VMware vSphere Best Practices Guide for the Pure Storage FlashArray
August 2015
Contents

Document Changes...................................................................................................................................................... 4
Executive Summary...................................................................................................................................................... 5
Goals and Objectives .................................................................................................................................................. 5
Audience ......................................................................................................................................................................... 5
Pure Storage Introduction .......................................................................................................................................... 6
Best Practice Checklist ................................................................................................................................................8
VMware Native Multipathing Plugin (NMP) Configuration...................................................................................... 10
  Configuring Round Robin..................................................................................................................................... 10
  Deleting a SATP Rule ............................................................................................................................................. 12
  Verifying Connectivity .......................................................................................................................................... 13
Choosing a Virtual Disk Allocation Method........................................................................................................... 16
SAN Design and Setup .............................................................................................................................................. 19
Virtual Machine Data Layout and Alignment ......................................................................................................... 21
Guest level tuning....................................................................................................................................................... 22
iSCSI Tuning ................................................................................................................................................................. 23
  Set Login Timeout to a Larger Value ................................................................................................................... 23
  Disable DelayedAck .............................................................................................................................................. 23
Pure Storage vSphere Web Client Plugin ................................................................................................................... 25
  User Access Control for the vSphere Web Client Plugin .................................................................................... 30
  Firewall requirements for the vSphere Web Client Plugin ................................................................................ 30
FlashArray Volume Management ............................................................................................................................... 31
  Creating a Volume ................................................................................................................................................. 31
  Creating Hosts .................................................................................................................................................... 32
Document Changes

Document changes have begun to be tracked starting with the August 2015 release of the document.

<table>
<thead>
<tr>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>August 2015</td>
<td>• Updated for vSphere 6.0: UNMAP changes (block count recommendation and option EnableBlockDelete), virtual disk recommendations, Web Client Plugin version.</td>
</tr>
<tr>
<td></td>
<td>• vRealize Operations Pack overview added.</td>
</tr>
<tr>
<td></td>
<td>• Document re-ordered.</td>
</tr>
<tr>
<td></td>
<td>• Disk.MaxiOSize recommendation added for EFI-enabled virtual machines.</td>
</tr>
<tr>
<td></td>
<td>• Checklist added.</td>
</tr>
<tr>
<td></td>
<td>• Updated for FlashArray/m.</td>
</tr>
</tbody>
</table>
Executive Summary

This document describes the best practices for using the Pure Storage FlashArray in VMware vSphere 5.x and 6.x environments. The Pure Storage FlashArray includes general support for VMware ESXi as well as a wide variety of direct product integrations to enhance and simplify the operation and management of both platforms. Throughout this paper, specific best practices on standard configuration as well as descriptions and recommendations on Pure Storage and VMware integration points will be explained.

This document is intended for use by pre-sales consulting engineers, sales engineers and customers who want to deploy the Pure Storage FlashArray in VMware vSphere-based virtualized datacenters.

Goals and Objectives

This document is intended to provide understanding and insight into any pertinent best practices when using VMware vSphere with the Pure Storage FlashArray. Options or configurations that are to be left at the default are generally not mentioned and recommendations for the default should be assumed. Changing or altering parameters not mentioned in this guide may in fact be supported but are likely not recommended and should be considered on a case-by-case basis. Please contact Pure Storage and/or VMware support in those cases.

This document will also detail out various Pure Storage integration pieces within the VMware ecosystem outside of the core hypervisor ESXi.

This document will be updated frequently, but for the most up-to-date information please refer to the following page on the Pure Storage Community

Audience

This document is intended for use by pre-sales consulting engineers, sales engineers, partners, and customers who want to deploy the Pure Storage FlashArray in VMware vSphere-based virtualized datacenters.
Pure Storage Introduction

Pure Storage is the leading all-flash enterprise array vendor, committed to enabling companies of all sizes to transform their businesses with flash.

Built on 100% consumer-grade MLC flash, Pure Storage FlashArray delivers all-flash enterprise storage that is 10X faster, more space and power efficient, more reliable, and infinitely simpler, and yet typically costs less than traditional performance disk arrays.

The Pure Storage FlashArray is ideal for:

**Accelerating Databases and Applications** Speed transactions by 10x with consistent low latency, enable online data analytics across wide datasets, and mix production, analytics, dev/test, and backup workloads without fear.

**Virtualizing and Consolidating Workloads** Easily accommodate the most IO-hungry Tier 1 workloads, increase consolidation rates (thereby reducing servers), simplify VI administration, and accelerate common administrative tasks.

**Delivering the Ultimate Virtual Desktop Experience** Support demanding users with better performance than physical desktops, scale without disruption from pilot to >1000’s of users, and experience all-flash performance for under $100/desktop.

**Protecting and Recovering Vital Data Assets** Provide an always-on protection for business-critical data, maintain performance even under failure conditions, and recover instantly with FlashRecover.

Pure Storage FlashArray sets the benchmark for all-flash enterprise storage arrays. It delivers:

**Consistent Performance** FlashArray delivers consistent <1ms average latency. Performance is optimized for the real-world applications workloads that are dominated by I/O sizes of 32K or larger vs. 4K/8K hero performance benchmarks. Full performance is maintained even under failures/updates.
**Less Cost than Disk** Inline de-duplication and compression deliver 5 – 10x space savings across a broad set of I/O workloads including Databases, Virtual Machines and Virtual Desktop Infrastructure.

**Mission-Critical Resiliency** FlashArray delivers >99.999% proven availability, as measured across the Pure Storage installed base and does so with non-disruptive everything without performance impact.

**Disaster Recovery Built-In** FlashArray offers native, fully-integrated, data reduction-optimized backup and disaster recovery at no additional cost. Setup disaster recovery with policy-based automation within minutes. And, recover instantly from local, space-efficient snapshots or remote replicas.

**Simplicity Built-In** FlashArray offers game-changing management simplicity that makes storage installation, configuration, provisioning and migration a snap. No more managing performance, RAID, tiers or caching. Achieve optimal application performance without any tuning at any layer. Manage the FlashArray the way you like it: Web-based GUI, CLI, VMware vCenter, Rest API, or OpenStack.

FlashArray scales from smaller workloads to data center-wide consolidation. And because upgrading performance and capacity on the FlashArray is always non-disruptive, you can start small and grow without impacting mission-critical applications. Coupled with **Forever Flash**, a new business model for storage acquisition and lifecycles, FlashArray provides a simple and economical approach to evolutionary storage that extends the useful life of an array and does away with the incumbent storage vendor practices of forklift upgrades and maintenance extortion.
Best Practice Checklist

Below is a checklist to refer to when configuring a VMware environment for the Pure Storage FlashArray. All settings that are not mentioned here should remain set to the default. Please refer to Pure Storage and/or VMware support when considering changes to settings not mentioned below. For more details, read through the rest of the document.

<table>
<thead>
<tr>
<th>Acknowledged/Done?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Configure <strong>Round Robin</strong> and an <strong>IO Operations Limit</strong> of 1 for every FlashArray device. The best way to do this is to create an ESXi SATP Rule on every host (below). This will make sure all devices are set properly automatically.</td>
</tr>
<tr>
<td></td>
<td><code>esxcli storage nmp satp rule add -s &quot;VMW_SATP_ALUA&quot; -V &quot;PURE&quot; -M &quot;FlashArray&quot; -P &quot;VMW_PSP_RR&quot; -O &quot;iops=1&quot;</code></td>
</tr>
<tr>
<td></td>
<td>Set the <strong>DataMover MaxHWTransferSize</strong> to 16 MB.</td>
</tr>
<tr>
<td></td>
<td>For iSCSI, enable <strong>DelayedAck</strong> and set the <strong>Login Timeout</strong> to 30 seconds. Jumbo Frames are optional.</td>
</tr>
<tr>
<td></td>
<td>In vSphere 6.x, change <strong>EnableBlockDelete</strong> to enabled.</td>
</tr>
<tr>
<td></td>
<td>Run UNMAP frequently. When running UNMAP in ESXi 5.5 and later, use a <strong>block count</strong> that is equal to or less than 1% of the free capacity on the target VMFS.</td>
</tr>
<tr>
<td></td>
<td>For ESXi hosts running EFI-enabled VMs set the ESXi parameter <strong>Disk.DiskMaxIOSize</strong> to 4 MB.</td>
</tr>
<tr>
<td></td>
<td><strong>DataMover.HardwareAcceleratedMove</strong>, <strong>DataMover.HardwareAcceleratedInit</strong>, and <strong>VMFS3.HardwareAcceleratedLocking</strong> should all be enabled.</td>
</tr>
<tr>
<td></td>
<td>Ensure all ESXi hosts are connected to both FlashArray controllers. Ideally at least two paths to each.</td>
</tr>
<tr>
<td></td>
<td>Install <strong>VMware tools</strong> whenever possible.</td>
</tr>
<tr>
<td></td>
<td>Leverage the <strong>Pure Storage Web Client Plugin</strong> for all provisioning, resizing and deletion tasks whenever possible.</td>
</tr>
<tr>
<td></td>
<td>When mounting snapshots, use the ESXi <strong>resignature</strong> option and avoid force-mounting.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>Configure <strong>Host Groups</strong> on the FlashArray identically to clusters in vSphere. For example, if a cluster has four hosts in it, create a corresponding Host Group on the relevant FlashArrays with exactly those four hosts—no more, no less.</td>
</tr>
<tr>
<td></td>
<td>Use Paravirtual SCSI adapters for virtual machines whenever possible.</td>
</tr>
<tr>
<td></td>
<td>There are no <strong>volume size</strong> or absolute <strong>virtual disk type</strong> requirements.</td>
</tr>
</tbody>
</table>

A PowerCLI best practices script can be found [here](#).
VMware Native Multipathing Plugin (NMP) Configuration

VMware revamped their Native Multipathing Plugin (NMP) layer in vSphere 4.0 with the introduction of Storage Array Type Plugins (SATP) and Path Selection Plugins (PSP) as part of the VMware APIs for Pluggable Storage Architecture (PSA). The SATP has all the knowledge of the storage array to aggregate I/Os across multiple channels and has the intelligence to send failover commands when a path has failed. The Path Selection Policy can be either “Fixed”, “Most Recently Used” or “Round Robin”.

