



Technical Report

Best Practices for File System Alignment in Virtual Environments

Abhinav Joshi, Eric Forgette, Peter Learmonth | NetApp
March 2009 | TR-3747

Version 1.0

ABSTRACT

This document provides guidelines for preventing, detecting, and correcting file system misalignment issues for virtual machines hosted on VMware® ESX, Microsoft® Hyper-V™, and Citrix XenServer infrastructures.

TABLE OF CONTENTS

1	EXECUTIVE SUMMARY	3
2	NETAPP STORAGE AND VIRTUAL ENVIRONMENTS DATA LAYOUT CONCEPTS	3
2.1	VMWARE.....	3
2.2	MICROSOFT HYPER-V	4
2.3	CITRIX XENSERVR	4
3	FILE SYSTEM ALIGNMENT	5
3.1	CONCEPTS AND ISSUE	5
3.2	IMPACT	7
4	FILE SYSTEM MISALIGNMENT PREVENTION	8
4.1	VMWARE.....	8
4.2	MICROSOFT HYPER-V	9
4.3	CITRIX XENSERVR	11
4.4	GUEST/CHILD VM ALIGNMENT PROCEDURE.....	12
5	FIXING FILE SYSTEM ALIGNMENT	18
5.1	DETECTION	18
5.2	CORRECTION.....	20
6	CONCLUSION	23
7	FEEDBACK.....	23
8	REFERENCES.....	24
9	VERSION HISTORY	24
10	APPENDIX	24

1 EXECUTIVE SUMMARY

File system misalignment is a known issue in virtual environments and can cause performance issues for virtual machines (VMs). This document provides an overview of the storage layers in a virtualized environment, provides details on the proper alignment of guest file systems, and describes the performance impact misalignment can have on the virtual infrastructure. It also provides guidance on how the issue can be prevented by following proper steps when provisioning storage for the VM. For existing VMs, this document provides guidelines on how to determine and correct misalignment issues.

Please note that this issue is not unique to NetApp® storage arrays and can occur with any storage array from any vendor. VMware has also identified that this can be an issue in virtual environments. For more details please refer to the VMware article [VMware Infrastructure 3 Recommendations for Aligning VMFS Partitions](#).

2 NETAPP STORAGE AND VIRTUAL ENVIRONMENTS DATA LAYOUT CONCEPTS

In any server virtualization environment using shared storage, there are different layers of storage involved for the VMs to access storage. In this section we will explore the different ways shared storage can be presented for the different hypervisors and also highlight the different layers of storage involved.

Please note that the scope of this paper is limited to VMware ESX, Microsoft Hyper-V, and Citrix XenServer environments.

2.1 VMWARE

VMware Virtual Infrastructure (VI) has four ways of using shared storage for deploying virtual machines:

- **VMFS** (Virtual Machine File System) on a Fibre Channel or iSCSI LUN attached to the ESX host
- **NFS** (Network File System) export mounted on an ESX host
- **RDM** (Raw Device Mapping) is the primary method of presenting a VM direct access and ownership of a LUN; the guest formats the RDM LUN as it would for any disk
- **LUNs directly mapped by the guest OS** by using an iSCSI software initiator where the guest OS supports it

For both the VMFS and NFS options, the files that make up a VM are stored in a directory on the LUN or NFS export and each VM will have a separate directory. Each virtual disk of the VM is made up of two files:

- **<vmname>-flat.vmdk**, which is the monolithic file containing the actual disk image of the guest VM
- **<vmname>.vmdk**, which is a text descriptor file that contains information about the size of the virtual disk as well as cylinder, head, and sector information for the virtual BIOS to report to the guest operating system (OS)

Both VMFS and NFS mounted by ESX are referred to as datastores. For the NFS option, there is no VMFS layer and the VM directories are directly stored on the NFS mount presented as a datastore.

The different layers of storage involved for each of the shared storage options are shown in Table 1 below. The check mark (✓) indicates that alignment should be ensured at this layer of guest OS or hypervisor file system or the NetApp storage array blocks.

Table 1) Layers of storage for VMware shared storage options.

Layers of Storage	VMware Shared Storage Options			
	VMFS-Formatted LUN	NFS Export	RDM LUN	LUNs Directly Mapped by the Guest OS
Guest OS	✓	✓	✓	✓
VMware ESX	✓	N/A	N/A	N/A
NetApp Storage Array	✓	N/A	✓	✓

For example, for the VMFS option, there are three layers involved—the guest OS, VMware ESX, and the NetApp storage array and alignment must be ensured at all these levels as indicated.

2.2 MICROSOFT HYPER-V

Microsoft Windows® Server 2008 Hyper-V has three ways of using shared storage for deploying VMs:

- **NTFS-formatted LUNs** (Fibre Channel or iSCSI) attached to the Hyper-V parent partition as physical disks. The VMs are represented by VHD files hosted on the LUNs. There are three different types of VHDs:
 - **Fixed-size VHD:** This type of VHD allocates the full amount of storage configured at VHD creation and does not expand over time. It offers the lowest performance overhead of all three VHD variants and is the NetApp recommended best practice.
 - **Dynamically expanding VHD:** This type of VHD does not allocate the full amount of storage configured at the time of VHD creation and expands over time as new data is added to the VM's disk. This type of VHD differs mostly in the area of performance because of the impact associated with having to grow the VHD file each time data is added to the VM's disk.
 - **Differencing VHD:** This type of VHD is created not at the time the VM is created, but for example, when a Hyper-V snapshot is made of an existing VM. A differencing VHD points to a parent VHD file, which can be any type of VHD, and functions similar to a dynamically expanding VHD.
- **Pass-through disks** are LUNs that are attached to the Hyper-V parent partition but assigned directly to a VM and formatted with the child OS file system.
- **LUNs directly mapped to the child OS** by using an iSCSI software initiator where the child OS supports it.

The different layers of storage involved for each of the shared storage options are shown in Table 2 below. The check mark (✓) indicates that alignment should be ensured at this layer of guest or hypervisor file system or the NetApp storage array blocks.

Table 2) Layers of storage for Microsoft Hyper-V shared storage options.

Layers of Storage	Hyper-V Shared Storage Options		
	NTFS-Formatted LUNs	Pass-through Disks	LUNs Directly Mapped to the Child OS
Child OS	✓	✓	✓
Hyper-V Parent Partition	✓	N/A	N/A
NetApp Storage Array	✓	✓	✓

2.3 CITRIX XENSERVER

The Citrix XenServer host accesses containers named Storage Repositories (SRs) in which Virtual Disk Images (VDIs) are stored. A VDI is a disk abstraction that, when attached to a XenServer host, appears as a physical disk drive to the VM. The interface to storage hardware provided on the XenServer host allows VDIs to be supported on a large number of different SR substrate types. VDIs may be files on a local disk, on an NFS mount, Logical Volumes within a LUN, or a raw LUN itself directly attached to the VM.

When hosting shared SRs on a NetApp storage controller, the different options to provision storage are:

- **NetApp managed LUNs:** LUNs hosted on a NetApp storage array are accessible via the NetApp Data ONTAP® SR type and are hosted on NetApp storage running a version of Data ONTAP 7.0 or greater. LUNs are allocated on demand via the XenServer management interface and mapped dynamically to the XenServer host via the XenServer host management framework (using the open iSCSI software initiator) while a VM is active. All the thin provisioning, FlexClone®, and data deduplication capabilities in the NetApp storage controllers are available via the NetApp Data ONTAP adapter. Please see [TR 3732—Citrix XenServer 5.0 and NetApp Storage Best Practices](#) for further details.

- **NFS** export mounted as SR on the Xen Master Server using XenServer host. In this option, the VHD format is used to store VDIs on an NFS mount exported from a NetApp storage array. There are two types of VHDs:
 - **Sparse VHD:** This is the default type of VHD created when using the NFS SR. It does not fully provision the storage upfront at the time of creating the VHD.
 - **Chained VHD:** This type of VHD allows two VDIs to share common data. In cases in which an NFS-based VM is cloned, the resulting VMs will share the common on-disk data at the time of cloning.

LUNs with the Logical Volume Manager (LVM) layer: Shared storage can be provided using a Logical Volume Manager layered over either a Fibre Channel or iSCSI LUN hosted on a NetApp storage controller and accessed via Fibre Channel HBAs or iSCSI initiators (hardware or software).

