

Advantages of Using VMware VAAI with the Hitachi Adaptable Modular Storage 2000 Family

Lab Validation Report

By Henry Chu and Roger Clark

February 2011

Feedback

Hitachi Data Systems welcomes your feedback. Please share your thoughts by sending an email message to SolutionLab@hds.com. Be sure to include the title of this white paper in your email message.

Table of Contents

| | |
|------------------------------------|-----------|
| VAAI Overview | 3 |
| Full Copy | 4 |
| Block Zeroing | 5 |
| Hardware-assisted Locking | 6 |
| Engineering Validation..... | 7 |
| Test Environment | 7 |
| Test Methodology | 9 |
| Test Results..... | 9 |
| Conclusion | 17 |

Advantages of Using VMware VAAI with the Hitachi Adaptable Modular Storage 2000 Family

Lab Validation Report

Data center administrators are always looking for ways to improve scalability, performance and efficiency and to reduce administrative overhead and costs, and one way to do this in VMware environments is through VAAI, or VMware vStorage APIs for Array Integration (VAAI).

VAAI is a set of APIs, or primitives, that allow IT organizations to offload processing for certain data-related services to VAAI-supported storage systems, such as the Hitachi Adaptable Modular Storage 2000 family. Doing so can enable significant improvements in virtual machine performance, virtual machine density and availability in vSphere 4.1 environments. Moving these functions to a storage system offers many benefits, but also requires the use of a highly available, scalable, high performance storage system like the Hitachi Adaptable Modular 2000 family.

The Hitachi Adaptable Modular Storage 2000 family works seamlessly with VMware's ESX 4.1 and ESXi 4.1 virtualization software to improve the efficiency of tasks such as Storage vMotion, cloning and provisioning new VMs, all while reducing SCSI reservation locks and increasing scalability. Hitachi Dynamic Provisioning software enables the creation of a storage pool from which capacity can be used as needed to further improve performance, scalability and utilization. In addition, the 2000 family's load balancing active-active symmetric controllers distribute vSphere workloads across all paths, eliminating I/O path thrashing and improving performance.

This white paper documents the benefits of using VAAI with the Hitachi Adaptable Modular Storage 2000 family. It is written for storage administrators, vSphere administrators and application administrators who are charged with managing large, dynamic environments. It assumes familiarity with SAN-based storage systems, VMware vSphere and general IT storage practices.

VAAI Overview

VAAI enables key data operations to be executed at the storage level (for example, on the Hitachi Adaptable Modular Storage 2000 family) rather than at the ESX server layer. Doing so reduces resource utilization and potential bottlenecks on physical servers and enables more consistent server performance and higher virtual machine density.

When used with vSphere 4.1, the 2000 family supports the following API primitives:

- **Full copy** — Enables the storage system to make full copies of data within the storage system without having the ESX host read and write the data.
- **Block zeroing** — Enables storage systems to zero out a large number of blocks to speed provisioning of virtual machines.
- **Hardware-assisted locking** — Provides an alternative means to protect the metadata for VMFS cluster file systems, thereby improving the scalability of large ESX host farms sharing a datastore.

Full Copy

For common ESX administration tasks, the ESX host can use the full copy primitive to offload the actual data copy to a 2000 family storage system. For example, the full copy primitive is helpful for tasks such as provisioning VMs or migrating VMDK file between datastores within a storage system using Storage vMotion. The following operations are some examples of when the full copy primitive is used:

- **VM provisioning** — The source and destination locations are within the same volume. Hitachi integrates with the full copy API to clone VMs or datastores from a template. The constant read and write operation during cloning is offloaded to the storage system. This process dramatically reduces I/O between the ESX nodes and Hitachi storage.
- **Storage vMotion** — The source and destination locations are different volumes within the same storage system. This feature enables VMDK files to be relocated between datastores within a storage system. VMs can be migrated to facilitate load-balancing or planned maintenance without service interruption. By integrating with full copy, host I/O offload for Storage vMotion operations accelerates VM migration times considerably.

Figure 1 shows a comparison of copy functions with and without VAAI.

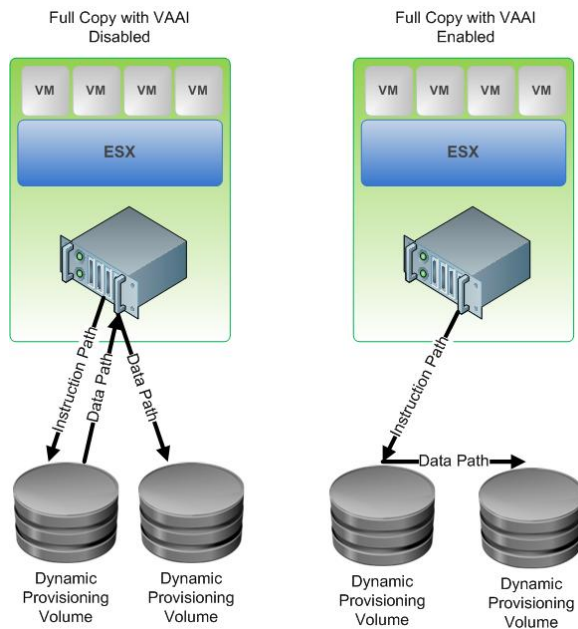


Figure 1

Figure 1 shows that the full copy primitive removes the ESX host from the data path of the VMDK cloning operation. This reduces the number of disk I/Os from the ESX host, saving host-side I/O bandwidth while copying VMs.

Block Zeroing

ESX supports different space allocation options when creating new VMs or virtual disks. When the zeroedthick format is used, the virtual disk's space is pre-allocated but not all pre-zeroed. Instead, the space is zeroed when the guest OS first writes to the virtual disk. When the eagerzeroedthick format is used, the virtual disk's space is pre-allocated and pre-zeroed, meaning that it can take much longer to provision eagerzeroedthick virtual disks. With the block zeroing primitive, these zeroing operations are offloaded to the storage system without the host having to issue multiple commands.