Configuring Round Robin

The Pure Storage FlashArray is an ALUA-compliant array with all the paths optimized for doing active I/Os (active/optimized). When a FlashArray is connected to an ESXi farm, the array’s devices get claimed by the “VMW_SATP_ALUA” SATP.

It is important to note that the default path selection policy for this SATP is “Most Recently Used”. For optimal performance this needs to be changed to Round Robin.

Furthermore, an advanced parameter within the Round Robin PSP allows an administrator to configure how often Round Robin moves on to a different logical path for a given device when issuing I/O. By default, ESXi will switch to a new path every 1,000 I/Os that it issues to that device. Pure Storage recommends lowering this down to the minimum value of 1. This will cause ESXi to change paths after every I/O. This behavior will allow ESXi to better leverage the active-active nature of the FlashArray, therefore improving path failover time in the case of failure and improving I/O balance across the entire system. Making this change can increase performance around 10-15% or more for certain types of workloads. Please note that there will be a small additional cost to the ESXi CPUs.
The preferred option to set Round Robin is to create a rule that will cause any new FlashArray device that is added in the future to automatically get the Round Robin PSP and an IO Operation Limit value of 1. The following command creates a rule that achieves both of these for only Pure Storage FlashArray devices:

```
esxcli storage nmp satp rule add -s "VMW_SATP_ALUA" -V "PURE" -M "FlashArray" -P "VMW_PSP_RR" -O "iops=1"
```

This must be repeated for each ESXi host.

This can also be accomplished through PowerCLI. Once connected to a vCenter Server this script will iterate through all of the hosts in that particular vCenter and create a default rule to set Round Robin for all Pure Storage FlashArray devices with an I/O Operation Limit set to 1.

```
$hosts = get-vmhost
foreach ($esx in $hosts)
{
   $esxcli = get-esxcli -VMHost $esx
   $esxcli.storage.nmp.satp.rule.add($null, $null, "PURE FlashArray RR IO Operation Limit Rule", $null, $null, $null, "FlashArray", $null, "VMW_PSP_RR", "iops=1", "VMW_SATP_ALUA", $null, $null, "PURE")
}
```

It is important to note that existing, previously presented devices will need to be either manually set to Round Robin and an I/O Operation Limit of 1 or unclaimed and reclaimed through either a reboot of the host or through a manual device reclaim process so that it can inherit the configuration set forth by the new rule. For setting a new I/O Operation Limit on an existing device, see appendix I.
Deleting a SATP Rule

If you accidentally make a mistake in creating a SATP rule or you would like to change it, the rule must first be deleted and then re-created with the new settings you desire. Run the following command to list existing rules and their details:

```
esxcli storage nmp satp rule list
```

Identify the rule you would like to remove and/or change and then run the removal command. It is important to note that the removal command is very specific, you must include any field that is populated in the rule (except the description and rule group). So if the rule has an option that says `iops=1`, that must be in the removal command. If the rule has a vendor and model, both must be included and so on.

Let’s say we have the below rule (boxed in orange) and we want to delete it.

![Figure 4. SATP rule to be removed](image)

The name, vendor, model, default PSP and PSP options columns are populated. Therefore they must be included in the removal command. Forgetting the model name, for instance, will error out like such:

```
[root@csg-vmw-esx3:~] esxcli storage nmp satp rule remove -s VMW_SATP_ALUA -V PURE -P "VMW_PSP_RR" -O iops=1
```

Error deleting SATP rule: Sysinfo error on operation returned status : Not found. Please see the VMkernel log for detailed error information.

You will see an error like this in the vmkernel log:

```
```

Make sure you include all of those fields and any options are entered in identically. Case matters, for example, “iops=1” will work, but “IOPS=1” will not.

The following command will remove the rule listed in the above image correctly:

```
esxcli storage nmp satp rule remove -s VMW_SATP_ALUA -V PURE -M FlashArray -P "VMW_PSP_RR" -O iops=1
```
Verifying Connectivity

It is important to verify proper connectivity prior to implementing production workloads on a host or volume. This consists of a few steps:

1. Verifying proper multipathing settings in ESXi.
2. Verifying the proper numbers of paths
3. Verifying I/O balance and redundancy on the FlashArray

The Path Selection Policy and number of paths can be verified easily inside of the vSphere Web Client.

Verifying multipathing

This will report the path selection policy and the number of logical paths. The number of logical paths will depend on the number of HBAs, zoning and the number of ports cabled on the FlashArray.

The I/O Operations Limit cannot be checked from the vSphere Web Client—it can only be verified or altered via command line utilities. The following command can check a particular device for the PSP and I/O Operations Limit:

```
esxcli storage nmp device list -d naa.<device NAA>
```

Verifying multipathing

This will report the path selection policy and the number of logical paths. The number of logical paths will depend on the number of HBAs, zoning and the number of ports cabled on the FlashArray.
Please remember that each of these settings are a per-host setting, so while a volume might be configured properly on one host, it may not be correct on another. The PowerCLI script mentioned [here](#) can help you verify this at scale in a simple way.

It is also possible to check multipathing from the FlashArray.

In Purity 4.1.5 a CLI command was added to monitor I/O balance coming into the array:

```
purehost monitor -balance -interval <how long of a sample> --repeat <how many iterations>
```

The command will report a few things:

1. The host name
2. The individual initiators from the host. If they are logged into more than one FlashArray port, it will be reported more than once. If an initiator is not logged in at all, it will not appear
3. The port that initiator is logged into
4. The number of I/Os that came into that port from that initiator over the time period sampled
5. The relative percentage of I/Os for that initiator as compared to the maximum

The balance command will count the I/Os that came down the particular initiator during the sampled time period, and it will do that for all initiator/target relationships for that host. Whichever relationship/path has the most I/Os will be designated as 100%. The rest of the paths will be then denoted as a percentage of that number. So if a host has two paths, and the first path has 1,000 I/Os and the second path has 800, the first path will be 100% and the second will be 80%.

A well balanced host should be within a few percentage points of each path. Anything more than 15% or so might be worthy of investigation. Refer to [this post](#) for more information.

The GUI will also report on host connectivity in general, based on initiator logins.

![Host Connections](#)

**Figure 7. Host connectivity**

The connectivity may report one of the follows statuses:

- **None**: This means that the host does not have any paths to any of the controllers.
• **Single controller:** This means that the host has one or more paths to only one of the controllers. If this is reported it could be any number of issues, but likely troubleshooting should start with the switch.

• **Redundant (Uneven):** This means that the host does not have an equal number of paths to the two controllers (one has more than the other). This is most likely a zoning issue.

• **Redundant (Unused Port):** This means that the host has an equal number of paths to the two controllers. However, there is a host port that is defined but not connected to any of the controllers on the array. This probably means a HBA, SFP or cable is bad and/or zoning is incorrect. Once again, it would be wise to start with the switch in this case for troubleshooting.

• **Redundant:** All is good!

---

**Figure 8. Host connectivity—single controller**

Refer to [this post](#) for more information.
Choosing a Virtual Disk Allocation Method

Storage provisioning in virtual infrastructure involves multiple steps of crucial decisions. As discussed in the WRITE SAME section, VMware vSphere offers three virtual disks formats: thin, zeroedthick and eagerzeroedthick.

Choosing a virtual disk is a factor of a few different variables that need to be evaluated. The following considerations should be taken into account when choosing a virtual disk. Note that at the end of each consideration is a recommendation but that recommendation is valid only when that specific consideration is important. When choosing a virtual disk type take into account your vSphere design requirements and utilize these requirements to motivate your design decisions. Based on those decisions choose the virtual disk type that is best suitable for your environment.

