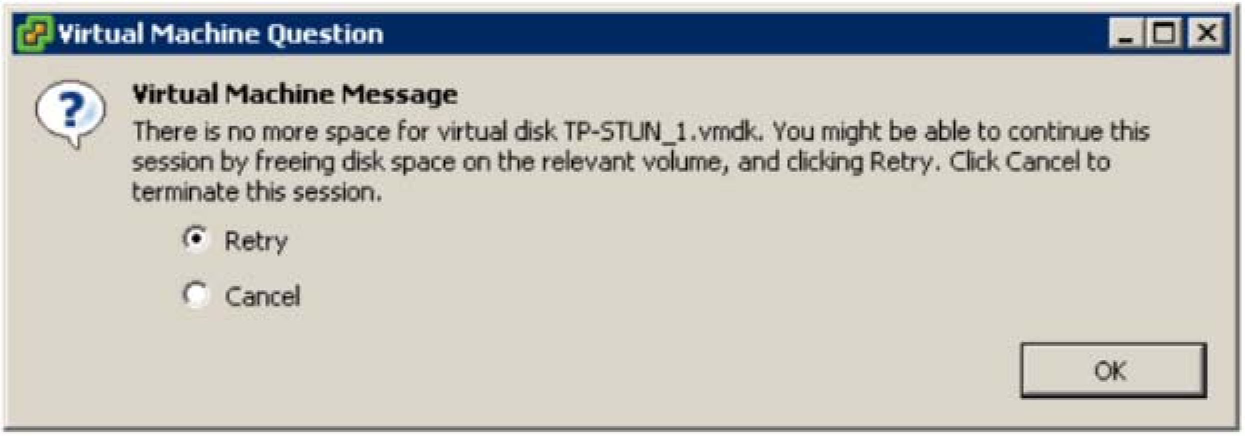

Figure 14

If the space condition is resolved, then clicking the Retry option results in the resumption of all I/O writes and reads within the given virtual machine.

Key Finding — Thin Provision Stun is useful in environments where disk space is over provisioned to help ensure that virtual machines do not become corrupted and guest operating systems do not crash when disk space is depleted.

Conclusion

Virtual Storage Platform G1000, when coupled with VMware VAAI primitives, allows you to build and maintain more scalable and efficient virtual environments. When you use these primitives with Virtual Storage Platform G1000 and Hitachi Dynamic Provisioning, you create a robust and highly available infrastructure to support high density virtual machine workloads.

The full copy primitive reduces host-side I/O during common tasks such as the following:

- Moving virtual machines with Storage vMotion
- Deploying a new virtual machine from a template by instructing the storage system to copy data within Virtual Storage Platform G1000, rather than sending the traffic back and forth through the ESXi host's host bus adapter
- The block zeroing primitive speeds virtual machine deployment by offloading the repetitive zeroing of large numbers of blocks to Virtual Storage Platform G1000. This frees the ESXi host resources for other tasks.
- The hardware-assisted locking primitive greatly reduces the probability of ESXi hosts being locked out when attempting to access files on a VMFS datastore. A lock out can degrade performance, and in some cases, cause tasks to time out or fail completely.
- Thin Provisioning-Stun (TP-Stun) primitive helps ensure the integrity of the virtualized environment by pausing virtual machines when the underlying storage is full.

The tight integration of VMware’s VAAI and the Virtual Storage Platform G1000 provides a proven, high-performance, highly scalable storage solution for an ESXi environment.

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.

Hitachi Data Systems Corporation

Corporate Headquarters
2845 Lafayette Street, Santa Clara, California 95050-2627 USA
www.HDS.com

Regional Contact Information
Americas: +1 408 970 1000 or info@HDS.com
Europe, Middle East and Africa: +44 (0) 1753 618000 or info.emea@HDS.com
Asia-Pacific: +852 3189 7900 or hds.marketing.apac@HDS.com

AS-294-00, April 2014