Figure 2 shows a comparison of block zeroing with and without VAAI.

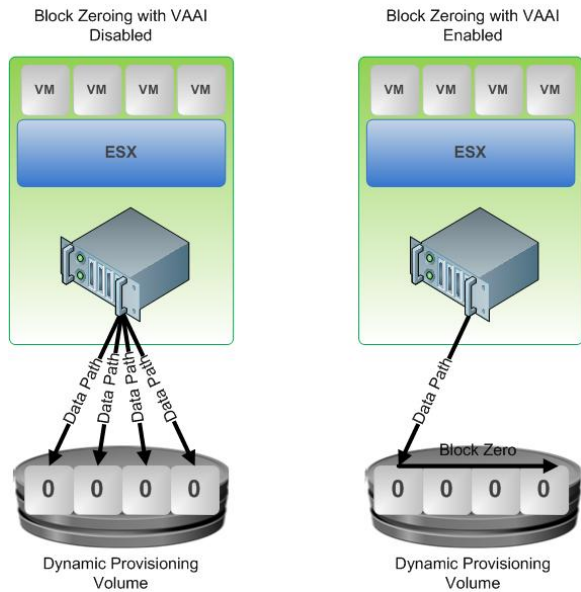


Figure 2

Figure 2 shows that the block zeroing primitive allows zeroedthick or eagerzeroedthick VMDKs to be provisioned quickly by writing zeros across hundreds or thousands of blocks on the VMFS datastores, off-loading much of the process from the ESX hosts to the storage system. This is particularly useful when provisioning eagerzeroedthick VMDKs for VMware Fault Tolerance or VMs for write intensive applications due to the large number of blocks that need to be zeroed.

Hardware-assisted Locking

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.

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

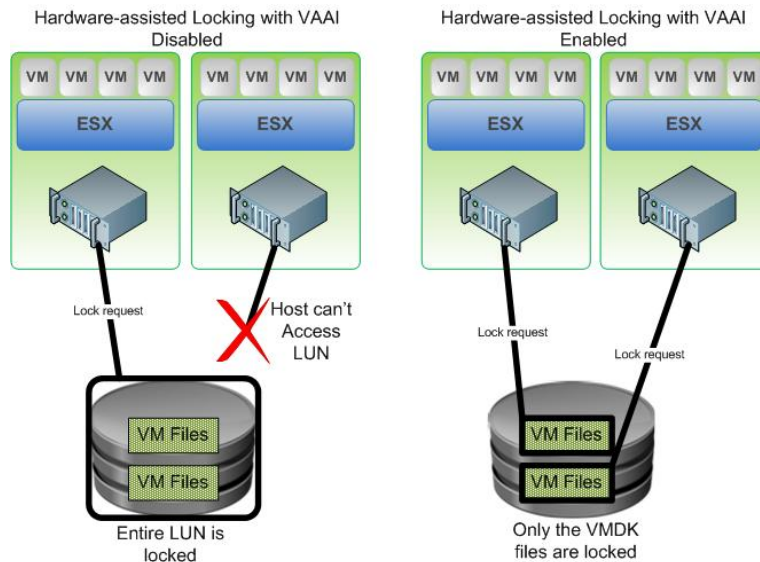


Figure 3

ESX hosts rely on locking mechanisms to protect VMFS metadata, particularly in clustered environments where multiple ESX hosts access the same LUN. ESX uses SCSI reservations to prevent hosts from accessing disk content on multiple hosts at the same time by locking the entire LUN, which can introduce SCSI reservation contention. Newer versions of ESX have made improvements to optimize the locking mechanisms to reduce SCSI reservations. However as an ESX environment grows, more ESX hosts are expected to access the same LUN. Additional optimization is needed to improve scalability for large environments.

Transferring the LUN locking process to a 2000 family storage system 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 VMs per datastore and the number of hosts accessing the datastore.

Following are example use cases for hardware-assisted locking:

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

Engineering Validation

To demonstrate the VAAI capabilities of the Adaptable Modular Storage 2000 family, Hitachi Data Systems configured a vSphere environment and tested it using the following use cases:

- **Cloning with full copy** — Cloning a 30GB VM from one datastore to another
- **VMDK provisioning with block zeroing and zero page reclaim** — Provisioning a new 30GB VMDK file on a VMFS datastore consisting of a single Dynamic Provisioning volume
- **Large scale VM boot storm** — Simultaneously powering on 512 linked clone VMs
- **Large scale simultaneous vMotion** — Using vMotion to move 128 VMs of various types across four ESX hosts

The goal of these tests was to compare times and I/O performance with VAAI both enabled and disabled. The test results report IOPS on each Fibre Channel port, response time and total completion times.

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 best 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 the 2000 family.

Test Environment

The environment for these tests consisted of four VMware ESX 4.1 hosts attached to a Hitachi Adaptable Modular Storage 2100. All of the ESX hosts used redundant paths for both the HBAs and the NICs. The host configuration followed VMware best practices. For more information, see the [Optimizing the Hitachi Adaptable Modular Storage 2000 in vSphere 4 Environments Best Practices Guide](#) white paper.

Table 1 lists the hardware used in the Hitachi Data Systems lab.