The different layers of storage involved for each of the options are shown in Table 3 below. The check mark (✓) indicates that alignment should be ensured at this layer of guest or hypervisor file system or NetApp storage array blocks.

Table 3) Layers of storage for Citrix XenServer shared storage options.

Layers of Storage	Citrix XenServer Shared Storage Option		
	NetApp Managed LUNs	NFS Export	LUNs with the LVM Layer
Guest OS	✓	✓	✓
Citrix XenServer Control Domain	N/A	N/A	N/A
NetApp Storage Array	✓	N/A	✓

3 FILE SYSTEM ALIGNMENT

As highlighted in the previous section, there are multiple layers of storage involved. Each layer is organized into blocks or “chunks,” to make accessing the storage more efficient. The size and the starting offset of each of these blocks can be different at each layer. While a different block size across the storage layers doesn’t require any special attention, the starting offset does. For optimal performance, the starting offset of a file system should align with the start of a block in the next lower layer of storage. For example, an NTFS file system that resides on a LUN should have an offset that is divisible by the block size of the storage array presenting the LUN. Misalignment of block boundaries at any one of these layers of storage can result in performance degradation.

Please note that this issue is not unique to NetApp storage arrays and can occur for storage arrays from any vendor. VMware has also identified that this can be an issue in virtual environments. For more details please refer to the VMware article [VMware Infrastructure 3 Recommendations for Aligning VMFS Partitions](#).

3.1 CONCEPTS AND ISSUE

Historically, hard drives (LUNs) presented the OS with a logical geometry that would be used to partition and format the disk in an efficient manner. Logical geometry today is virtual and fabricated by the host BIOS and operating system. Operating partitioning programs such as *fdisk* use the fake disk geometry to determine where to begin a partition. Unfortunately, some partitioning programs create disk partitions that do not align to underlying block boundaries of the disk. Notable examples of this are the GNU *fdisk* found on many Linux® distributions and Microsoft Diskpart found on Windows 2000 and Windows 2003. For more detailed information, please refer to the appendix at the end of this document.

NetApp uses 4KB blocks (4 x 1,024 = 4,096 bytes) as its basic storage building block. Write operations can consume no less than a single 4KB block and can consume many 4KB blocks depending on the size of the write operation. Ideally, the guest/child OS should align its file system(s) such that writes are aligned to the storage device’s logical blocks. The problem of unaligned LUN I/O occurs when the partitioning scheme used by the host OS doesn’t match the block boundaries inside the LUN, as shown in Figure 1 below. If the guest file system is not aligned, it may become necessary to read or write twice as many blocks of storage than the guest actually requested, since any guest file system block actually occupies at least two partial storage blocks. As a simple example, assuming only one layer of file system and that the guest allocation unit is equal to the storage logical block size (4K or 4,096 bytes), each guest block (technically an “allocation

unit") would occupy 512 bytes of one block and 3,584 bytes (4,096 - 512) of the next. This results in inefficient I/O because the storage controller must perform additional work such as reading extra data to satisfy a read or write I/O from the host.

By default, many guest operating systems, including Windows 2000 and 2003 and various Linux distributions, start the first primary partition at sector (logical block) 63. The reasons for this are historically tied to disk geometry. This behavior leads to misaligned file systems since the partition does not begin at a sector that is a multiple of eight.

Please note that Windows Server 2008 and Vista default at 1,048,576 (which is divisible by 4,096) and do not require any adjustments.

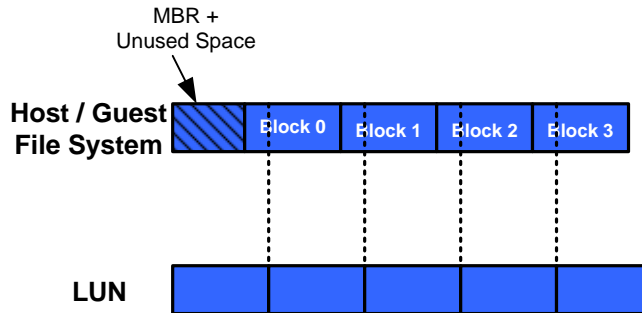


Figure 1) Misaligned file system.

This issue is more complex when the file system on the virtualization host contains the files (e.g., vmdk, vhd) that represent the VM virtual disks as shown in Figure 2 below. In this case, the partition scheme used by the guest OS for the virtual disks must match the partition scheme used by the LUNs on the hypervisor host and the NetApp storage array blocks.

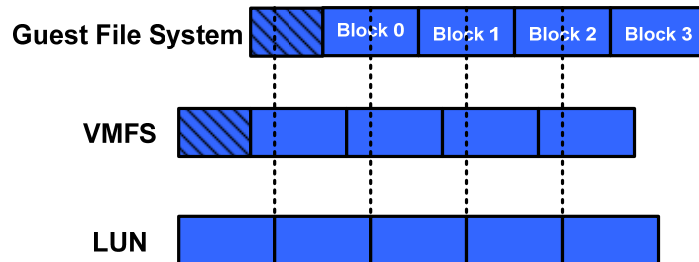


Figure 2) Guest OS and VMFS file system not aligned with the storage array blocks.

VMWARE

- In the case of VMs hosted on VMFS, there are two layers of alignment involved—the VMFS and the file system on the guest vmdk files inside the VMFS should align to the NetApp storage blocks. The default starting offset of VMFS2 is 63 blocks, which will result in misaligned I/O. The default offset of VMFS3, when created with the Virtual Infrastructure Client, is 128 blocks by default, which will NOT result in misaligned I/O. Datastores migrated from VMFS2 to VMFS3 as parts of an ESX/VI3 upgrade are not realigned; virtual machine files will need to be copied from the old datastore to a newly created datastore. In addition to properly aligning VMFS, each virtual machine guest file system will need to be properly aligned as well (see section 4.1).
- RDM and LUNs directly mapped by the guest VM do not require special attention if the “lun type” on the LUN matches the guest operating system type. For more information on “lun type,” see the LUN Multiprotocol Type section in the Data ONTAP Block Access Management Guide or Commands Manual Page Reference Document that can be downloaded from the [NetApp NOW™ \(NetApp on the Web\) site](#).

- While NFS datastores do not require alignment themselves, each virtual machine guest file system will need to be properly aligned (see section 4.1).

For all of these storage options, misalignment at any layer can cause performance issues as the system scales.

MICROSOFT HYPER-V

- For VHDs hosted on NTFS formatted LUNs attached as physical disks on the Hyper-V parent partition, there are two layers of alignment involved. The NTFS file system on the physical disk and the file system on the child VM hosted on the physical disk should align with the NetApp storage blocks.
- Pass-through disks and LUNs directly mapped by the child VM do not require special attention if the “lun type” of the LUN matches the guest operating system type. For more information on “lun type,” please see the LUN Multiprotocol Type section in the Data ONTAP Block Access Management Guide or Commands Manual Page Reference Document that can be downloaded from the [NetApp NOW site](#).

For all of these storage options, misalignment at any layer can cause performance issues as the system scales.

CITRIX XENSERVER

For all the storage options in Citrix XenServer discussed in section 2.3, there is only one layer of alignment involved—the file system on the guest VM should align with the NetApp storage blocks. Misalignment can result in performance issues.

3.2 IMPACT

Misalignment may cause an increase in per-operation latency. It requires the storage array to read from or write to more blocks than necessary to perform logical I/O. Figure 3 shows an example of a LUN with and without file system alignment. In the first instance, the LUN with aligned file systems uses four 4KB blocks on the LUN to store four 4KB blocks of data generated by a host. In the second scenario, where there is misalignment, the NetApp storage controller has to use five blocks to store the same 16KB of data. This is an inefficient use of space, and the performance suffers when the storage array has to process five blocks to read or write what should only be four blocks of data. This results in inefficient I/O, because the storage array is doing more work than is actually requested.