- **Performance**—with the introduction of WRITE SAME support, the performance difference between the different types of virtual disks is essentially gone. In lab experiments, a difference can be observed during writes to unallocated portions of a thin or zeroedthick virtual disk. This difference is negligible but of course still non-zero. Therefore, performance is no longer an important factor in the type of virtual disk to use, but if absolute 100% consistent performance is absolutely required, eagerzeroedthick is slightly better than the others. **Recommendation: eagerzeroedthick.**
• **Protection against space exhaustion**—each virtual disk type, based on its architecture, has varying degrees of protection against space exhaustion. Thin virtual disks do not reserve space on the VMFS upon creation and instead grow in 1 MB blocks as needed. Therefore, if unmonitored, as one or more thin virtual disks grow on the datastore, they could exhaust the capacity of the VMFS. Even if the underlying array has plenty of additional capacity to provide. If careful monitoring is in place that provides the ability to make proactive resolution of capacity exhaustion (move the virtual machines around or grow the VMFS) thin virtual disks are a perfectly acceptable choice. Storage DRS is an excellent solution for space exhaustion prevention. While careful monitoring can protect against this possibility, it can still be of a concern and should be contemplated upon initial provisioning. Zeroedthick and eagerzeroedthick virtual disks are not susceptible to VMFS logical capacity exhaustion because the space is reserved on the VMFS upon creation. Due to zero removal in FlashReduce space for both types of thick virtual disks is not reserved on the physical layer so while they provide better protection against capacity exhaustion than thin, they are still equal to each other as neither ensure physical capacity. **Recommendation: eagerzeroedthick or zeroedthick.**

• **Virtual disk density**—it should be noted that while all virtual disk types take up the same amount of physical space on the FlashArray due to FlashReduce technology, they have different requirements on the VMFS layer. Thin virtual disks can be oversubscribed (more capacity provisioned than the VMFS reports as being available) allowing for far more virtual disks to fit on a given volume than either of the thick formats. This provides a greater virtual machine to VMFS density and reduces the number or size of volumes that are required to store them. This, in effect, reduces the management overhead of provisioning and managing additional volumes in a VMware environment. **Recommendation: thin.**

• **XCOPY performance**—eagerzeroedthick and zeroedthick virtual disks copy significantly faster than thin virtual disks when cloned or moved using VAAI XCOPY. As mentioned earlier in this document, the transfer size greatly affects the speed at which a virtual disk is copied via the VAAI XCOPY mechanism. Both “thick” types of virtual disks use the configured maximum transfer size of the host, which can go up to 16 MB. Thin virtual disks do not adhere to this value and instead adhere to a transfer size equal to 1 MB (the block size of VMFS). This will noticeably slow down XCOPY operations of thin virtual disks when compared to the thick disks. Therefore, in situations where copy rates are very important (such as virtual machine templates) using either of the thick types of virtual disks is highly recommended. This behavior has been fixed in vSphere 6.0 and thin virtual disks actually outperform the thick-type virtual disks. So this consideration is different for vSphere 6. **Recommendation in vSphere 5.x: eagerzeroedthick. In vSphere 6.x: thin.**

• **Time to create**—the virtual disk types also vary in how long it takes to initially create them. Since thin and zeroedthick virtual disks do not zero space until they are actually written to by a guest they are both created in trivial amounts of time—usually a second or two. Eagerzeroedthick disks, on the other hand, are pre-zeroed at creation and consequently take additional time to create. If the time-to-first-IO is paramount for whatever reason, thin or zeroedthick is best. **Recommendation: thin or zeroedthick.**
• **Array Space Efficiency**—the aforementioned bullet on “virtual disk density” describes efficiency on the VMFS layer. Efficiency on the underlying array should also be considered. By default all virtual disk (prior to storing guest OS data) are no different when it comes to consumed space on the FlashArray. All virtual disks type only consume SSD space when the guest OS has written actual data. The difference here is in space reclamation. In vSphere 6.0, thin virtual disks support guest-OS initiated UNMAP through the virtual disk, through the VMFS and down to the physical storage. Therefore, thin virtual disks can be more space efficient as time wears on and data is written and deleted. For more information on this functionality in vSphere 6.0, refer to the section, In-Guest UNMAP in ESXi 6.x, that can be found later in this paper. **Recommendation in vSphere 5.x: Not Applicable. In vSphere 6.x: thin.**

No virtual disk option quite fits all possible use-cases perfectly, so choosing an allocation method should generally be decided upon on a case-by-case basis. VMs that are intended for short term use, without high performance requirements, fit nicely with thin virtual disks. For VMs that are intended for long-term use and/or have higher performance needs eagerzeroedthick is a good choice. In vSphere 5.x, VM templates should always be eagerzeroedthick or zeroedthick. When there are no specific vSphere design requirements Pure Storage recommends eagerzeroedthick as a default choice for vSphere 5.x. For vSphere 6.0, thin virtual disks are highly recommended as the major lingering disadvantages of the format have been removed in 6.0.

**Figure 10. Virtual disk choice**
SAN Design and Setup

The need for a storage area network (SAN) in a virtual environment is well understood in the datacenter today as an overwhelming majority of the virtual infrastructure deployments are based on SANs. SANs provide storage connectivity to a virtual server or a cluster through a protocol like Fibre Channel or iSCSI which offer a high degree of reliability, availability and scalability (RAS). Designing a highly resilient SAN fabric (i.e. no single points of failure) includes at minimum: dual initiator ports per host, dual fabric switch infrastructures and a highly-available storage platform. These components are paramount for business critical application deployment.

For specific SAN switch configuration recommendations, it is advised to consult the switch vendor’s best practices. The following are some of the general guidelines while configuring a SAN switch:

- Implement dual SAN fabric design to supply redundant and resilient inter-hardware connectivity.
- As the Pure Storage FlashArray supports 8 Gb or 16 Gb modules on its front-end ports, a common mistake is connecting to a lower bandwidth SFP module on the switch (such as 4 Gb which is still prevalent) and due to auto-negotiation the port bandwidth drops to 4 Gb. Therefore, it is important to make sure you have 8 Gb or 16 Gb (as the case may be) SFP modules on the SAN switch.
- Implement a single initiator to single target zoning practice. Single initiator and multiple target zoning will work, but is not the best practice. Multiple initiators and multiple target is a definite risk for spurious Registered State-Change Notifications (RSCNs) waiting to overwhelm the initiators and targets, hence it should be avoided. Multiple initiators in a single zone widens the disaster area of SCSI resets.
- Some SAN switches may have restrictions around targets connected to port groups and in those environments you should configure accordingly per the switch vendors best practices. A common practice is to spread the targets to different port groups and hook the initiators in similar fashion.
- Always monitor the port statistics to see if the I/O load is spread evenly across all of the target ports. Since the Pure Storage FlashArray is an active/active array, there is no point in using just one port or only one controller.

Below is an example of a single server with dual port HBA connected to all the eight ports of a Pure Storage FlashArray FA-420 via two Fibre Channel switches. This provides highly-reliable connectivity from the host, across the SAN and to the storage.
Figure 11. SAN Connectivity Diagram
Virtual Machine Data Layout and Alignment

Misalignment leads to extra I/Os for every I/O initiated from the guest and this leads to longer latency and a potentially huge performance hit from a host CPU cycle and I/O throughput perspective. Almost universally, storage vendors have noted this as a problem and have devised tools to detect misaligned partitions, which can be cumbersome and painstakingly hard to use. Common reasons for misalignment are cloning from a template, P2V operation and creation of partitions with master boot record (MBR) that consumes 63 sectors. This makes partition alignment a very tricky problem. This problem is further exacerbated with vSphere 5.x where ESXi does not have a console OS to run the ordinary commands from. This causes grief to deploy applications written for Windows 2003 and other Operating Systems where alignment problems are quite real.

The Pure Storage FlashArray does not have to deal with alignment, as it is supports variable length block sizes down to a size of 512 bytes. Therefore, whatever the size of the allocation unit on the guest, the FlashArray can still deliver sub-millisecond latency on reads/writes. It really doesn’t matter if the I/O size is sector aligned or 4k aligned, as received reads and writes are subject to various optimizations. For further discussion on partition alignment read the blog article at:

Guest level tuning

There are no special configurations that are required to use the Pure Storage FlashArray in customer environments. All of the best practices that customers have currently deployed are equally applicable. A summary of best practices is noted below.

1. Configure guest Operating Systems in accordance with the corresponding vendor installation guidelines.

2. Provision vCPUs and memory as per the application requirements.

3. Install and configure VMware tools. Clock Sync, VMware paravirtual driver, and advanced graphics support are all included in the package, so the install of VMware Tools should be considered a crucial step in VM configuration.

4. Configure the paravirtual SCSI adapter for intensive-workload guests. In testing, Pure Storage has found this to perform much better than other adapters. See the VMware KB here for information on how to configure this—note if configuring for a template this is a one-time configuration that will persist for all VMs deployed from it.

5. If the paravirtual driver is not supported on the guest OS, then LSI Logic SAS is an acceptable alternative.

6. If a VM is using RDMs make sure the device multipathing policy is managed from the virtual machine settings and is configured to use the Round-Robin PSP.

7. If a VM is using EFI (Extensible Firmware Interface) instead of BIOS, it is necessary to reduce the ESXi parameter called Disk.DiskMaxIOMinSize from the default of 32 MB (32768 KB) down to 4 MB (4096 KB). This should be set on every ESXi host EFI-enabled virtual machines have access to, in order to provide for vMotion support. For more information about EFI, refer to the VMware article here.
iSCSI Tuning

Just like any other array that supports iSCSI, Pure Storage recommends the following changes to an iSCSI-based vSphere environment for the best performance.

iSCSI tuning is mutually exclusive to FC tuning. Though, multi-pathing recommendations such as configuring Round Robin and a decreased I/O Operations Limit are still required.

For a detailed walkthrough of setting up iSCSI on VMware ESXi and on the FlashArray please refer to the following blog post:


Set Login Timeout to a Larger Value

For example, to set the Login Timeout value to 30 seconds, use commands similar to the following:

1. Log in to the vSphere Web Client and select the host under Hosts and Clusters.
2. Navigate to the Manage tab.
3. Select the Storage option.
4. Under Storage Adapters, select the iSCSI vmhba to be modified.
5. Select Advanced and change the Login Timeout parameter. This can be done on the iSCSI adapter itself or on a specific target.