Table 1. Hardware Resources

| <i>Hardware</i> | <i>Description</i> | <i>Version</i> |
|---|---|-----------------------|
| Hitachi Adaptable Modular Storage 2100 storage system | Dual controllers 4 x 4GB Fiber Channel ports, 2 per controller 8GB cache memory, 2GB per controller 60 x 300GB, 15K RPM, SAS disks (30 used) | Microcode 0890/H-S |
| Brocade 48000 director | Director-class SAN switch with 4Gb Fibre Channel ports | FOS 5.3.1a |
| Dell R905 server | 4 x Quad-Core AMD Opteron 8347 Processors 1.9GHz, 128GB RAM | ESX4.1 |

Windows 2008 R2 Enterprise was installed on the VMs used in the Hitachi Data Systems lab for this test environment. Each VM was configured with one virtual CPU and 1GB of RAM. A standalone VM running Windows 2008 R2 Enterprise with two virtual CPUs and 4GB RAM was used to host the vCenter Server.

Table 2 lists RAID group configuration details.

Table 2. RAID Group Configuration

| <i>Dynamic Provisioning Pool</i> | <i>RAID Configuration</i> | <i>Number of RAID Groups</i> | <i>Number of Spindles</i> | <i>Dynamic Provisioning Pool Size (TB)</i> |
|----------------------------------|---------------------------|------------------------------|---------------------------|--|
| Pool 1 | RAID-5 (4D+1P) | 5 | 25 | 5.2 |
| Pool 2 | | 1 | 5 | 1.0 |

A single DP-VOL was created within each Dynamic Provisioning pool and was presented to the four ESX hosts.

Figure 4 shows the architecture used for these tests.

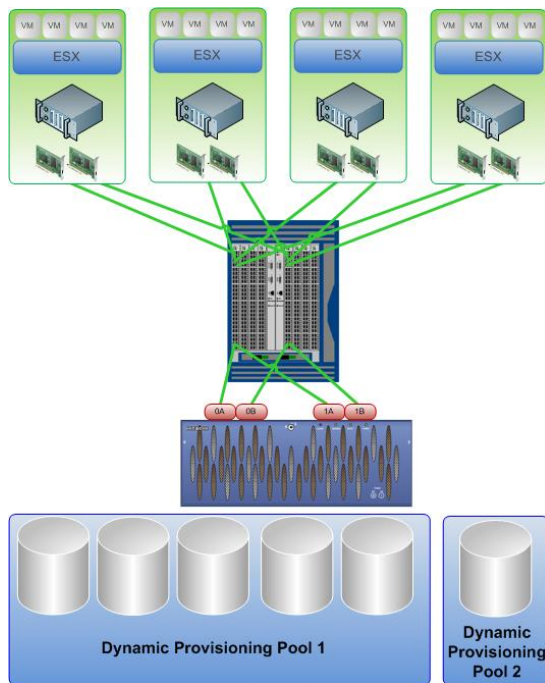


Figure 4

Test Methodology

To measure the duration of each test case, the VMware vSphere client was used to capture 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, and the average value was reported.

To measure the number of SCSI reservation locks for each test case, the VMkernel logs were collected from each host at the conclusion of each test. The logs were parsed to count the number of conflicts recorded.

For the tests that required disk I/O to be measured, a custom script was used to simultaneously launch Vdbench 5.02 across all of the VMs. Vdbench repeatedly created and deleted a 500GB file on each VM to generate a consistent level of VMFS metadata operations and disk I/O traffic..

esxtop was used to measure the number of host I/Os being generated on each host. esxtop logs were collected from all the ESX hosts at the end of each test and then averaged for each host.

Test Results

The following sections describe the results of the testing conducted in the Hitachi Data Systems lab.

Cloning With Full Copy

A 30GB zeroedthick VM and a 30GB eagerzeroedthick VM were each cloned from a source datastore residing on a Dynamic Provisioning pool to a destination datastore on a separate Dynamic Provisioning pool. The test was done with VAAI enabled and disabled.

Figure 5 shows the time it took to complete the cloning with VAAI enabled and disabled.

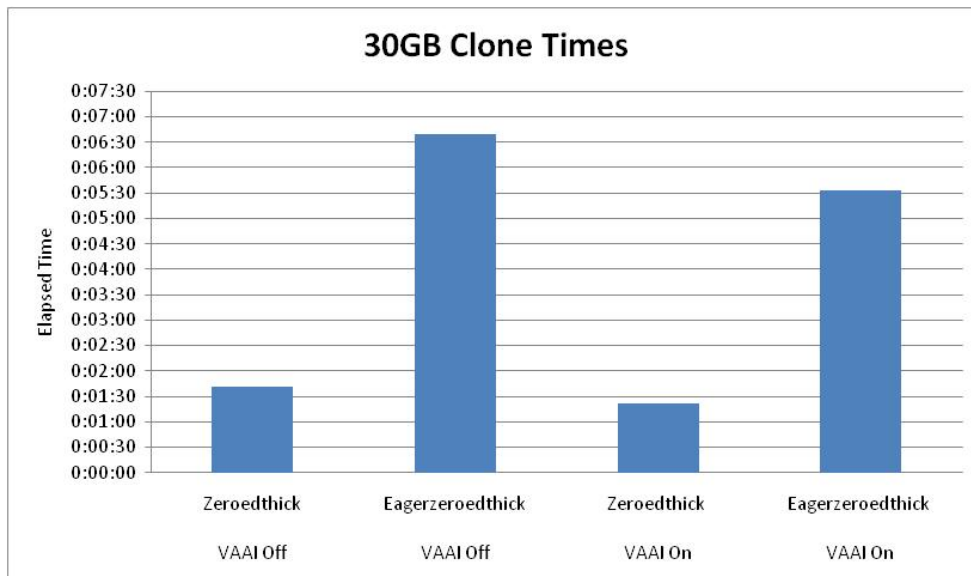


Figure 5

With VAAI disabled, the 30GB zeroedthick VM cloning task took one minute and 42 seconds; with VAAI enabled, the 30GB zeroedthick VM cloning task took one minute and 22 seconds, a 14 percent decrease. With VAAI disabled, the 30GB eagerzeroedthick VM cloning task took six minutes and 42 seconds; with VAAI enabled, the 30GB eagerzeroedthick VM cloning task took five minutes and 33 seconds, an 18 percent decrease

Key Finding — With the full copy primitive, you can greatly reduce provisioning times and deploy large numbers of VMs in less time and with less intrusion on your production environments.