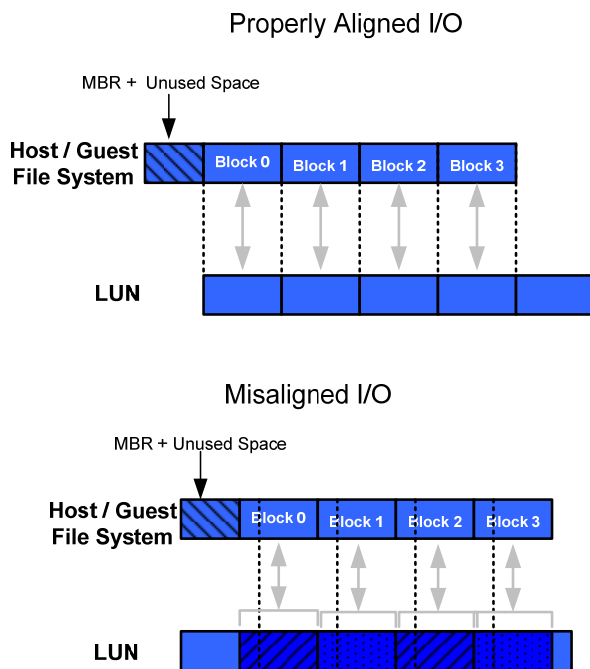


Figure 3) I/O impact for aligned and misaligned file systems.

4 FILE SYSTEM MISALIGNMENT PREVENTION

4.1 VMWARE

VMFS

VMFS-based datastores (FCP or iSCSI) should be set to the “lun type” VMware and created using the Virtual Infrastructure Client. This will result in the VMFS file system being aligned with the LUN on the storage controller. If you are using vmkfstools, please make sure that you first partition the LUN using fdisk. This will allow the correct offset to be set.

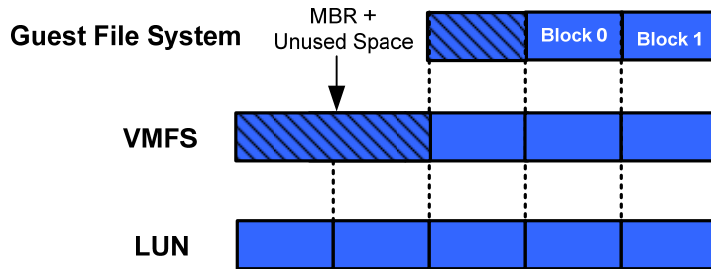


Figure 4) Guest VM file system and VMFS file system are aligned with the storage array blocks.

Unaligned VMFS partition can be detected using the procedure described below. For instructions on VMware VMFS partition alignment using fdisk, see the following VMware article: http://www.vmware.com/pdf/esx3_partition_align.pdf

VMs hosted on unaligned VMFS partitions should be migrated to aligned VMFS partition using cold migration or storage vmotion.

Step	Action
1.	<p>Get the vmhba device from the VMFS label (mount point) by running the following vmkfstools command:</p> <pre>vmkfstools -P /vmfs/volumes/vmprod</pre> <pre>VMFS-3.31 file system spanning 1 partitions. File system label (if any): vmprod Mode: public Capacity 214479929344 (204544 file blocks * 1048576), 12256804864 (11689 blocks) avail UUID: 47bd0f4d-8c6c0b00-202b-000423c3e841 Partitions spanned (on "lvm"): vmhba0:1:1:1</pre> <p>Note that vmkfstools returns the partition (ends in :1) rather than the whole disk or LUN device (which ends in :0)</p>
2.	<p>Use fdisk to check the starting offset of the VMFS partition.</p> <pre>fdisk -lu /vmfs/devices/disks/vmhba0:1:1:0</pre> <pre>Disk /vmfs/devices/disks/vmhba0:1:1:0: 214.7 GB, 214748364800 bytes 255 heads, 63 sectors/track, 26108 cylinders, total 419430400 sectors Units = sectors of 1 * 512 = 512 bytes</pre> <pre>Device Boot Start End Blocks Id System /vmfs/devices/disks/vmhba0:1:1:0p1 128 419425019 209712446 fb Unknown</pre> <p>Note that Start is listed as 128 sectors (physical disk blocks). The default in the GUI in VirtualCenter prior to 2.0 and from the command line in any version of ESX is 63.</p>

NFS

With NFS, there is no VMFS layer involved, so only the alignment of the guest VM file system within the VMDK to the NetApp storage array is required. This can be enabled using the procedure described in section 4.4.

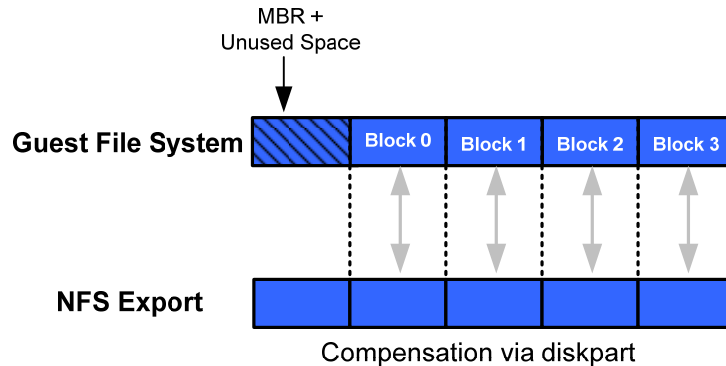


Figure 5) Guest VM file system aligned with the storage array blocks.

RDM

For RDM, selecting the “LUN type” of the intended guest OS when creating the LUN will enable the file system on the LUN to align with the blocks on the NetApp storage array.

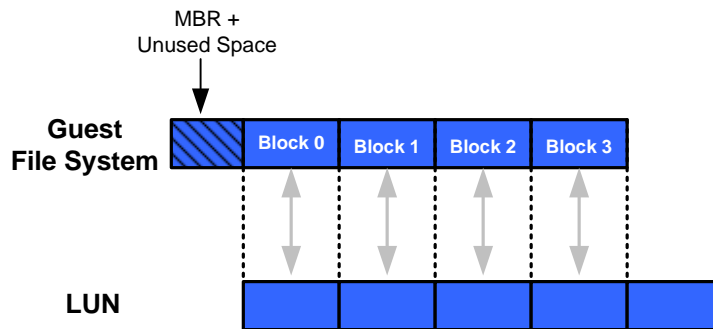


Figure 6) Guest VM file system aligned with the storage array blocks.

LUNS DIRECTLY MAPPED TO THE GUEST OS

For LUNs directly mapped to the guest OS using the iSCSI software initiator, selecting the LUN type of the intended guest OS when creating the LUN will enable the file system on the LUN to align with the NetApp storage array blocks, as shown in Figure 6 above.

4.2 MICROSOFT HYPER-V

NTFS-FORMATTED LUNS

For NTFS-formatted LUNs attached to the Hyper-V parent partition as physical disks hosting VHDs, selecting the correct LUN type is very important. For NetApp storage systems running Data ONTAP version 7.3.1 and higher, LUN type “Hyper-V” should be used.

Note: For NetApp storage systems running Data ONTAP 7.2.5 through 7.3, the LUN type “windows_2008” should be used. For NetApp storage systems running Data ONTAP version 7.2.4 and earlier, LUN type “linux” should be used.

For Data ONTAP version 7.3 and earlier, the LUN type “windows_2008” is available only through the Data ONTAP CLI. Therefore, the LUNs for Hyper-V parent partition and Windows Server 2008 Child VMs must be created through the LUN setup command on the Data ONTAP CLI.

Using NetApp SnapDrive® 6.0 and higher for provisioning LUNs will enable the “windows_gpt” LUN type to be selected and ensure alignment. However, the alignment of the file system of the child VM to the file system of the underlying physical disk will still be required. This can be enabled using the procedure described in section 4.4.

Fixed-Size VHDs: In the case of dynamically expanding and differencing VHDs, proper alignment cannot be guaranteed and there is a performance penalty. Therefore, NetApp recommends utilizing fixed-size VHDs within your Hyper-V environment whenever possible, avoiding use of dynamically expanding and differencing VHDs unless a good reason is found for their use. For further details on this recommendation, please refer to [TR-3702—NetApp and Microsoft Virtualization Storage Best Practices](#).