The default Login Timeout value is 5 seconds and the maximum value is 60 seconds.

Disable DelayedAck

DelayedAck is an advanced iSCSI option that allows or disallows an iSCSI initiator to delay acknowledgement of received data packets.

Disabling DelayedAck in ESXi 5.x:

1. Log in to the vSphere Web Client and select the host under Hosts and Clusters.
2. Navigate to the Manage tab.
3. Select the Storage option.

4. Under Storage Adapters, select the iSCSI vmhba to be modified.

Navigate to Advanced Options and modify the DelayedAck setting by using the option that best matches your requirements, as follows:

**Option 1:** Modify the DelayedAck setting on a particular discovery address (recommended) as follows:

1. Select Targets.
2. On a discovery address, select the Dynamic Discovery tab.
3. Select the iSCSI server.
4. Click Advanced.
5. Change DelayedAck to false.

**Option 2:** Modify the DelayedAck setting on a specific target as follows:

1. Select Targets.
2. Select the Static Discovery tab.
3. Select the iSCSI server and click Advanced.
4. Change DelayedAck to false.

**Option 3:** Modify the DelayedAck setting globally for the iSCSI adapter as follows:

1. Select the Advanced Options tab and click Advanced.
2. Change DelayedAck to false.

iSCSI CHAP is also supported for uni- or bidirectional authentication. Please refer to the following post for a detailed walkthrough.


Please note that iSCSI CHAP is not currently supported with dynamic iSCSI targets on the FlashArray. If CHAP is going to be used, please configure your iSCSI FlashArray targets as static only. Dynamic support for CHAP will be added in a future Purity release.

For information on jumbo frames, please refer to the appendix. Please note that jumbo frames is not a mandatory configuration and its use is up to the discretion of the user.
Pure Storage vSphere Web Client Plugin

Beginning with vSphere 5.1, Pure Storage offers a direct plugin to the vSphere Web Client that allows for full management of a FlashArray as well as a variety of integrated menu options to provision and manage storage.

Prior to use, the Web Client Plugin must be installed and configured on the target vCenter server. There is no requirement to go to an external web site to download the plugin—it is stored on the FlashArray controllers.

The Pure Storage vSphere Web Client Plugin is supported with both Windows and Linux-based vCenter servers. For vSphere 6.0, the 2.0.6 version or later of the plugin is required.

Login to the Pure Storage GUI and navigate to the System > Plugins > vSphere section as seen in the figure below.

![Figure 12. Pure Storage vSphere Web Client Plugin installation](image)

Enter in the vCenter IP address or FQDN as well as the appropriate administrative credentials and click “Connect”. Click “Install” for a new Web Client install or “Update” to upgrade from an older version of the plugin.
Once the plugin has been configured, it can be accessed from the vSphere Web Client interface. Login to the respective instance of the Web Client, navigate to the home screen and a new Pure Storage icon will be listed under “Inventories”.

Before the plugin can be used, one or more arrays must be added. Individual FlashArrays can be added from within the home screen of the plugin by supplying a name, URL and credentials for the target array. Note that the plugin only needs to be installed once from a FlashArray on a given vCenter instance. Once it has been installed any number of other FlashArrays can be configured for use from within the plugin. Always use the virtual IP address of the FlashArray instead of a direct IP address of a controller. This will provide for Web Client connection resiliency.
From the Pure Storage plugin home page, individual arrays can be authorized for access.

One of the most basic features of the plugin is the ability to see underlying storage information in-context of a given datastore residing on a FlashArray. This is built directly into the standard datastore views in the Web Client. In the vSphere Web Client, a new tab will appear for datastores called Pure Storage.

From that tab, data reduction rates and performance information for that device can be seen.

Additionally, users can manage existing storage or create new volumes in-context from within the vSphere Web Client. The options provided are:

1. **Provision new volumes**—this will create a new volume on the Pure Storage FlashArray and present it to a host or cluster. It will automatically rescan the host(s) and then format the volume as VMFS. Optionally, you can have the wizard add the new volume to a pre-existing Protection Group. A Protection Group is a management object on the FlashArray that provides a local snapshot and/or remote replication schedule for the volumes in that group.

2. **Expand existing volumes**—any existing volume on an authorized array can be expanded non-disruptively. The plugin will resize the FlashArray volume to the new size, rescan the host(s) and then automatically resize the hosted VMFS to encompass the new capacity.
3. **Destroy volumes**—this will unmount and remove a given VMFS volume and then destroy it on the FlashArray. A destroyed volume can be recovered through the Pure GUI or CLI for 24 hours after destruction.

4. **Manage snapshots**—users can create snapshots of a datastore or a set of datastores, restore volumes from snapshots, create new volumes from snapshots and delete snapshots.

5. **Adjust datastore protection**—datastores can be added or removed to FlashArray Protection Groups to start or cease local and/or remote replication for that datastore.

6. **Rename volumes**—underlying FlashArray volumes can be renamed from within the Web Client.

7. **Configure multipathing**—all of the FlashArray datastores in a host or a host cluster can be quickly configured to use the Round Robin multipathing policy. It is important to note the Web Client Plugin does not currently alter the IO Operations limit so it is left at the default of 1,000.

8. **Check storage health**—ESXi hosts can be quickly checked to make sure host storage limits are not being exceeded.

Prior to provisioning new storage the target host or host group must be configured on the array—the plugin will not create a host or host group itself. The plugin will recognize the proper host or host group and connect the new volume to the appropriate object. When provisioning to an individual host, a host group does not need to be created. When provisioning to a cluster there must be a corresponding host group pre-created that includes exactly all of the hosts in the target cluster.
If a name is entered that is already in use on the target array, (or within the vCenter by an existing VMFS) the provisioning operation will not be allowed until a unique name is chosen. The Pure Storage volume and the VMFS will be both be named with the user-supplied label.

When a volume is destroyed using the plugin, the following steps are followed:

1. The VMFS volume is gracefully unmounted.
2. The FlashArray SCSI device is detached from all of the hosts that have access to it.
3. The FlashArray volume is disconnected from any hosts or host groups on the array.
4. The FlashArray volume is destroyed on the array.

The last feature of the Pure Storage plugin for the vSphere Web Client is full integration of the Pure GUI directly into the Web Client interface. In the home page for the plugin a configured array can be selected and the Pure GUI can be shown by clicking on the “Pure Storage” tab.
For a full description on the use of the plugin, please refer to the user guide.

**User Access Control for the vSphere Web Client Plugin**

Presently, the FlashArray supports three levels of access roles when integrated with a directory service like Active Directory:

- **Array Admin Group**: Administrators that are allowed to perform every FlashArray operation including configuration. Array Admin Group administrators have the same privileges as the original built-in `pureuser`.

- **Storage Admin Group**: Administrators that are allowed to perform FlashArray storage operations (provision, snap, etc).

- **Read Only Group**: Users with read-only privileges on the FlashArray—they can view information but not provision/change anything.

An array can be registered within the vSphere Web Client with an authenticated user account that is in any one of the three user groups (or of course the built-in `pureuser`). The access rights of the user that registered a particular array will dictate the access rights for plugin access to the array for any user that logs into the vSphere Web Client. If an admin would like to allow for the viewing of underlying capacity information and performance data, but prevent provisioning, they should register the array with a read-only account. To enable provisioning, temporarily change the array registration with that of an elevated user account.

**Firewall requirements for the vSphere Web Client Plugin**

For initial install of the plugin from a FlashArray controller to the Web Client, network access must be granted to the target port of TCP 443 on the vCenter server from the FlashArray primary controller IP address. The plugin is always “pushed” from the primary controller regardless to which controller the user is logged into. This firewall rule does not need to persist, it must only be enabled for initial installs, upgrades or uninstalls of the plugin.

For use of the plugin, a rule to allow access to target TCP port 443 on the FlashArray from the vCenter must be created. The IP address on the FlashArray is whichever IP address the user plans to register that particular array with within the plugin. It is a best practice to always use the virtual IP of the FlashArray.
FlashArray Volume Management

In addition to using the Pure Storage vSphere Web Client Plugin, standard provisioning methods through the Pure GUI or Pure CLI can be utilized. This section highlights the end-to-end provisioning of storage volumes on the Pure Storage FlashArray from creation of a volume to formatting it on an ESXi host. The management simplicity is one of the guiding principles of FlashArray as just a few clicks are required to configure and provision storage to the server.

Whenever possible in VMware environments, all volume creation, expansion, and deletion should be run via the Pure Storage vSphere Web Client Plugin.

Creating a Volume

It is highly recommended to use the Pure Storage Web Client Plugin for all volume management (page Error! Bookmark not defined.), but in special circumstances, such as boot from SAN, either the standard Pure GUI or CLI are necessary.

To create a volume, the CLI command below can be executed with the user supplying a unique volume name and a capacity.

![Image: Creating a volume with the Pure GUI](image)

Figure 19. Creating a volume with the Pure GUI
Creating Hosts

In order to present a volume to an ESXi host (or cluster), a host (or host group) must be created on the FlashArray. In clustered storage environments like vSphere with VMFS, it is highly recommended to assign hosts to host groups. This simplifies storage provisioning by making it simple to provision the same volume to a number of hosts at once, in this case, you would provision to the host group not the individual host. To provide for virtual machine mobility, with the exception of devices that should not be clustered, such as boot volumes, all devices should be connected to a host group and not just a host.