In a similar test, a 30GB zeroedthick VM was cloned to capture the number of IOPS on the ESX host HBA. Figure 6 shows that the full copy primitive reduced the number of IOPS consumed by the host HBA by more than 90 percent.

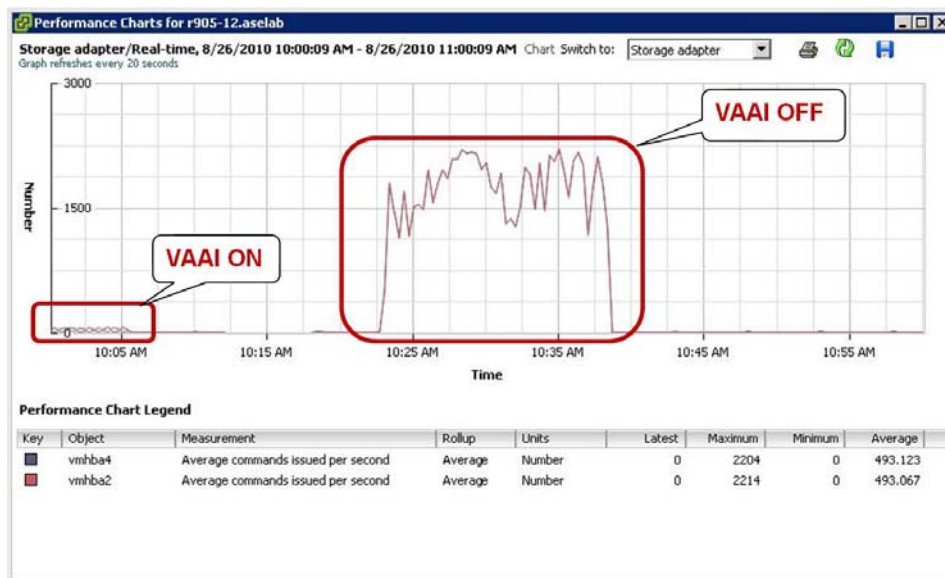


Figure 6

Figure 6 shows that with VAAI enabled, nearly all of the IOPS from the cloning process were offloaded from the ESX hosts to the storage system. With VAAI disabled, testing showed a prolonged spike in the number of IOPS on the ESX host.

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

VMDK Provisioning and Block Zeroing with Hitachi Dynamic Provisioning and Zero Page Reclaim

A new 30GB eagerzeroedthick VMDK file was created on a Dynamic Provisioning volume and the actual disk space utilization was measured on the storage system. With VAAI disabled, the virtual disk consumed 31GB of storage capacity on the Dynamic Provisioning pool. With VAAI enabled, it consumed only 1GB of storage capacity on the Dynamic Provisioning pool. Figure 7 shows storage space allocated when using eagerzeroedthick virtual disk type and a Dynamic Provisioning pool.

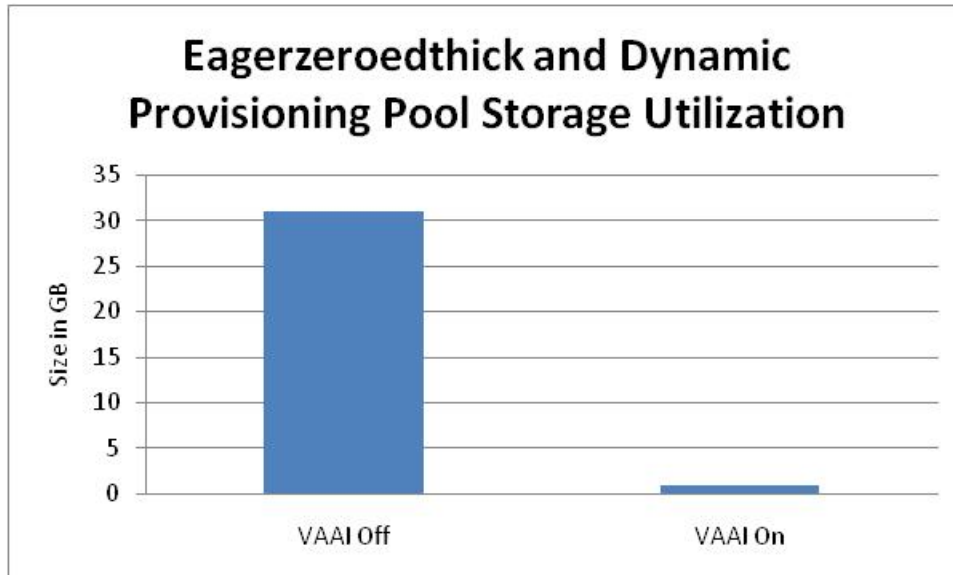


Figure 7

Block zeroing enables the formatting and provisioning of eagerzeroedthick VMDKs to be handled by the storage system rather than the ESX hosts.

As shown in Figure 7, an eagerzeroedthick VMDK is thin provisioned on the Hitachi Adaptable Modular Storage 2000 when provisioned on a Dynamic Provisioning Pool with VAAI enabled. In addition the ESX 4.1 host considers this fully provisioned with space pre-allocated and pre-zeroed. However, thin provisioned or any VMDKs that have not been pre-zeroed have a warm-up time. A test was conducted to see the warm-up effects of an eagerzeroedthick thin provisioned VMDK on a Dynamic Provisioning Pool.