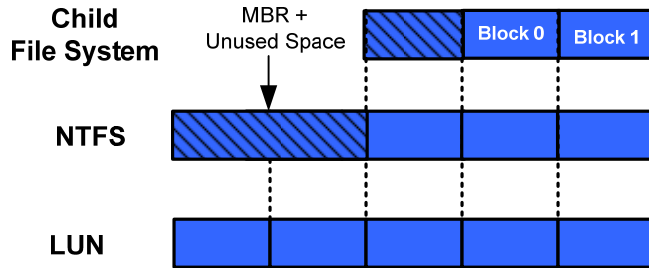


Figure 7) Child VM file system and NTFS file system are aligned with the storage array blocks.

PASS-THROUGH DISKS

For pass-through disks, selecting the LUN type of the intended child OS when creating the LUN will enable the file system on the LUN to align with the NetApp storage array blocks.

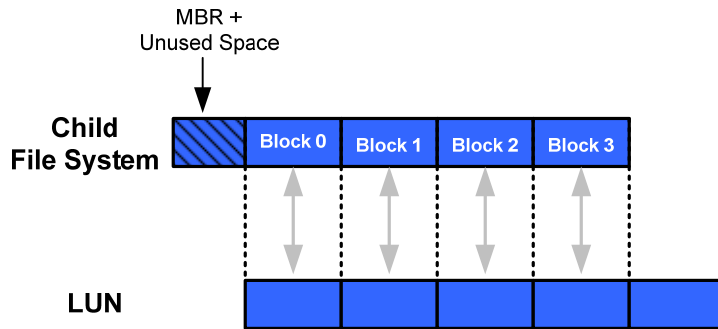


Figure 8) Child file system aligned with the storage array blocks.

LUNS DIRECTLY MAPPED TO THE GUEST OS

For LUNs directly mapped to the child OS using the iSCSI software initiator, the LUN type of the intended child OS should be selected when creating the LUN. This will enable the file system on the LUN to align with NetApp storage array blocks as shown in Figure 8 above.

4.3 CITRIX XENSERVER

NETAPP MANAGED LUNS

For LUNs provisioned using the NetApp SR, only the alignment of the guest VM file system within the VDI to the NetApp storage array is required. This can be enabled using the procedure described in section 4.4.

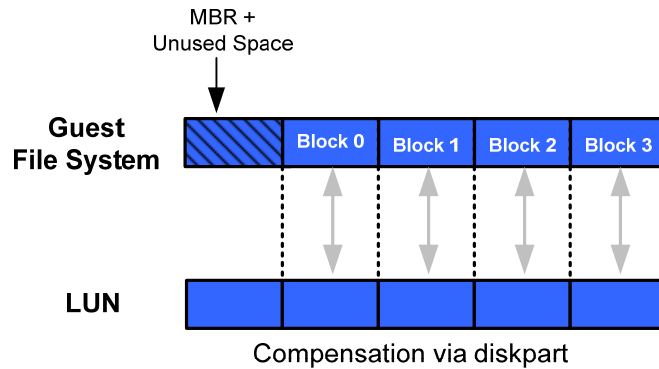


Figure 9) Guest VM file system aligned with the storage array blocks.

NFS EXPORT

For the VHD on NFS storage option, NetApp recommends **thick provisioning of the VHD** first using the steps described below.

Step	Action
1.	In XenCenter, click the newly created NFS SR in the Resources pane, and click the Storage tab.
2.	Click the “Add Disk” button and enter details for the size of VDI you want. Make sure the newly created NFS SR is highlighted and click <i>Add</i> .

Add New Disk

Name: Windows XP SP2 10GB

Description:

Size: 10.000 GB

Select a storage repository to create the disk on

- alignednfs 426.1 GB free of 500.0 GB
- Infra storage 519.1 GB free of 567.7 GB
- KC3070-2_alignednfs01 437.3 GB free of 500.0 GB
- KC3070-2_alignednfs02 437.3 GB free of 500.0 GB
- KC3070-2_alignednfs03 437.3 GB free of 500.0 GB
- KC3070-2_alignednfs04 437.3 GB free of 500.0 GB
- Local storage 32.4 GB free of 32.4 GB
- Local storage 32.4 GB free of 32.4 GB

Add Cancel

The next step would be the alignment of the guest VM file system within the VHD with the NetApp storage array blocks. This can be enabled using the procedure described in section 4.4.

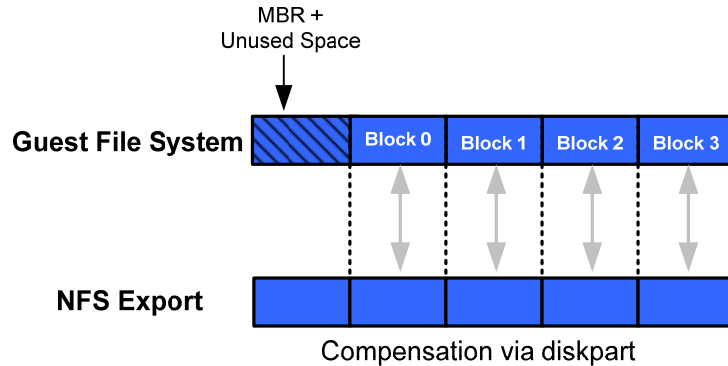


Figure 10) Guest VM file system aligned with the storage array blocks.

Note: If you need to copy the VHD file and enable disk alignment, you need to use the `rfwm` command available in the Master XenServer and then use `sr-scan` to scan the SR to sync with XenCenter.

4.4 GUEST/CHILD VM ALIGNMENT PROCEDURE

There are multiple ways in which misalignment can be prevented when provisioning VMs. Each of these options is described in detail below.


4.4.1 Using diskpart to format with the Correct Starting Partition Offset

This procedure works for Windows VMs hosted on any hypervisor, including VMware ESX (vmdk files hosted on VMFS or NFS datastores), Citrix XenServer (fixed-size vhd files hosted on NFS SR), and Microsoft Hyper-V (fixed-size VHDs). Please note that this procedure is not required for Windows Server 2008 and Vista VMs, which are aligned by default.

ALIGNING BOOT DISK


Virtual disks to be used as boot disk can be formatted with the correct offset at the time of creation by connecting the new virtual disk to a running VM before installing an operating system and manually setting the partition offset. For Windows guest operating systems, one may consider using an existing Windows Preinstall Environment boot CD or alternative tools like Bart's PE CD. To set up the starting offset, follow these steps.

Step	Action
1.	Boot the VM with the WinPE CD.
2.	Select Start > Run and enter Diskpart.
3.	Enter Select Disk0.

4.	Enter Create Partition Primary Align=32. 
5.	Reboot the VM with WinPE CD.
6.	Install the operating system as normal.


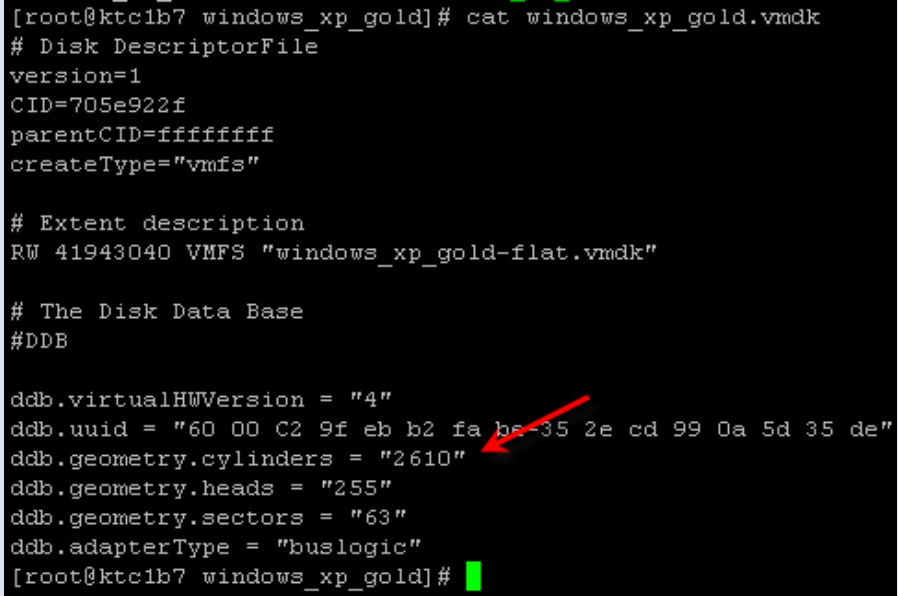
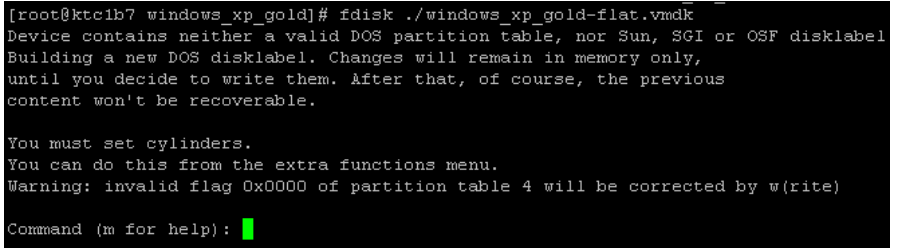
ALIGNING DATA DISK

Virtual disks to be used as data disk can be formatted with the correct offset at the time of creation by using Diskpart in the VM.