Furthermore host groups should exactly correspond to the group of ESXi hosts (no more and no less) in a given cluster. So for every ESXi host in a cluster, it should be configured in an identical host group on the FlashArray. The WWNs or iSCSI addresses of the HBAs of each ESXi host are required in order to properly create the host record on the FlashArray. These can be retrieved through the vSphere Web Client (for information on retrieving this from the CLI refer to VMware documentation).

![Figure 20. Locating the port WWN (pwwn or wwpn) of the ESX HBA on vCenter](image)

Once the WWNs have been identified, the hosts can be added to the array using the GUI.
Figure 21. Creating a host group with the Pure GUI

Figure 22. Adding WWNs to a host with the Pure GUI
Connecting Volumes to ESXi Hosts

Once hosts and/or host groups have been created, the newly created volumes must be “connected” to the hosts. This is a process traditionally referred to as volume masking.

To connect a volume to a host or host group with the GUI, use the following process in Figure 24.

---

**Figure 23. Adding a host to a host group with the Pure GUI**
Rescanning an ESXi Host or Cluster

Once the creation of volumes and hosts/host groups and the volume connection is complete, the volumes will be accessible to the ESXi host(s). Using the vSphere Web Client, initiate a “Rescan Storage…” to make the newly-connected Pure Storage volume(s) fully-visible to the ESXi servers in the cluster as shown below. One can then use the “Add Storage” wizard to format the newly added volume.

1 Presuming SAN zoning is completed.
Growing a Volume

In VMware environments, it is always recommended to use the Pure Storage vSphere Web Client Plugin to increase volume capacities. Refer to Figure 8 for instructions.

Shrinking a Volume

While it is possible to shrink a FlashArray volume non-disruptively, vSphere does not have the ability to shrink a VMFS partition. Therefore, **do not shrink** FlashArray volumes that contain VMFS datastores as doing so could incur **data loss**.

Deleting a Volume

Whenever possible, use the Pure Storage Web Client Plugin to delete volumes. For deletion of boot from SAN volumes, the Pure GUI or CLI must be used. This will ensure VMFS volumes are not accidentally deleted. Refer to appendix II for information on manually unmounting and deleting VMFS volumes.
Mounting a Snapshot Volume

The Pure Storage FlashArray provides the ability to take local or remote point-in-time snapshots of volumes which can then be used for backup/restore and/or test/dev. When a snapshot is taken of a volume containing a VMFS, there are a few additional steps from both the FlashArray and vSphere sides to be able to access the snapshotted data.

When a FlashArray snapshot is taken, a new volume is not created—essentially it is a point-in-time reference to a past data version that has no volume identifier. Instead a snapshot has to be “copied” to an actual volume which then allows the snapshot data to be presented to a host. This behavior allows the snapshot to be re-used again and again without changing the data in that snapshot. If a snapshot is not needed more than one time an alternative option is to create a direct snap copy from one volume to another—merging the snapshot creation step with the association step.

The following images show the process of copying the snapshot to a new volume and then mounting the copied VMFS volume (the process of connecting it to a host group and rescanning the host is not shown).

![Figure 26. Copying a snapshot to a volume with the Pure GUI](image-url)
Since the VMFS is on a volume that has a different serial number, the VMFS will be reported as having an invalid signature since the VMFS signature is a hash partially based on the serial of the hosting device. Therefore the device will not be automatically mounted upon rescan—the new datastore wizard needs to be run to find the device and resignature the volume. It is almost always recommended to resignature rather than force mounting a device.

Figure 27. Resignaturing a copied VMFS volume with the vSphere Web Client
VMware vStorage APIs for Array Integration (VAAI)

The vStorage APIs for Array Integration (VAAI) is a feature set introduced vSphere 4.1 by VMware that provides hardware acceleration and offloading functionality. These features enable an ESXi host to offload specific virtual machine and storage management operations to compliant storage hardware. With the storage hardware assistance, an ESXi host can perform these operations faster and more efficiently while consuming far less CPU, memory, and storage fabric bandwidth. The options to support VAAI are enabled by default and will automatically be invoked if an ESXi host detects that there is support from the underlying storage (this check is done on a per-device basis). Pure Storage FlashArray supports VAAI in ESXi 5.0 and later.

The following five vStorage APIs are available for block-storage hardware vendors to implement and support:

- **Hardware Assisted Locking**—commonly referred to as Atomic Test & Set (ATS) uses the SCSI command COMPARE and WRITE (0x89), which is used during the creation, alteration and deletion of files and metadata on a VMFS volume.

- **Full Copy**—leverages the SCSI command XCOPY (0x83), which is used to copy or move virtual disks.

- **Block Zero**—leverages the SCSI command WRITE SAME (0x93) which is used to zero-out virtual disk regions during virtual disk block allocation operations.

- **Dead Space Reclamation**—leverages the SCSI command UNMAP (0x42) to reclaim previously used but now deleted space on a block SCSI device.

- **Thin Provisioning Stun and Resume**—allows for the underlying storage to inform ESXi that capacity has been entirely consumed, which then causes ESXi to immediately “pause” virtual machines until additional capacity can be provisioned.

Pure Storage does not support DISABLING VAAI features on ESXi hosts—if a customer is required to disable one or more features for some reason they must contact Pure Storage support if the affected ESXi hosts have access to Pure Storage FlashArray devices.

Pure Storage FlashArray supports ATS, XCOPY, WRITE SAME and UNMAP in Purity release 3.0.0 onwards on ESXi 5.x and later.

This section is meant to be an overview of VAAI integration with the Pure Storage FlashArray. For a detailed discussion, please refer to the white paper, *VMware Storage APIs for Array Integration with the Pure Storage FlashArray*.

---

2 Thin Provisioning Stun & Resume is not currently supported by the Pure Storage FlashArray.
**ATS or Hardware Assisted Locking**

Prior to the introduction of VAAI ATS (Atomic Test and Set), VMFS used LUN-level locking via full SCSI reservations to acquire exclusive metadata control for a VMFS volume. In a cluster with multiple nodes, all metadata operations were serialized and hosts had to wait until whichever host, currently holding a lock, released that lock. This behavior not only caused metadata lock queues but also prevented standard I/O to a volume from VMs on other ESXi hosts which were not currently holding the lock.

With VAAI ATS, the lock granularity is reduced to a much smaller level of control (specific metadata segments, not an entire volume) for the VMFS that a given host needs to access. This behavior makes the metadata change process not only very efficient, but more importantly provides a mechanism for parallel metadata access while still maintaining data integrity and availability. ATS allows for ESXi hosts to no longer have to queue metadata change requests, which consequently speeds up operations that previously had to wait for a lock. Therefore, situations with large amounts of simultaneous virtual machine provisioning operations will see the most benefit. The standard use cases benefiting the most from ATS include:

- High virtual machine to VMFS density
- Extremely dynamic environments—numerous provisioning and de-provisioning of VMs (e.g. VDI using non-persistent linked-clones).
- High intensity virtual machine operations such as boot storms, or virtual disk growth

**Full Copy or Hardware Accelerated Copy**

Prior to Full Copy (XCOPY) API support, when virtual machines needed to be copied or moved from one location to another, such as with Storage vMotion or a virtual machine cloning operation, ESXi would issue many SCSI read/write commands between the source and target storage location (the same or different device). This resulted in a very intense and often lengthy additional workload to this set of devices. This SCSI I/O consequently stole available bandwidth from more “important” I/O such as the I/O issued from virtualized applications. Therefore, copy or move operations often had to be scheduled to occur only during non-peak hours in order to limit interference with normal production storage performance. This restriction effectively decreased the ability of administrators to use the virtualized infrastructure in the dynamic and flexible nature that was intended.

The introduction of XCOPY support for virtual machine movement allows for this workload to be offloaded from the virtualization stack to almost entirely onto the storage array. The ESXi kernel is no longer directly in the data copy path and the storage array instead does all the work. XCOPY functions by having the ESXi host identify a region of a VMFS that needs to be copied. ESXi describes this space into a series of XCOPY SCSI commands and sends them to the array. The array then translates these block descriptors and copies/moves the data from the described source locations to the described target locations. This architecture therefore does not require the moved data to be sent back and forth between the host and array—the SAN fabric does
not play a role in traversing the data. This vastly reduces the time to move data. XCOPY benefits are leveraged during the following operations:

- Virtual machine cloning
- Storage vMotion
- Deploying virtual machines from template

During these offloaded operations, the throughput required on the data path is greatly reduced as well as the ESXi hardware resources (HBAs, CPUs etc.) initiating the request. This frees up resources for more important virtual machine operations by letting the ESXi resources do what they do best: run virtual machines, and lets the storage do what it does best: manage the storage.

On the Pure Storage FlashArray, XCOPY sessions are exceptionally quick and efficient. Due to FlashReduce technology (features like deduplication, pattern removal and compression) similar data is never stored on the FlashArray more than once. Therefore, during a host-initiated copy operation such as with XCOPY, the FlashArray does not need to copy the data—this would be wasteful. Instead, Purity simply accepts and acknowledges the XCOPY requests and creates new (or in the case of Storage vMotion, redirects existing) metadata pointers. By not actually having to copy/move data, the offload duration is greatly reduced. In effect, the XCOPY process is a 100% inline deduplicated operation. A non-VAAI copy process for a virtual machine containing 50 GB of data can take on the order of multiple minutes or more depending on the workload on the SAN. When XCOPY is enabled this time drops to a matter of a few seconds.