In this case, Vdbench was used with a profile of 100 percent random writes with I/O rate set to maximum. To establish a baseline, a 100GB zeroedthick VMDK was used. Figure 8 shows the warm-up progression for IOPS and latency.

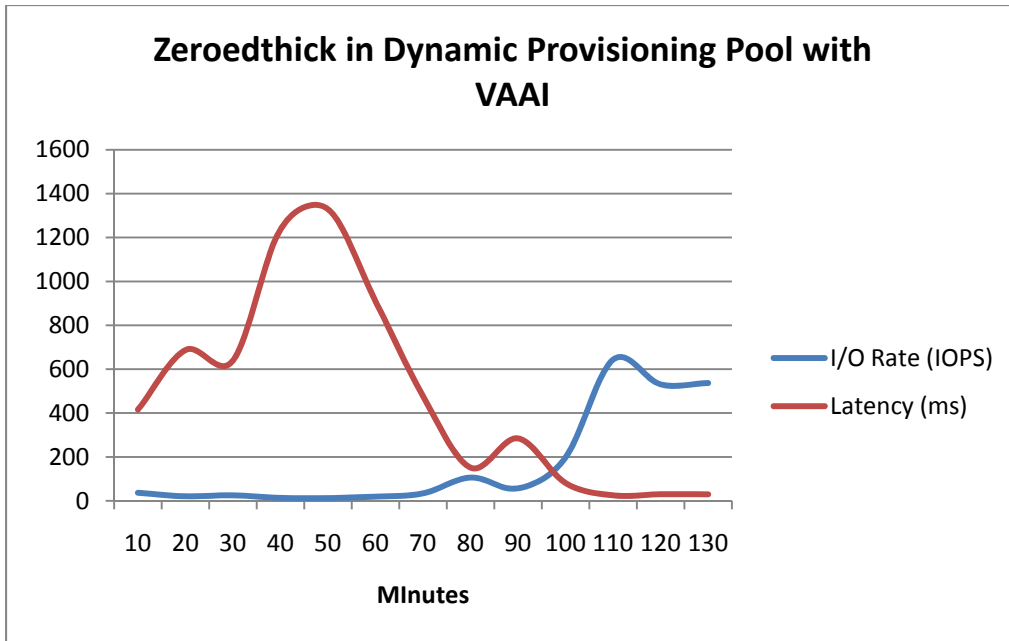


Figure 8

In this particular case, a zeroedthick VMDK in a Dynamic Provisioning pool with VAAI enabled took 120 minutes for the IOPS and latency to stabilize.

The same test was repeated on an eagerzeroedthick VMDK. Figure 9 shows the IOPS and latency.

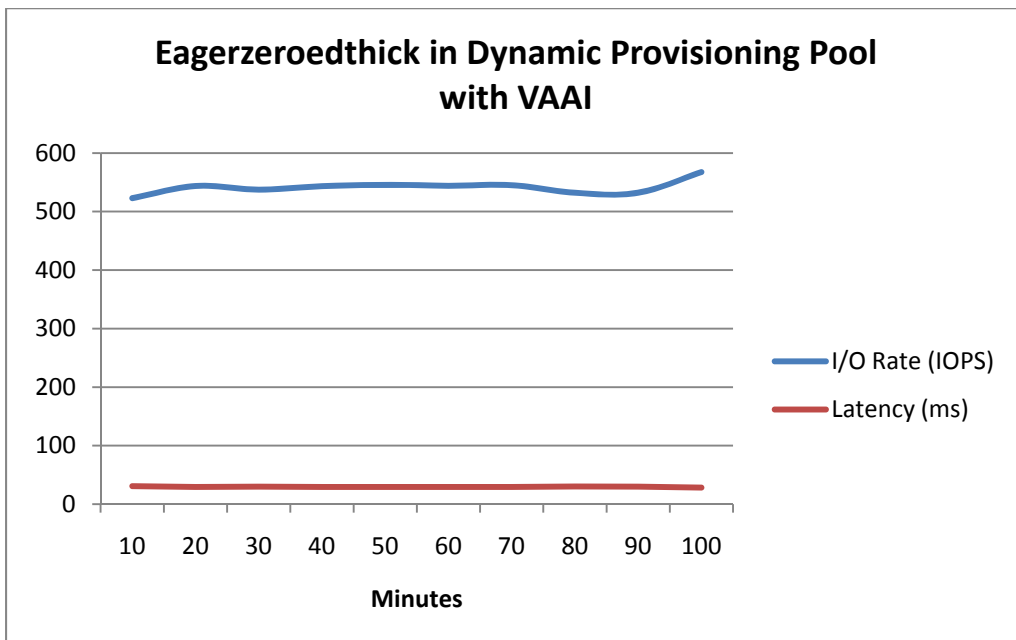


Figure 9

An eagerzeroedthick VMDK in a Dynamic Provisioning pool with VAAI enabled showed stable IOPS and latency from the start of the test. The I/O rate and latency were identical to that of zeroedthick VMDK post warm-up effects as shown in Figure 8. This shows eagerzeroedthick VMDK are thin provisioned in a Dynamic Provisioning pool with VAAI enabled, no warm-up anomalies occur.

Another benefit of the Hitachi's Adaptable Modular Storage 2000 family is the ability to run a zero page reclaim function on a Dynamically Provisioned volume. With zero page reclaim, any pages that are zeroed can be reallocated as needed. To demonstrate this functionality, VAAI was enabled and a 100GB zeroedthick VMDK file with 8GB of used space within the VMDK file was converted to an eagerzeroedthick VMDK file.

Figure 10 shows the existing eagerzeroedthick VMDK file on a Dynamic Provisioning volume before and after running a zero page reclaim operation.