Step	Action
1.	Attach the data disk to the VM. Ensure there is no data on the disk.
2.	Select Start > Run and enter Diskpart.
3.	Determine the Disk # for the new data disk and enter Select Disk # (E.g. select disk 1)
4.	Enter Create Partition Primary Align=32. 
5.	Enter Exit to exit out of the Diskpart utility
6.	Format the data disk as you do normally

4.4.2 Using fdisk from ESX service console (Linux and Windows Guest VM)

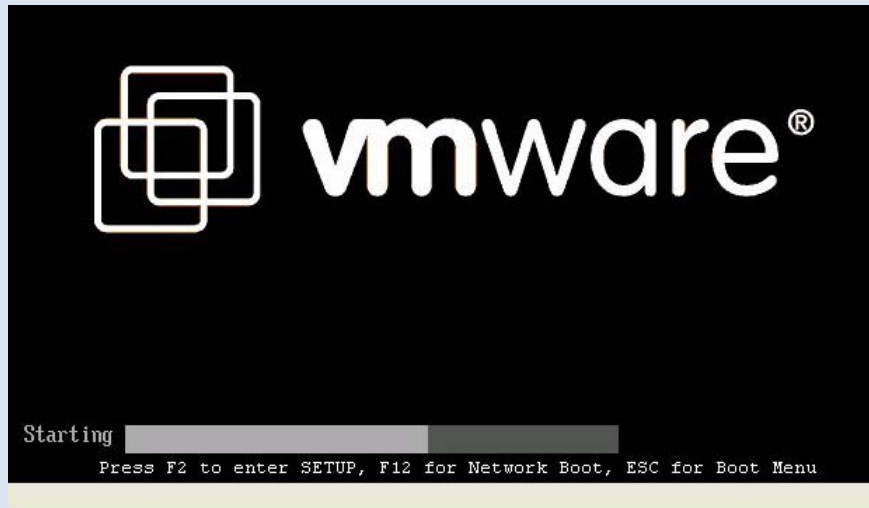
This procedure works for VMware vmdk files hosted on VMFS or NFS datastores, for both Windows and Linux VMs. Please note that this procedure is not required for Windows Server 2008 and Vista VMs, which are aligned by default. To set up the starting offset using the fdisk command in the ESX service console, follow these steps.

Step	Action
1.	Log in to the ESX service console.
2.	<p>CD to the VM directory and view this directory by typing the following commands (shown below):</p> <pre>cd /vmfs/volumes/vdi_gold /windows_xp_gold</pre> <pre>ls -l</pre>  <pre>[root@ktc1b7 root]# cd /vmfs/volumes/VDI_VM_Gold_Datastore/windows_xp_gold [root@ktc1b7 windows_xp_gold]# ls -l total 152 -rw----- 1 root root 21474836480 Aug 22 03:44 windows_xp_gold-flat.vmdk -rw----- 1 root root 382 Aug 22 03:44 windows_xp_gold.vmdk -rw----- 1 root root 0 Aug 22 03:44 windows_xp_gold.vmsd -rwxr-xr-x 1 root root 1181 Aug 22 03:44 windows_xp_gold.vmx -rw----- 1 root root 270 Aug 22 03:44 windows_xp_gold.vmxfs [root@ktc1b7 windows_xp_gold]#</pre>
3.	<p>Get the number of cylinders from the vdisk descriptor by typing the following command (this number will differ depending on several factors involved with the creation of your .vmdk file):</p> <pre>cat windows_xp_gold.vmdk</pre>  <pre>[root@ktc1b7 windows_xp_gold]# cat windows_xp_gold.vmdk # Disk DescriptorFile version=1 CID=705e922f parentCID=ffffffff createType="vmfs" # Extent description RW 41943040 VMFS "windows_xp_gold-flat.vmdk" # The Disk Data Base #DDB ddb.virtualHWVersion = "4" ddb.uuid = "60 00 C2 9f eb b2 fa bc-35 2e cd 99 0a 5d 35 de" ddb.geometry.cylinders = "2610" ddb.geometry.heads = "255" ddb.geometry.sectors = "63" ddb.adapterType = "buslogic" [root@ktc1b7 windows_xp_gold]#</pre>
4.	<p>Run fdisk on the windows_xp_gold-flat.vmdk file by typing the following command:</p> <pre>fdisk ./windows_xp_gold-flat.vmdk</pre>  <pre>[root@ktc1b7 windows_xp_gold]# fdisk ./windows_xp_gold-flat.vmdk Device contains neither a valid DOS partition Table, nor Sun, SGI or OSF disklabel Building a new DOS disklabel. Changes will remain in memory only, until you decide to write them. After that, of course, the previous content won't be recoverable. You must set cylinders. You can do this from the extra functions menu. Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite) Command (m for help):</pre>
5.	<p>You will have to set the number of cylinders.</p> <p>Type in x and then press Enter.</p>
6.	<p>Enter c and press Enter.</p>

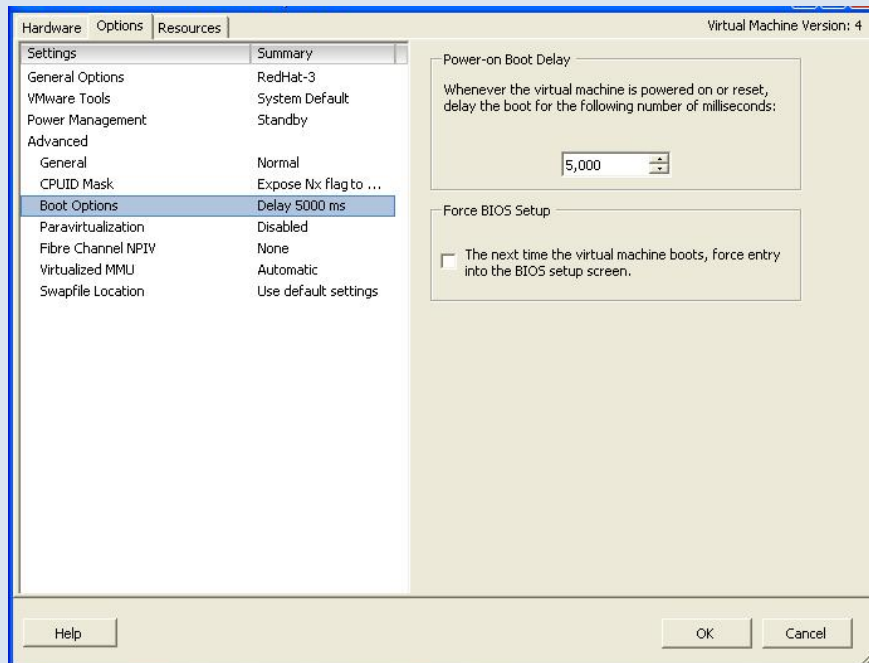
7.	<p>Type in the number of cylinders that you found from step 3.</p> <pre> Command (m for help): x Expert command (m for help): c Number of cylinders (1-1048576): 2610 The number of cylinders for this disk is set to 2610. There is nothing wrong with that, but this is larger than 1024, and could in certain setups cause problems with: 1) software that runs at boot time (e.g., old versions of LILO) 2) booting and partitioning software from other OSs (e.g., DOS FDISK, OS/2 FDISK) Expert command (m for help): █ </pre>
8.	<p>Type p at the expert command screen to look at the partition table (which should be blank).</p> <pre> Expert command (m for help): p Disk ./windows_xp_gold-flat.vmdk: 255 heads, 63 sectors, 2610 cylinders Nr AF Hd Sec Cyl Hd Sec Cyl Start Size ID 1 00 0 0 0 0 0 0 0 0 00 2 00 0 0 0 0 0 0 0 0 00 3 00 0 0 0 0 0 0 0 0 00 4 00 0 0 0 0 0 0 0 0 00 Expert command (m for help): █ </pre>
9.	<p>Return to regular (non-extended) command mode by typing r at the prompt.</p> <pre> Command (m for help): n Command action e extended p primary partition (1-4) p Partition number (1-4): 1 First cylinder (1-2610, default 1): 1 Last cylinder or +size or +sizeM or +sizeK (1-2610, default 2610): Using default value 2610 Command (m for help): █ </pre>
10.	<p>Create a new partition by typing n and then p when asked for the partition type.</p>
11.	<p>Enter 1 for the partition number, 1 for the first cylinder, and press Enter for the last cylinder question to make it use the default value.</p>
12.	<p>Go into extended mode to set the starting offset by typing x.</p>
13.	<p>Set the starting offset by typing b and pressing Enter, selecting 1 for the partition and pressing Enter, and entering 64 and pressing Enter. Please note that '64' is used as an example. You can choose any value as long as it is divisible by 4KB.</p>
14.	<p>Check the partition table by typing p.</p> <pre> Expert command (m for help): p Disk ./windows_xp_gold-flat.vmdk: 255 heads, 63 sectors, 2610 cylinders Nr AF Hd Sec Cyl Hd Sec Cyl Start Size ID 1 00 1 1 0 254 63 1023 64 41929586 83 2 00 0 0 0 0 0 0 0 0 00 3 00 0 0 0 0 0 0 0 0 00 4 00 0 0 0 0 0 0 0 0 00 </pre>