3 Note that there are VMware-enforced caveats in certain situations that would prevent XCOPY behavior and revert to legacy software copy. Refer to VMware documentation for this information at www.vmware.com.
XCOPY on the Pure Storage FlashArray works directly out of the box without any configuration required. Nevertheless, there is one simple configuration change on the ESXi hosts that will increase the speed of XCOPY operations. ESXi offers an advanced setting called the MaxHWTransferSize that controls the maximum amount of data space that a single XCOPY SCSI command can describe. The default value for this setting is 4 MB. This means that any given XCOPY SCSI command sent from that ESXi host cannot exceed 4 MB of described data.

The FlashArray, as previously noted, does not actually copy the data described in a XCOPY transaction—it just moves or copies metadata pointers. Therefore, for the most part, the bottleneck of any given virtual machine operation that leverages XCOPY is not the act of moving the data (since no data is moved), but how quickly an ESXi host can send XCOPY SCSI commands to the array. Therefore, this is directly related to the number of commands sent and correct multipathing configuration. Assuming the multipathing is correct, the only other option is to decrease the number of commands sent. This is achieved by changing the MaxHWTransferSize. The more data that can be described in a single XCOPY command, the less commands overall need to be sent and it takes less time for the total operation to complete. For this reason, Pure Storage recommends setting this to the maximum value of 16 MB4.

The following commands provide for retrieval of the current value, and for setting a new one.

```bash
esxcfg-advcfg -g /DataMover/MaxHWTransferSize
Value of MaxHWTransferSize is 4096
esxcfg-advcfg -s 16384 /DataMover/MaxHWTransferSize
Value of MaxHWTransferSize is 16384
```

This can also be set using PowerCLI. The below command (after a connection to a vCenter is established) connects to and changes the transfer size setting on all of the hosts in the vCenter:

```powershell
$hosts = get-vmhost
foreach ($esx in $hosts)
{
    $esxcli -get-esxcli -VMHost $esx
    $esx | Get-AdvancedSetting -Name DataMover.MaxHWTransferSize | Set-AdvancedSetting -Value 16384 -Confirm:$false
}
```

**Block Zero or WRITE SAME**

ESXi supports three disk formats for provisioning virtual disks:

1. Eagerzeroedthick—the entirety of the virtual disk is completely reserved and pre-zeroed on the VMFS upon creation. This virtual disk allocation mechanism offers the most predictable performance and

4 Note that this is a host-wide setting and will affect all arrays attached to the host. If a third party array is present and does not support this change leave the value at the default or isolate that array to separate hosts.
the highest level of protection against capacity exhaustion.

2. **Zeroedthick**—this format reserves the space on the VMFS volume upon creation but does not pre-zero the allocated blocks until the guest OS writes to them. New writes cause iterations of on-demand zeroing in segments of the block size of the target VMFS (almost invariably 1 MB with VMFS 5). There is a slight performance impact on writes to new blocks due to the on-demand zeroing.

3. **Thin**—this format neither reserves space on the VMFS volume nor pre-zeroes blocks. Space is allocated and zeroed on-demand in segments in accordance to the VMFS block size. Thin virtual disks allow for the highest virtual machine density but provide the lowest protection against possible capacity exhaustion. There is a slight performance impact on writes to new blocks due to the on-demand zeroing behavior.

Prior to WRITE SAME support, the performance differences between these virtual disk allocation mechanisms were distinct. This was due to the fact that before an unallocated block could be written to, zeroes would have to be written first causing an allocate-on-first-write penalty (increased latency). Therefore, for every new block written to there were actually two writes; the zeroes then the actual data. For thin and zeroedthick virtual disks, this zeroing was on-demand so the penalty was seen by applications. For eagerzeroedthick, it was noticed during deployment because the entire virtual disk had to be zeroed prior to use. This zeroing caused unnecessary I/O on the SAN fabric, subtracting available bandwidth from “real” I/O.

To resolve this issue, VMware introduced WRITE SAME support. WRITE SAME is a SCSI command that tells a target device (or array) to write a pattern (in this case, zeroes) to a target location. ESXi utilizes this command to avoid having to actually send a payload of zeros but instead simply communicates to any array that it needs to write zeros to a certain location on a certain device. This not only reduces traffic on the SAN fabric, but also speeds up the overall process since the zeros do not have to traverse the data path.

This process is optimized even further on the Pure Storage FlashArray. Since the array does not store space-wasting patterns like contiguous zeros on the SSDs, the zeros are discarded and any subsequent reads will result in the array returning zeros to the host. This additional array-side optimization further reduces the time and penalty caused by pre-zeroing of newly-allocated blocks.

The addition of WRITE SAME and further FlashArray optimizations, ESXi zeroing behaviors that cause differences in performance in the various virtual disks is a thing of the past.
Dead Space Reclamation or UNMAP

In block-based storage implementations, the file system is managed by the host, not the array. Because of this, the array does not typically know when a file has been deleted or moved from a storage volume and therefore does not know when to release the space. This behavior is especially detrimental in thinly-provisioned environments (which today is nearly ubiquitous) where that space could be immediately allocated to another device/application or just returned to the pool of available storage.

In vSphere 5.0, VMware introduced Dead Space Reclamation which makes use of the SCSI UNMAP command to help remediate this issue. UNMAP enables an administrator to initiate a reclaim operation from an ESXi host. In turn, ESXi informs the storage array of the space that previously had been occupied by a virtual disk and is now free and it can be reclaimed. This enables an array to accurately manage and report space consumption of a thinly-provisioned datastore and enables users to better monitor and forecast new storage requirements.

The UNMAP process is not automatic—it is only available via command line tools in all versions of ESXi 5.x and 6.0.

To reclaim space in vSphere 5.0 U1 through 5.1, SSH into the ESXi console and run the following commands:

1. Change into the directory of the VMFS datastore you want to run a reclamation on:
   ```
cd /vmfs/volumes/<datastore name>
   ```

Then run vmkfstools to reclaim the space by indicating the percentage of the free space you would like to reclaim (up to 99%):

   ```
   vmkfstools -y 99
   ```

To reclaim space in vSphere 5.5 and 6.0, the vmkfstools -y option has been deprecated and UNMAP is now available in esxcli. UNMAP can be run anywhere esxcli is installed and therefore does not require an SSH session:

1. Run esxcli and supply the datastore name. Optionally a block iteration count can be specified, otherwise it defaults to reclaiming 200 MB per iteration:
   ```
esxcli storage vmfs unmap -l <datastore name> -n (blocks per iteration)
   ```

The esxcli option can also be leveraged from the VMware vSphere PowerCLI using the cmdlet GetEsxCl:

   ```
$esxcli=get-esxcli -VMHost <ESXi host name>
$esxcli.storage.vmfs.unmap(<block count>, "<datastore name>", $null)
   ```
Choose an ESXi host for the VMHost for the first line. In the second line, the first parameter is the block count, the second parameter is the target datastore name and the third parameter can remain null (unless you prefer to specify a UUID instead of a datastore name to “future proof” scripts against name changes).

It should be noted that while UNMAP on the FlashArray is a quick and unobtrusive process, ESXi does create a balloon file, when using the vmkfstools method, to fill the entire specified range of free space for the duration of the reclaim process. This could possibly lead to a temporary out-of-space condition on the datastore and could detrimentally affect virtual machines that have thin virtual disks on that datastore that need to grow during the UNMAP operation. When a datastore contains a large amount of thin virtual disks, large UNMAP reclaim percentages should be entered with care. Please note though that in ESXi 5.5+ the new esxcli process does not have this issue.

The esxcli version of UNMAP allows the user to delineate the number of blocks unmapped per iteration of the process. The default value is 200 (in other words 200 MB) and can be increased or decreased as necessary. Since the UNMAP process is a workload of negligible impact on the Pure Storage FlashArray, this value can be increased to dramatically reduce the duration of the reclaim process. The time to reclaim reduces exponentially as the block count increases. Therefore, increasing the block count to something sufficiently higher is recommended.

While the FlashArray can handle very large values for this, ESXi does not support increasing the block count any larger than 1% of the free capacity of the target VMFS volume (one block equals one megabyte). Consequently, the best practice for block count during UNMAP is no greater than 1% of the free space. So as an example, if a VMFS volume has 1,048,576 MB free, the largest block count supported is 10,485 (always round down). If you specify a larger value the command will still be accepted, but ESXi will override the value back down to the default of 200 MB, which will profoundly slow down the operation.

! It is imperative to calculate the block count value based off of the 1% of the free space only when that capacity is expressed in megabytes—since VMFS 5 blocks are 1 MB each. This will allow for simple and accurate identification of the largest allowable block count for a given datastore. Using GB or TB can lead to rounding errors, and as a result, too large of a block count value. Always round off decimals to the lowest near MB in order to calculate this number (do not round up).