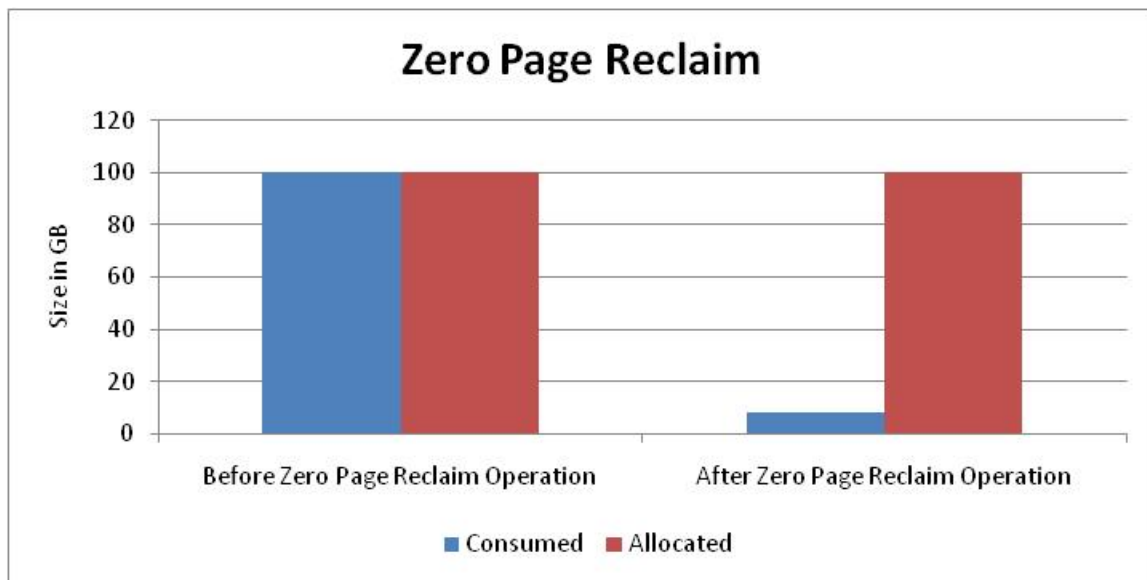


Figure 10

As expected, after the conversion, the entire 100GB that was allocated to the converted VMDK file was zeroed and committed to the VM. After the zero page reclaim operation ran, 92GB of the Dynamic Provisioning pool space was freed. As a result, after the zero page reclaim operation, the VM was still allocated the original 100GB but only 8GB of space was actually being consumed on the storage system by the VM.

Provisioning time was also captured using the vSphere client. Figure 11 shows that with VAAI enabled, the time required to provision a 30GB eagerzeroedthick VMDK file is reduced by 85 percent.

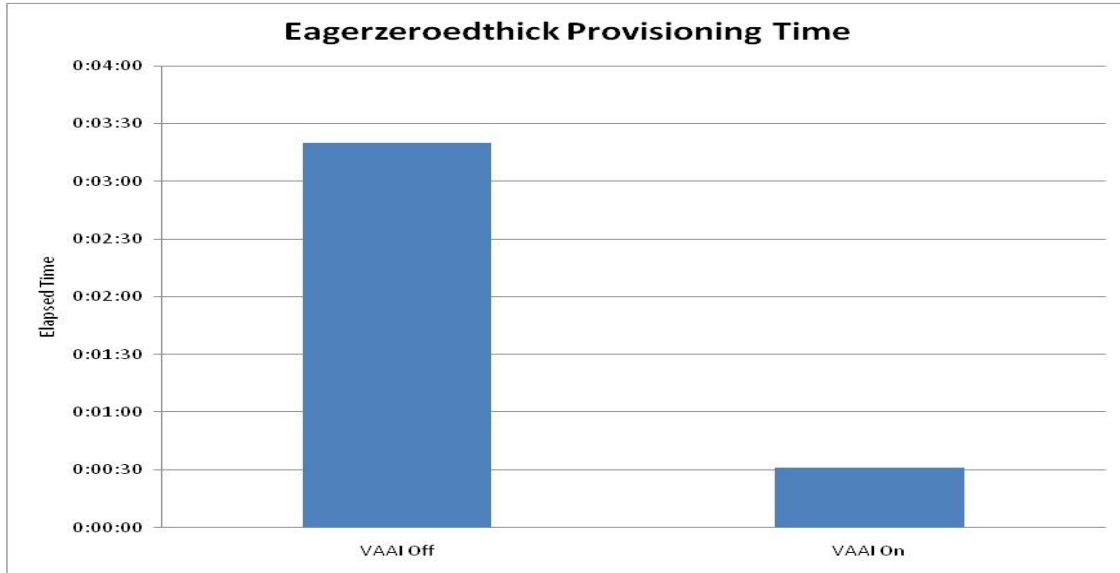


Figure 11

Key Finding — The block zeroing primitive allows for faster provisioning of eagerzeroedthick VMDK files. When combined with Hitachi Dynamic Provisioning, eagerzeroedthick VMDK thin provisioning is accelerated by the storage system hardware.

Large-scale VM Boot Storm with Hardware-assisted Locking

In this test case, 512 linked clone VMs were evenly distributed across four ESX 4.1 hosts using a single shared datastore, which was created on an LDEV provisioned from a 25-spindle Dynamic Provisioning pool. All the VMs were simultaneously powered on. The same test was then repeated using a datastore that was created on an LDEV provisioned from a five-spindle a Dynamic Provisioning pool to increase the likelihood of SCSI locking conflicts.