15.	Type r to return to the regular menu.
16.	To set the system type to HPFS/NTF type t.
17.	<p>For the hexcode type 7.</p> <pre data-bbox="443 348 1330 525"> Command (m for help): t Selected partition 1 Hex code (type L to list codes): 7 Changed system type of partition 1 to 7 (HPFS/NTFS) </pre>
18.	<p>Save and write the partition by typing w. Ignore the warning, as this is normal.</p> <pre data-bbox="443 585 1305 779"> Command (m for help): w The partition table has been altered! Calling ioctl() to re-read partition table. WARNING: Re-reading the partition table failed with error 25: Inappropriate ioctl The kernel still uses the old table. The new table will be used at the next reboot. Syncing disks. </pre>

19. Start the VM and run Windows setup. Make sure to press Esc to bring up the boot menu and select “CD ROM drive” to boot from the CD.



If you miss the boot menu, the VM may appear to hang with a black screen with only a blinking cursor. Press ctrl-alt-insert to reboot the VM and try again to catch the boot menu by pressing Escape. If you have trouble catching the boot process above, you can insert a boot delay in the VM settings. In the VI Client, right-click the VM, then → Edit Settings → Options → Advanced / Boot Options



Note that boot delay is in milliseconds. You should return the boot delay to 0 after the VM boots from its virtual disk normally.

20. When the install gets to the partition screen, install on the existing partition. DO NOT DESTROY or RECREATE! C: should already be highlighted. Press Enter at this stage.

5 FIXING FILE SYSTEM ALIGNMENT

5.1 DETECTION

There are multiple ways in which alignment issues can be detected. Each of these options is described in detail below.

USING NETAPP MBRSCAN TOOL

The NetApp mbrscan tool interrogates a VM disk file and reports on the file system alignment. It can effectively check for alignment on VMware vmdk and thick type vhd files for Citrix XenServer that are partitioned using MBR.

- For VMware VMFS and NFS datastores this tool can be run from the ESX console. For NFS datastores, it can also be run from any UNIX or Linux host that has permissions to mount the NFS datastore.
- For Citrix XenServer NFS SR, this tool can be run from domain 0 (dom0).

The mbrscan tool is available as part of VMware ESX Host Utilities 5.0 or higher, which can be downloaded from the [NetApp NOW site](#). For the detailed procedure on how to use the mbrscan tool, please refer to the Installation and Setup Guide for VMware ESX Host Utilities 5.0 or higher available on the NetApp NOW site.

USING MSINFO32 (WINDOWS GUEST VM ONLY)

This procedure works for all of the hypervisors (including VMware ESX, Citrix XenServer, and Microsoft Hyper-V). When aligning the file systems of virtual disks for use with NetApp storage systems, the starting partition offset must be divisible by 4,096. For Windows guest operating systems, verifying this value is easy. Run msinfo32 on the guest VM by selecting Start > All Programs > Accessories > System Tools > System Information. Next, navigate to Components > Storage > Disks and check the value for Partition Starting Offset. For misaligned VMs, you will typically find that the VM is running with a default starting offset value of 32,256, which is not completely divisible by 4,096 and hence the partition is not aligned as highlighted in Figure 11 below.

Note: For Windows LUN type RDM or raw LUNs directly mapped to the VM, 32,256 is reported as the correct starting offset value. This is true because the storage controller compensated for the offset when selecting the correct LUN type.

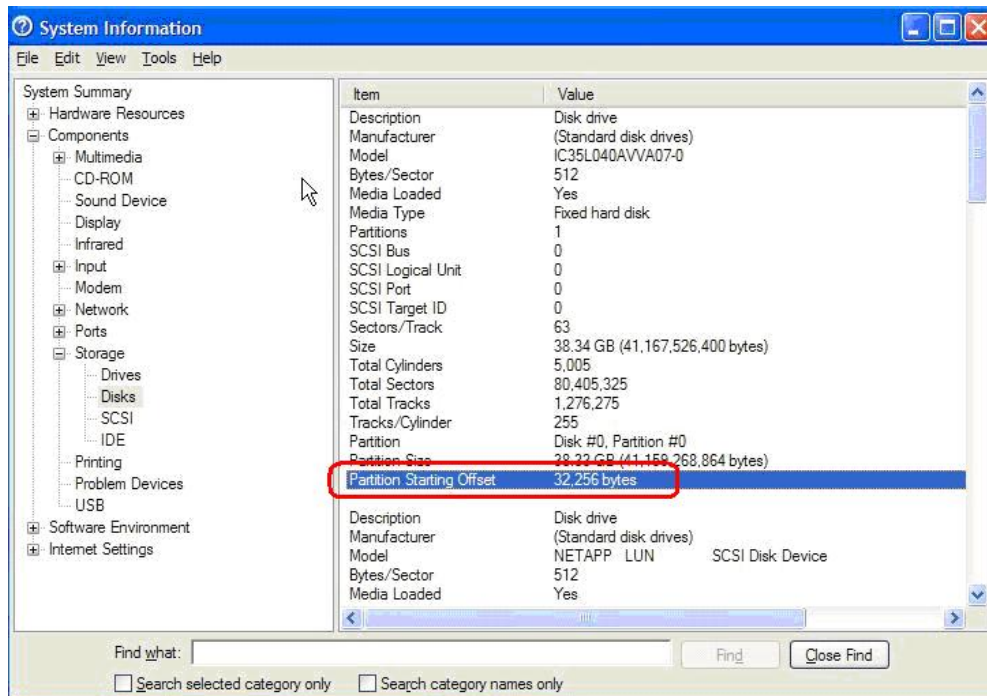


Figure 11) Using system information to identify the starting partition offset.

USING FDISK FROM THE ESX SERVICE CONSOLE (WINDOWS AND LINUX GUEST VM)

For VMware deployments you can also check the LUN alignment using the fdisk tool from the ESX service console.

Enter the following command at the service console command prompt.

```
fdisk -lu /vmfs/volumes/<datastore>/<vm>/<vm>-flat.vmdk
```

This will report the required information, including starting offsets, and then exit. If the Start value is 63, the partition is not aligned. This procedure works for VMware ESX only. For more information, see the following VMware article: http://www.vmware.com/pdf/esx3_partition_align.pdf.