In order to see if the value was overridden or not, you can check the hostd.log file in the /var/log/ directory on the target ESXi host. For every UNMAP operation there will be a series of messages that dictate the block count for every iteration. Examine the log and look for a line that indicates the UUID of the VMFS volume being reclaimed, the line will look like the example below:

Unmap: Async Unmapped 5000 blocks from volume 545d6633-4e026dce-d8b2-90e2ba392174

! From ESXi 5.5 Patch 3 and later, any UNMAP operation against a datastore that is 75% or more full will use a block count of 200 regardless to any block count specified in the command. For more information refer to the VMware KB article here.
The UNMAP procedure (regardless of the ESXi version) removes the metadata pointers to the data for that given volume and if no other pointers exist—the data is also tagged for garbage collection. Therefore, capacity numbers may not necessarily change on the array after an UNMAP operation. If the data is heavily deduplicated, it is highly possible that the data that was reclaimed is still in use by another volume or other parts within the same volume. In that case, the specific metadata pointers are removed but the data itself remains, since it is still in use by other pointers. So data reduction ratios may change, but not raw used capacity. That being said, it is still important to UNMAP regularly to make sure stale pointers do not remain in the system. Regular reclamation procedures allow this data to eventually be removed as the remaining pointers are deleted.

In-Guest UNMAP in ESXi 6.x

The discussion above speaks only about space reclamation directly on a VMFS volume. This pertains to dead space accumulated by the deletion or migration of virtual disks, ISOs or swap files (mainly). The CLI-initiated UNMAP operation does not pertain though to dead space accumulated inside of a virtual disk. Dead space accumulates inside of a virtual disk in the same way that it accumulates on a VMFS volume—deletion or movement of files.

Prior to ESXi 6.0 and virtual machine hardware version 11, guests could not leverage native UNMAP capabilities on a virtual disk because ESXi virtualized the SCSI layer and did not report UNMAP capability to the guest.

In ESXi 6.0, guests running in a virtual machine using hardware version 11 can now issue UNMAP directly to thin virtual disks. The process is as follows:

1. A guest application or user deletes a file from a file system residing on a thin virtual disk
2. The guest automatically (or manually) issues UNMAP to the guest file system on the virtual disk
3. The virtual disk is then shrunk in accordance to the amount of space reclaimed inside of it.
4. If EnableBlockDelete is enabled, UNMAP will then be issued to the VMFS volume for the space that previously was held by the thin virtual disk. The capacity is then reclaimed on the FlashArray.

Currently, in-guest UNMAP support is limited to Windows 2012 R2 or Windows 8. Linux requires a newer version of SCSI support that is under consideration for future versions of ESXi.

Prior to ESXi 6.0, the parameter EnableBlockDelete was a defunct option that was previously only functional in very early versions of ESXi 5.0 to enable or disable automated VMFS UNMAP. This option is now functional in ESXi 6.0 and has been re-purposed to allow in-guest UNMAP to be translated down to the VMFS and accordingly the SCSI volume. By default, EnableBlockDelete is disabled and can be enabled via the Web Client or CLI utilities.
In-guest UNMAP support does actually not require this parameter to be enabled though. Enabling this parameter allows for end-to-end UNMAP or in other words, in-guest UNMAP commands to be passed down to the VMFS layer. For this reason, enabling this option is a best practice for ESXi 6.x and later.

The process to issue UNMAP from guest applications is beyond the scope of this document. Please refer to guest OS documentation for UNMAP support information.

Verifying that VAAI is Enabled

In ESXi 5.x hosts, to determine whether or not VAAI is enabled, use the service console in ESXi or the vCLI in ESXi, and run these command to check if the Int Value is set to 1 (enabled):

```
esxcli system settings advanced list -o /DataMover/HardwareAcceleratedMove
esxcli system settings advanced list -o /DataMover/HardwareAcceleratedInit
esxcli system settings advanced list -o /VMFS3/HardwareAcceleratedLocking
```

You will see an output similar to:

```
Path: /VMFS3/HardwareAcceleratedLocking
Type: integer
Int Value: 1  ➙ Value is 1 if enabled
Default Int Value: 1
```

**IMPORTANT**: Pure Storage does not support DISABLING VAAI features on ESXi hosts—if a customer is required to disable it for any reason they must contact Pure Storage support if the affected ESXi hosts have access to Pure Storage FlashArray devices.
Hardware acceleration is enabled by default and requires no work on the array or ESXi to use out of the box. In the case it was somehow disabled, follow these steps to re-enable the primitives:

To enable atomic test and set (ATS) AKA hardware accelerated locking:

```
esxcli system settings advanced set -i 1 -o /VMFS3/HardwareAcceleratedLocking
```

To enable Hardware accelerated initialization AKA WRITE SAME:

```
esxcli system settings advanced set --int-value 1 --option /DataMover/HardwareAcceleratedInit
```

To enable Hardware accelerated data move AKA XCOPY (full copy):

```
esxcli system settings advanced set --int-value 1 --option /DataMover/HardwareAcceleratedMove
```

The figure below describes the above steps pictorially using the vSphere Web Client. Go to an ESXi host and then Settings, then Advanced System Settings and search for “Hardware”

![vSphere Web Client](image)

**Figure 30. VAAI advanced options in the vSphere Web Client**

**How to Confirm XCOPY Operations are Working**

Run esxtop on the ESXi server or resxtop from the vMA and use the options “u”, “f”, “o” to watch for the fields CLONE_RD and CLONE_WR which show IOPS, and MBC_RD/s and MBC_WR/s which show throughput, they should all increment during the attempted XCOPY session. If they are not incrementing and CLONE_F (failed clones) is (which should never happen), something is interrupting the XCOPY session and causing the operation to revert to software copy. Refer to the vmkernel log to find more information.
**Pure Storage Management Pack for VMware vRealize Operations**

Traditional operations management systems do not meet the requirements of today’s virtual and cloud infrastructures. They make IT too reactive because they lack the intelligence to aggregate, correlate, and analyze metrics across applications and infrastructure stacks.

vRealize Operations is built on a scale-out, resilient platform designed to deliver intelligent operational insights to simplify and automate management of applications and infrastructure across virtual, physical and cloud environments—from vSphere to Hyper-V, Amazon Web Services (AWS), and more.

With vRealize Operations, IT organizations of all sizes can improve performance, avoid business disruption, and become more efficient with comprehensive visibility across applications and infrastructure in one place.

vRealize Operations delivers

- Intelligent operations – Self-learning tools, predictive analytics, and Smart Alerts about application and infrastructure health enable proactive identification and remediation of emerging performance, capacity, and configuration issues.

- Policy-based automation – Out-of-the-box and customizable policies for critical IT operations are associated with Smart Alerts, guided remediation, and compliance standards to deliver recommendations, or trigger actions, that optimize performance and capacity and enforce configuration standards.

- Unified management – An open and extensible platform, supported by third-party management packs for Microsoft, SAP, and others, provides complete visibility in a single console across applications, storage, and network devices.

The Pure Storage FlashArray Management Pack for VMware vRealize Operations provides comprehensive view of the storage array inventory, alerts, capacity, and performance statistics in context of compute resources. Based on collected information vROPS proactively identifies emerging issues with predictive analytics and smart alerts, ensuring optimum performance and availability of applications.

- Dependency hierarchical maps for mapping virtual compute to the storage array objects.

- Heat maps for capacity and performance

- Charts for capacity and performance

- Alerting with KB descriptions of the action
The Pure Storage FlashArray Management Package for VMware vRealize Operations allows for end-users to analyze, report and project capacity and performance-based information on their FlashArray in a VMware-context. Through the vRealize platform, administrators can quickly identify heavy-hitters for capacity or performance to troubleshoot or trend for future use. Leveraging either the build in dashboards for capacity or performance, or creating their own.

A variety of statistics are ingested for the following FlashArray objects and then merged and related in-context to the VMware vSphere environment:

- FlashArrays
- Host Groups
- Hosts
- Volumes

Gathered statistics include, but are not limited to:

- Data Reduction
- Used Capacity
- Provisioned Space
• Snapshot Usage
• IOPS
• Throughput
• Latency

Leveraging the built-in REST API on the FlashArray, this package works without any management appliance or additional configuration outside of importing into vRealize itself. The pack requires:

• VMware vRealize Operations Management 6.x
• Pure Storage FlashArray 400 series (405, 420 or 450) and FlashArray//m (m20, m50, m70)
• Purity Operation Environment 4.0.7 or later

The pack can be downloaded [here](#).
Pure Storage Content Pack for VMware vRealize Log Insight

VMware vRealize Log Insight provides real-time log administration for heterogeneous environments that span across physical, virtual and cloud environments. Log Insight provides:

- Universal log collection
- Powerful log analytics
- Enterprise-class scalability
- Ease of use and deployment
- Built-in vSphere knowledge

Log Insight collects and analyzes all types of machine-generated log data, including application logs, network traces, configuration files, messages, performance data and system state dumps. Administrators can connect it to everything in their environment—operating systems, applications, storage, firewalls, and network devices etc. — for enterprise-wide visibility via log analytics.

VMware Log Insight allows partners to create integration plugins referred to as Content Packs to provide additional intelligence into their products. Content Packs are customized by various partners to be distributed to users of Log Insight that include custom queries, dashboards, alerts and fields. These custom properties are created in context of the source system sending syslog messages (whether it be a storage array, an application or other device). By allowing partners to create these Content Packs, customers can easily begin to leverage Log Insight with their source IT objects with little configuration. Content Packs reduce initial configuration because the partner has created them with the most important metrics, events and alerts in mind therefore doing most of the legwork for you. The partners know the syntax, fields and the message types their systems send. So the Content Pack can intelligently decide what is most important, what should be noted and how they can be displayed. This is achieved by built-in Log Insight objects such as queries and dashboards in the Content Pack. Customers then can just plug-in the Content Pack and begin analyzing their environment.