To determine the number of SCSI locking conflicts, the VMkernel logs were collected from each host and checked for SCSI reservation locking conflicts. In addition, the elapsed time was captured through the vSphere client. Figure 12 shows that with VAAI enabled, boot times decreased by 27 percent when using the datastore on the 25-spindle Dynamic Provisioning pool and 36 percent when using the datastore on the 5-spindle Dynamic Provisioning pool.

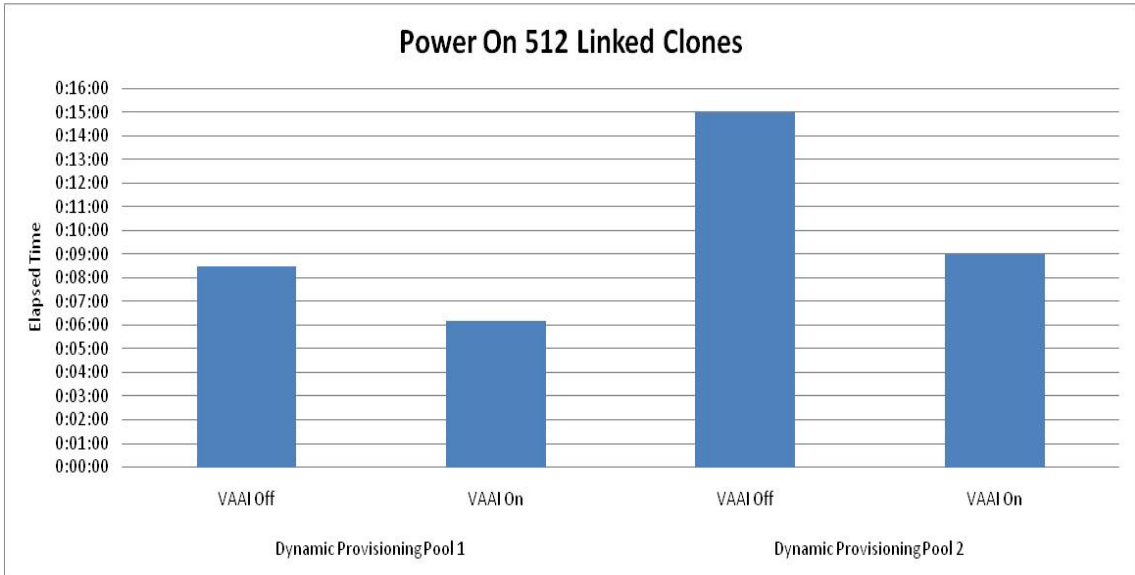


Figure 12

Figure 13 shows that the number of SCSI locking conflicts decreased by 70 percent for the datastore on the 25-spindle Dynamic Provisioning pool and by 75 percent for the datastore on the five-spindle Dynamic Provisioning pool with VAAI enabled.

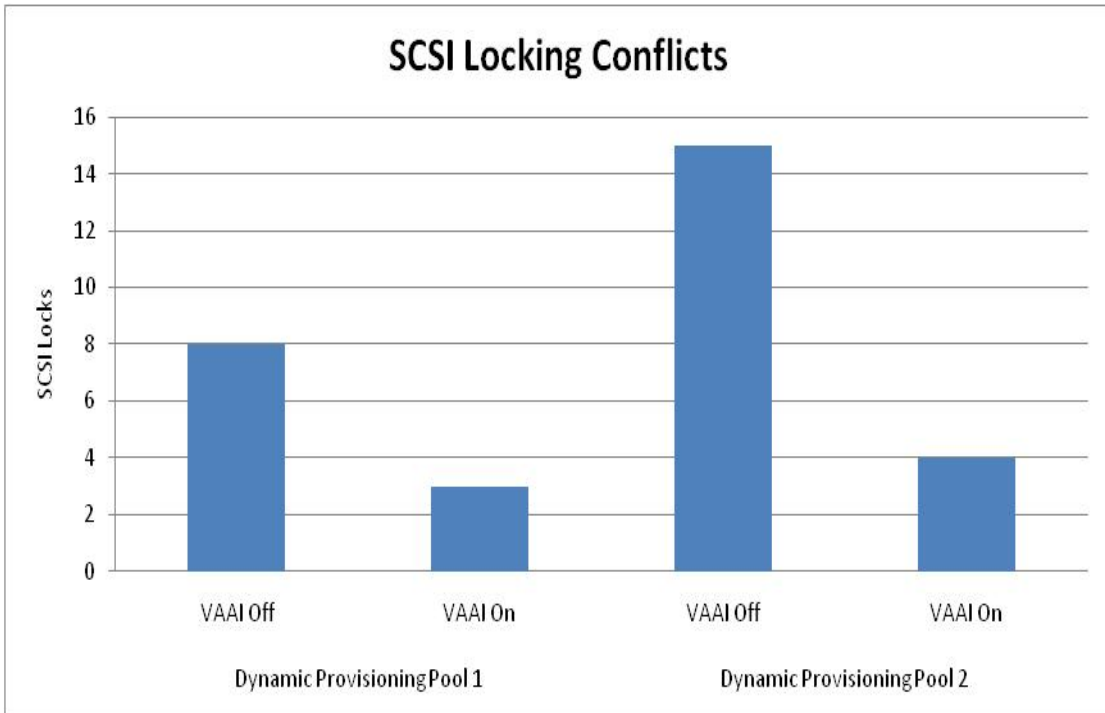


Figure 13

This showed that because the number of spindles decreased, the datastore was more prone to SCSI reservation locks, which can dramatically reduce performance and the number VMs that can be run on a datastore. However, with the hardware-assisted locking primitive enabled, the likelihood of SCSI reservation locking conflicts during everyday tasks such as Storage vMotion, creating or deleting VMDK files, powering off or on of VMs was greatly reduced.

Key Finding — Use of hardware-assisted locking primitive greatly improves the scalability of vSphere by allowing more VMs per datastore to run concurrently.