I/O ALIGNMENT PERFORMANCE STATISTICS

Data ONTAP contains performance counters that track I/O alignment for specific LUNs. These counters are accessible using the Data ONTAP command line.

- read/write_align_histo.n displays an 8-bin histogram that tracks the percentage of I/Os that begin at a given 512-byte logical block offset from the beginning of a 4K storage block.
- read/write_partial_blocks.XX displays the percentage of read/writes that are smaller than 4K.

To view the counters, access the Data ONTAP command line through SSH, RSH, or Telnet and run the following command:

```
NetApp_Controller> priv set diag; stats show lun; priv set admin
```

Output:

```
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.0:100%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.1:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.2:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.3:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.4:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.5:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.6:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.7:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.0:100%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.1:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.2:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.3:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.4:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.5:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.6:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.7:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_partial_blocks:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_partial_blocks:0%
```

The example above shows that 100% of reads and writes to the LUN begin on a storage block boundary as indicated by the .0 bucket of each histogram.

The following output shows unaligned I/O to the same LUN over a different time interval.

```
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.0:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.1:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.2:100%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.3:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.4:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.5:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.6:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_align_histo.7:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.0:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.1:0%
```

```

lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.2:100%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.3:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.4:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.5:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.6:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_align_histo.7:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:read_partial_blocks:0%
lun:/vol/luns/vmdks/esx_1g_1m-C4dAFJL-c/Og:write_partial_blocks:0%

```

In this example, 100% of read and write I/Os begin in the .2 bucket of each histogram as indicated. The .2 bucket indicates that the I/O begins two 512-byte logical blocks into a 4K storage block.

5.2 CORRECTION

Correcting the starting offset is best addressed by correcting the template from which new VMs are provisioned. Aligning the boot disk in existing virtual machines is not required but will improve performance under some circumstances. If you have diagnosed performance issue due to misalignment, it may be necessary to align boot disks as well as data disks. In any case, it is essential to align templates and "gold image" VMs in order to prevent proliferation of misaligned VMs, and to align any new VMs using procedures in this document.

The following are examples of situations where misaligned boot disks can become an issue:

- The boot disk is also the data disk as is the case for VMs that only have one virtual disk.
- Oversubscription of memory within a VM triggering swapping or paging within the VM. This can be alleviated by aligning the virtual disk that contains the page file or swap partition (this is often the boot disk), increasing the memory allocated to the VM or both.
- Use of the balloon driver for memory management when physical memory is oversubscribed can trigger VMs to swap or page.

Note that ESX swapping a VM into the .vswp file is not affected by alignment of the VM boot disk, but is affected by alignment of VMFS.

CORRECTION USING FDISK OR DISKPART

Use either of the procedure(s) described above in section 4.4 to create a new aligned virtual disk. Attach this new aligned virtual disk to the VM and copy the contents from the existing misaligned virtual disk to the new disk. Finally, detach and destroy the misaligned virtual disk after verifying the contents and integrity of the data on the new aligned virtual disk. If the misaligned virtual disk is the boot partition, follow these steps.

Step	Action
1.	Back up the VM system image.
2.	Shut down the VM.
3.	Attach the misaligned system image virtual disk to a different VM.
4.	Attach a new aligned virtual disk to this VM.
5.	Copy the contents of the system image (e.g., C: in Windows) virtual disk to the new aligned virtual disk. There are various tools that can be used to copy the contents from the misaligned virtual disk to the new aligned virtual disk: <ul style="list-style-type: none"> • Windows xcopy • Norton/Symantec™ Ghost: Norton/Symantec ghost can be used to back up a full system image on the misaligned virtual disk and then be restored to a precreated, aligned virtual disk file system.

For VMware RDM and LUNs directly mapped to the guest OS, create a new LUN using the correct LUN type. Next, map the LUN to the VM and copy the contents from the misaligned LUN to this new LUN.

For Microsoft Hyper-V NTFS-formatted LUNs mapped to the Hyper-V parent partition using incorrect LUN type but with aligned VHDs, create a new LUN using the correct LUN type and copy the contents from the

misaligned LUN (VHDs) to this new LUN. However, if the VHDs are also misaligned in addition to the NTFS-formatted LUNs mapped to the Hyper-V parent partition, first create a new LUN using the correct LUN type and copy the contents from the misaligned LUN (VHDs) to this new LUN. Next, perform the steps described in section 4.4 to create new aligned VHDs on the new LUN and copy the contents from the existing VHDs to the new VHD. If the VHD is a boot partition, follow the steps described earlier in this section. For pass-through disks and LUNs directly mapped to the child OS, create a new LUN using the correct LUN type, map the LUN to the VM, and copy the contents from the misaligned LUN to this new LUN.

For Citrix XenServer LUNs created using NetApp Data ONTAP Adapter, create a new LUN and map the LUN to the VM and copy the contents from the misaligned LUN to this new LUN.

CORRECTION USING MBRALIGN

The NetApp mbralign tool properly aligns a guest file system to the NetApp storage array blocks. This tool works for VMware vmdk files hosted on VMFS or NFS datastores and is run on the ESX service console. The high-level steps to correct the misalignment are:

Step	Action
1.	Shut down the VM.
2.	Run the mbralign tool on each vmdk.
3.	Start up the VM.
4.	Once the VM starts correctly and is verified intact, delete the backup VMDK created by mbralign.

The mbralign tool along with detailed instructions is available for download [here](#) on the NetApp [Tool Chest](#) on the [NetApp NOW site](#).

FIXING GRUB FOR LINUX VMS

After aligning a Linux VM using mbralign, GRUB will usually no longer be able to find the stage 1.5 boot loader, and the VM will stop with a black screen with the word GRUB as in Figure 12 below.

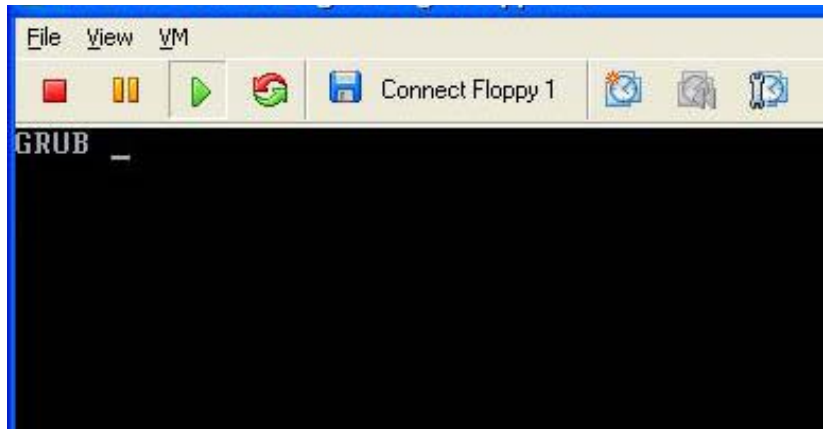
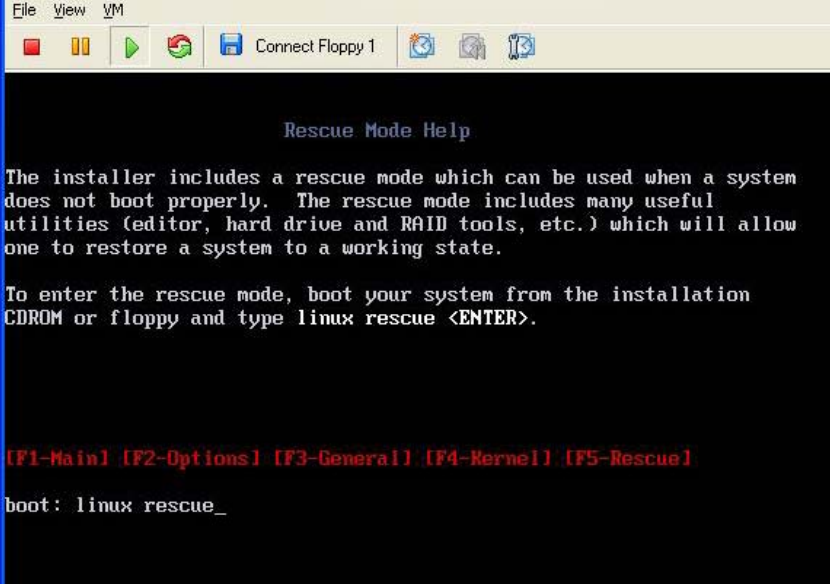


Figure 12) A VM hung at GRUB

To fix this issue, please follow these steps.