The Pure Storage Content Pack includes:

- Four dashboard groups offering 22 dashboard widgets
- Twenty-seven pre-defined extracted fields
- Six custom alerts

The Pure Storage Content Pack requires the following:

- VMware vRealize Log Insight 2.5
- Pure Storage FlashArray 400 series (405, 420 or 450) and FlashArray//m (m20, m50, m70)
- Purity 4.0+ (certain features will need Purity 4.1)
For more information on the configuration and use of the Pure Storage Content Pack for VMware vCenter Log Insight please refer to the white paper on http://www.purestorage.com/resources/datasheets-whitepapers.html.
Figure 34. Pure Storage Content Pack Audit Dashboard
Storage Replication Adapter for vCenter Site Recovery Manager

With the accelerated deployment of server virtualization technology in the data center, more business critical applications run on virtual machines and require support from business disaster recovery processes. VMware vCenter Site Recovery Manager (SRM) is a disaster recovery application that automates the workflow of failing over protected virtual machines from one data center to another. SRM manages the following processes for vCenter environments, including automation of several steps:

- Configuration of a recovery solution, including the protected site and the recovery site
- Specification of the recovery plan
- Recovery plan verification without disruption of the production environment
- Failover to a recovery site in the event of a data center disaster
- Planned migration of virtual machines from one site to another site, supporting data center mobility and disaster avoidance

The SRM disaster recovery solution supports a vendor-supplied plugin, known as a Storage Replication Adapter (SRA), to leverage the underlying storage array's replication technology during SRM operations. The Pure Storage SRA ensures the integration of both FlashArray storage and FlashRecover replication with Site Recovery Manager.

The FlashArray's replication technologies offer these advantages:

- Native replication with no additional license requirement
- Flexible, policy-based automation of replication
- Simple configuration that does not require professional services involvement
- FlashArray’s unique data reduction (pattern elimination, deduplication, and compression) and ultra-efficient micro-provisioning of physical storage
- Support for bi-directional protection

These FlashArray features offer savings in both original purchase price and in operational time, including replication time and Recovery Time Objective (RTO). VMware SRM and the FlashArray SRA combine to provide a complete disaster recovery solution based on an industry-leading all-flash technology, while reducing the disaster recovery budget through the efficient use of time and resources. The automation achieved with SRM and SRA vastly improves an organization's RTO.
The Pure Storage FlashArray Storage Replication Adapter is supported with VMware vCenter Site Recovery Manager 5.5, 5.8 and 6.0. The SRA requires FlashArray 400 Series or FlashArray//m running Purity 4.0 or later. It is recommended to use Purity 4.0.20 or later.

For details on the SRA, please refer to the user guide available on [http://community.purestorage.com](http://community.purestorage.com). The user guide is also packaged with the SRA installation files on the VMware website.

For best replication performance, Purity 4.1.7 or later is highly recommended.
Appendix I: Per-Device NMP Configuration

This appendix describes how to check and/or set Round Robin and an IO Operation Limit of 1 on existing devices.

The first step is to change the particular device to use the Round Robin PSP. This must be done on every ESXi host and can be done with through the vSphere Web Client, the Pure Storage Plugin for the vSphere Web Client or via command line utilities.

```bash
esxcli storage nmp device set -d naa.<device NAA> --psp=VMW_PSP_RR
```

Figure 36. Setting Round Robin with the Pure Storage Web Client Plugin
Note that changing the PSP using the Web Client Plugin is the preferred option as it will automatically configure Round Robin across all of the hosts. Note that this does not set the IO Operation Limit it 1 which is a command line option only—this must be done separately.

Round Robin can also be set on a per-device, per-host basis using the standard vSphere Web Client actions. The procedure to setup Round Robin policy for a Pure Storage volume is shown in the below figure. Note that this does not set the IO Operation Limit it 1 which is a command line option only—this must be done separately.

To set a device that is pre-existing to have an IO Operation limit of one, run the following command:

```
esxcli storage nmp psp roundrobin deviceconfig set -d naa.<naa address> -l 1 -t iops
```
Appendix II: Deleting a Volume Manually

Prior to the deletion of a volume, ensure that all important data has been moved off or is no longer needed. From the vSphere Web Client (or CLI) delete or unmount the VMFS volume and then detach the underlying device from the appropriate host(s).

After a volume has been detached from the ESXi host(s) it must first be disconnected (from the FlashArray perspective) from the host within the Purity GUI before it can be destroyed (deleted) on the FlashArray. Though note that Purity will not allow a volume to be destroyed that is still connected to one or more hosts and as any attempt to do so will return an error. The two steps can be summarized in the following figures:

Figure 38.Unmounting a VMFS volume in the vSphere Web Client
1. Disconnect the volume from host
   
   ```
   purehost disconnect --vol test_vol esx_host1
   OR
   puregroup disconnect --vol test_vol esx_host1
   ```

2. Destroy a volume on FlashArray
   
   ```
   purevol destroy test_vol
   ```

---

**Figure 39.** Detaching a device in the vSphere Web Client

**Figure 40.** Disconnecting a volume from a host group using the Pure GUI
By default a volume can be recovered after deletion for 24 hours to protect against accidental removal. This entire removal and deletion process is automated through the Pure Storage Plugin for the vSphere Web Client and its use is therefore recommended.

Figure 41. Deleting a volume with the Pure GUI

Figure 42. Deleted volumes in the Pure GUI
Appendix III: Configure End-to-End Jumbo Frames

In some iSCSI environments it is required to enable jumbo frames to adhere with the network configuration between the host and the FlashArray. Enabling jumbo frames is a cross-environment change so careful coordination is required to ensure proper configuration. It is important to work with your networking team and Pure Storage representatives when enabling jumbo frames. Please note that this is not a requirement for iSCSI use on the Pure Storage FlashArray. This is a fully supported yet optional change that it up to the discretion of the user.

1. Configure jumbo frames on the FlashArray iSCSI ports using the GUI or CLI.

   ```
   purenetwork -setattr -mtu <MTU> <Ethernet-interface>
   ```

2. Configure jumbo frames on the physical network switch/infrastructure for each port using the relevant switch CLI or GUI.

3. Configure jumbo frames on the appropriate VMkernel network adapter and vSwitch.

   A. Browse to a host in the vSphere Web Client navigator.
   B. Click the Manage tab, and select Networking > Virtual Switches.
   C. Select the switch from the vSwitch list.

Figure 43. Configuring Jumbo Frames on the Pure Storage FlashArray
D. Click the name of the VMkernel network adapter.
E. Click the pencil icon to edit.
F. Click NIC settings and set the MTU to your desired value.
G. Click OK.
H. Click the pencil icon to edit on the top to edit the vSwitch itself.
I. Set the MTU to your desired value.
J. Click OK.

Once jumbo frames are configured, verify end-to-end jumbo frame compatibility. To verify, try to ping a route with vmkping.

```
vmkping -s 9000 <ip address of Pure Storage iSCSI port>
```

If the ping operations does not return successfully, then jumbo frames are not properly configured on one of the hosts, networking devices, or storage arrays.
Appendix IV: Best Practices PowerCLI Script

This script does the following:

- Checks the transfer size setting and if not 16 MB changes it to 16 MB and enables EnableBlockDelete on all ESXi 6.0 hosts.
- Sets the I/O Operation Limit parameter to 1.
- Checks for an existing Pure Storage SATP Rule and if it finds one, it checks to see if it is configured for Round Robin and if the I/O Operations Limit is configured.
- If it finds a SATP rule that is correct it leaves it, if it doesn’t it will delete it create one correctly.
- If no SATP rule is found it creates a new one for Round Robin and an I/O Operations of 1.

Since existing devices will not inherit NMP configurations from a SATP rule until the host is rebooted, the script looks for existing Pure Storage volumes and checks them to see if they are using Round Robin and if their I/O Operations Limit is correct, if it isn’t the script will fix them.

The vCenter server is hard coded so you will need to enter them into the script or add a prompt for what vCenter/credentials are needed. After that it will work through each ESXi server in that vCenter. You can easily change this to just a cluster or a host.

Note that this was written in PowerCLI 6.0, so older versions may be different.

For all this and other PowerCLI scripts, please navigate to:

https://github.com/codyhosterman/powercli
References

2. Disabling VAAI Thin Provisioning Block Space Reclamation (UNMAP) in ESXi 5.0 - kb.vmware.com/kb/2007427

Acknowledgements

Special thanks to Kyle Grossmiller, Craig Waters, Jon Owings, Benjamin Troch and Craig Chapman for their contributions to this document.
About the Author

Cody Hosterman is a VMware-focused Solutions Architect at Pure Storage since 2014. His primary responsibility is overseeing, testing, documenting, and demonstrating VMware-based integration with the Pure Storage FlashArray platform. Cody has been working in vendor enterprise storage/VMware integration roles since 2008.

Cody graduated from the Pennsylvania State University with a bachelors degree in Information Sciences & Technology in 2008. Special areas of focus include core ESXi, vRealize, Site Recovery Manager and PowerCLI. Cody has been named a VMware vExpert from 2013 through 2015.

Blog: [http://www.codyhosterman.com](http://www.codyhosterman.com)

Twitter: [@codyhosterman](https://twitter.com/codyhosterman)