Large-scale Simultaneous vMotion with Hardware-assisted Locking

In this use case, 128 VMs of varying types (such as linked clones, thin provisioned VMs and VMs with snapshots) were evenly deployed across four ESX 4.1 hosts. To simulate a rolling upgrade or planned downtime, a single host was placed into maintenance mode forcing the VMs on that host to be moved with Storage vMotion to the remaining three hosts in the cluster. After all of the VMs had been moved off the host, it was brought back online. This operation was repeated on all four hosts and the time required to use Storage vMotion to move all the VMs from each host was collected through the vSphere client.

Figure 14 shows that the time required to move the VMs from each host was reduced by 34 percent with VAAI enabled.

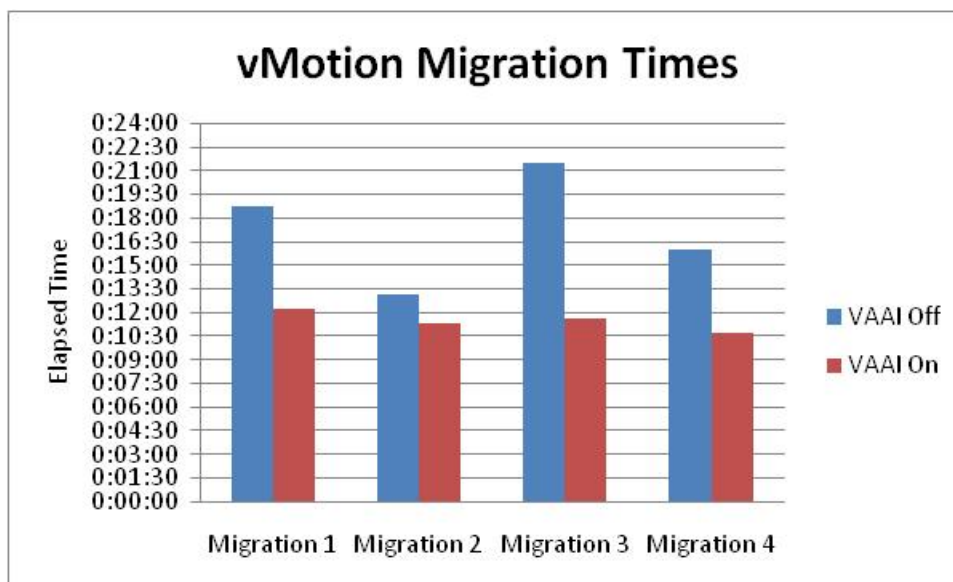


Figure 14. vMotion from Host

Key Finding — The hardware-assisted locking primitive accelerated large scale migration with vMotion, which benefits maintenance operations such as host patching and updates, resulting in shorter maintenance windows and less downtime.

Conclusion

VMware's VAAI primitives, when coupled with the Hitachi Adaptable Modular Storage 2000 family, provide you with the integration to build and maintain more scalable and efficient virtual environments. In addition, when you use these primitives with the 2000 family's load balancing active-active symmetric controllers and Hitachi Dynamic Provisioning software, you create a robust and high available infrastructure to support high density virtual machine workloads, as demonstrated by the testing reported in this white paper.

It is important to note that while the use of VAAI provides many benefits by offloading functions to your storage system, leveraging VAAI in your environment makes your choice of storage systems even more critical. It's key to choose a highly available, scalable, high performance storage system like the Hitachi Adaptable Modular 2000 family.

Table 3 summarizes the key benefits of using VAAI with the Hitachi Adaptable Modular Storage 2000 family as supported by the testing performed for this white paper.

Table 3. Key Findings

| <i>VAAI Primitive</i> | <i>Benefit</i> |
|---------------------------|---|
| Full copy | Speeds VM deployment with reduction of host HBA utilization. |
| Block zeroing | Enables storage systems to zero out a large number of blocks to speed provisioning of VMs. When combined with Hitachi Dynamic Provisioning software, eagerzeroedthick VMDK thin provisioning is accelerated by the storage system hardware, resulting in all virtual disk types being thin provisioned. |
| Hardware-assisted locking | 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 2TB single partition), thus simplifying storage configuration and sizing of LDEVs for VMFS volumes. |

The full copy primitive reduces host-side I/O during common tasks such as moving VMs with Storage vMotion or deploying a new VM from a template by instructing the storage system to copy data within the storage system rather than sending the traffic back and forth through the ESX hosts.

The block zeroing primitive speeds VM deployment by offloading the repetitive zeroing of large numbers of blocks to the storage system, freeing ESX host resources for other tasks.

The hardware-assisted locking primitive greatly reduces the probability of ESX hosts being locked out when attempting to access files on a VMFS datastore, which can degrade performance and in some cases cause tasks to time out or fail completely.

The tight integration of VMware's VAAI and the Hitachi Adaptable Modular Storage 2000 family provides a proven high-performance, highly scalable storage solution for any ESX environment.

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 [web site](#).

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 Academy [web site](#).

For more information about Hitachi products, contact your sales representative or channel partner or visit the Hitachi Data Systems [web site](#).

 **Hitachi Data Systems Corporation**

Hitachi is a registered trademark of Hitachi, Ltd., in the United States and other countries. Hitachi Data Systems is a registered trademark and service mark of Hitachi, Ltd., in the United States and other countries. All other trademarks, service marks and company names mentioned in this document are properties of their respective owners.

Notice: This document is for informational purposes only, and does not set forth any warranty, expressed or implied, concerning any equipment or service offered or to be offered by Hitachi Data Systems Corporation

© Hitachi Data Systems Corporation 2011. All Rights Reserved. AS-067-01 February 2011

Corporate Headquarters

750 Central Expressway,
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