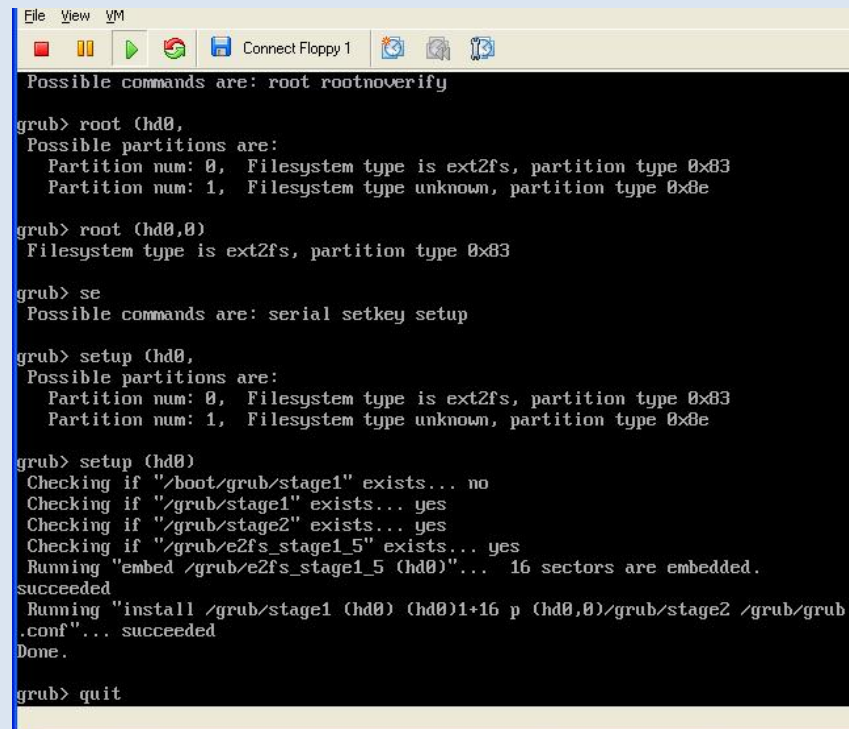
Step	Action
1.	Boot the VM from a Linux Rescue CD/ISO (Disk 1 of your Linux distribution will usually work).
2.	At the GRUB hang display in the VM remote console, click in the display area to make sure it is active.
3.	Hit ctrl-alt-insert to reboot the VM.

4.	As soon as you see the VMware BIOS splash hit escape once. You should have a boot menu. If you have trouble catching the boot menu, you can follow the procedure at the end of Section 4.4.2 to add a boot delay.
5.	Select CD-ROM.
6.	<p>At the Linux boot screen, enter the following command to boot Linux in rescue mode</p> <pre>:linux rescue</pre> 
7.	Take the defaults for Anaconda (the blue/red configuration screens). Networking is optional.
8.	Launch GRUB by typing the following <pre># grub</pre>

9. Select the correct disk and fix the GRUB stages as follows

```
grub> root hd(0,0)
grub> setup (hd0)
grub> quit
ctrl-d
```

Note that GRUB provides a form of command completion by pressing tab, which will also list hard disks and partitions where appropriate. This effect is illustrated in the screenshot below. Also note that if you have multiple disks, hd0 is probably your boot disk, but you must know how Linux and its components were installed.



The screenshot shows a GRUB terminal window with a menu bar at the top containing icons for File, View, VM, and a 'Connect Floppy 1' button. The terminal text is as follows:

```
Possible commands are: root rootnoverify
grub> root (hd0,
Possible partitions are:
  Partition num: 0, Filesystem type is ext2fs, partition type 0x83
  Partition num: 1, Filesystem type unknown, partition type 0x8e
grub> root (hd0,0)
Filesystem type is ext2fs, partition type 0x83
grub> se
Possible commands are: serial setkey setup
grub> setup (hd0,
Possible partitions are:
  Partition num: 0, Filesystem type is ext2fs, partition type 0x83
  Partition num: 1, Filesystem type unknown, partition type 0x8e
grub> setup (hd0)
Checking if "/boot/grub/stage1" exists... no
Checking if "/grub/stage1" exists... yes
Checking if "/grub/stage2" exists... yes
Checking if "/grub/e2fs_stage1_5" exists... yes
Running "embed /grub/e2fs_stage1_5 (hd0)"... 16 sectors are embedded.
succeeded
Running "install /grub/stage1 (hd0)1+16 p (hd0,0)/grub/stage2 /grub/grub
.conf"... succeeded
Done.
grub> quit
```

10. After logging out (ctrl-d) Linux rescue will shut down and reboot. Linux should come right up.

CORRECTION VERIFICATION

The alignment correction can be verified by following the procedure to collect the Data ONTAP performance counters on LUNs, as described earlier in the detection section 5.1.

6 CONCLUSION

File system misalignment is a known issue in virtual environments and can cause performance issues for VMs. The steps outlined in this document should be taken so that the issue can be prevented during the storage provisioning and VM creation phase of the project. For existing VMs, the steps outlined in this document should be taken to detect and correct any existing misalignment issues. Please note that this issue is not unique to NetApp storage and can occur with any storage array. For more details please refer to the VMware article [VMware Infrastructure 3 Recommendations for Aligning VMFS Partitions](#).

7 FEEDBACK

Please send an e-mail to xdl-vgibutmevmtr@netapp.com with questions or comments concerning this document.

8 REFERENCES

[VMware Infrastructure 3 Recommendations for Aligning VMFS Partitions](#)

[VMware ESX Server Performance Tuning Best Practices for ESX Server 3](#)

[Update Disk Partition Tool for Windows 2003](#)

[Partition Design](#)

[Disk Performance May Be Slower Than Expected When You Use Multiple Disks in Windows Server 2003, in Windows XP, and in Windows 2000](#)

[How to Align Exchange I/O with Storage Track Boundaries](#)

[Pre-deployment I/O Best Practices: SQL Server Best Practices article](#)

9 VERSION HISTORY

Version 1.0	March 2009	Original Document
-------------	------------	-------------------

10 APPENDIX

DISK GEOMETRY AND CHS (CYLINDER/HEAD/SECTOR)

Disks use geometry to identify themselves and their characteristics to the upper-layer operating system. The upper-layer operating system uses the disk geometry information to calculate the size of the disk and partitions the disk into predetermined addressable blocks. Just as with physical disks, logical disks (LUNs) report disk geometry to the host (physical host, virtualization host, or the VM, depending on the mode of usage) so that it can calculate space and partition the LUN into addressable blocks.

By default, many guest operating systems, including most versions of Windows, attempt to align the first sector on a full track boundary. The installer/setup routine requests the CHS information that describes the disk from the BIOS (PC firmware that manages disk I/O at a low level), or, in the case of many virtual machines, an emulated BIOS. The issue is that the CHS data hasn't actually corresponded to anything physical since the late 1980s (even in physical machines). At larger LUN sizes (usually 8GB or more), the S number (sectors per track) is always reported as 63, so the partitioning routine sets a starting offset of 63 sectors in an attempt to start the partition on a track boundary. While this may be correct for a single physical disk drive, it does not line up with any storage vendor's logical block size. Physical disk blocks always have 512 (usable/visible) bytes, but, for efficiency and scalability reasons, storage devices use a logical block size that is some number of physical blocks, usually a power of 2. For NetApp, the logical block size is 4K, that is, 8 disk blocks. For background information on Cylinder-Head-Sector concepts, please see <http://en.wikipedia.org/wiki/Cylinder-head-sector>.



www.netapp.com

© 2009 NetApp. All rights reserved. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FilerView, FlexClone, NOW, and SnapDrive are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Microsoft and Windows are registered trademarks and Hyper-V is a trademark of Microsoft Corporation. VMware is a registered trademark of VMware, Inc. Linux is a registered trademark of Linus Torvalds. UNIX is a registered trademark of The Open Group. Symantec is a trademark of Symantec Corporation. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-3747