



Technical Report

NetApp Thin-Provisioned LUNs on RHEL 6.2 Deployment Guide

Rishikesh Boddu, Martin George, NetApp
July 2012 | TR-4046

TABLE OF CONTENTS

1	Introduction	3
1.1	Intended Audience and Assumptions.....	3
2	Storage Efficiency	3
2.1	Thin Provisioning	3
2.2	Space Reclamation.....	4
2.3	Need for Space Reclamation	4
3	Red Hat Enterprise Linux Operating System	7
3.1	Overview	7
3.2	Discard.....	7
3.3	SCSI UNMAP.....	7
4	Best Practices for Deploying NetApp Thin LUNs with RHEL 6.2 Using Discard	8
4.1	Hardware and Software Prerequisites	8
4.2	Data ONTAP Parameter Prerequisites	9
5	Limitations	9
	Appendix	9
	NetApp Thin-Provisioned LUNs on RHEL 6.2	9
	Steps to Execute at the Storage Side.....	10
	Steps to Execute at the Host Side	10
	References	10

LIST OF TABLES

Table 1)	Recommended settings for thin-provisioned LUNs on RHEL 6.2.....	9
----------	---	---

LIST OF FIGURES

Figure 1)	Traditional provisioning versus thin provisioning.....	3
Figure 2)	Typical host versus storage LUN space utilization.....	4
Figure 3)	Typical host versus storage LUN space utilization: Delete File2.....	5
Figure 4)	Typical host versus storage LUN space utilization: Write File4 using 20% of LUN.....	6
Figure 5)	Host versus storage space utilization using space reclamation.....	8

1 Introduction

Storage efficiency is one of the most important aspects of the IT lifecycle. By improving the efficacy of the storage environment, we can optimize the overall IT efficiency. In the face of exponential data growth, efficient management of data is critical. NetApp® thin provisioning, one of the significant features of optimizing storage efficiency, is used extensively in current NetApp storage environments. In the province of storage efficiency, space reclamation—an important feature of NetApp thin provisioning—is a vital aspect of the full data management lifecycle for NetApp buyers.

This report discusses the importance of space reclamation and deployment of NetApp thin LUNs on Red Hat Enterprise Linux® 6.2 (RHEL 6.2).

1.1 Intended Audience and Assumptions

This document is for storage administrators, system administrators, and data center managers. This document assumes familiarity with the following:

- NetApp FAS systems and Data ONTAP® software
- Linux operating systems

2 Storage Efficiency

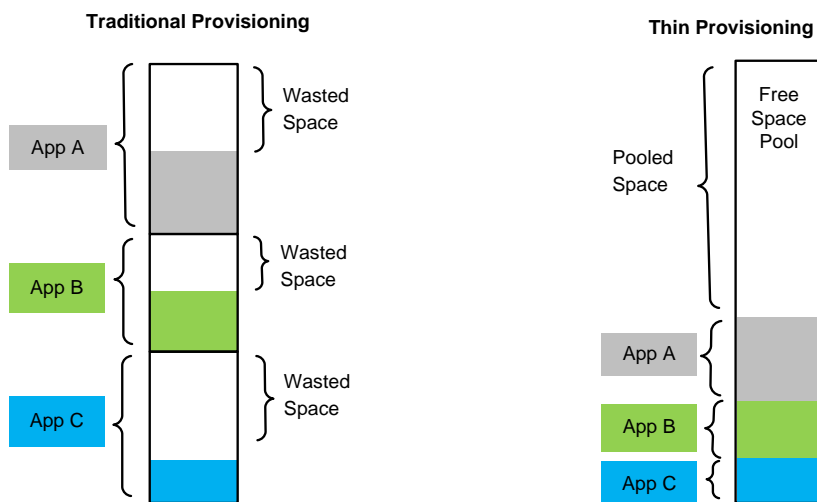
Thin provisioning and space reclamation play a major role in optimizing storage efficiency. Appropriate deployment and management of both these features can significantly boost storage utilization rates. Thin provisioning can be achieved through various combinations of volume guarantees with FlexVol® volumes and LUN space reservation settings. For more information about thin provisioning, refer to [TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide Data ONTAP 8.1 7-Mode.](#)

2.1 Thin Provisioning

Thin provisioning allows the provisioning of more logical storage space to the hosts or servers connecting to the storage system than what is actually available on the storage system. In other words, thin provisioning allocates storage on demand when data comes in, instead of preallocating it.

Figure 1 illustrates the advantages of thin provisioning compared with the traditional provisioning in improving storage utilization.

Figure 1) Traditional provisioning versus thin provisioning.



Traditional provisioning results in poor storage utilization, and increased infrastructure and management costs. Thin provisioning in comparison, improves storage utilization, reduces capital outlays, and increases cost savings.

For additional information on thin-provisioning fundamentals in NetApp storage systems, refer to [TR-3965: NetApp Thin Provisioning Deployment and Implementation Guide Data ONTAP 8.1 7-Mode](#).

2.2 Space Reclamation

NetApp Data ONTAP enables thin provisioning so that customers can size their storage and LUNs independently to optimize space utilization; for example, we could have an appliance with 2TB of disk space holding LUNs of 2.5TB. Thin provisioning helps keep space utilization at near capacity.

Space reclamation assumes that system administrators using LUNs for applications, such as file systems or a database, do not ever fill up their file systems and maintain some percentage of free space within them. LUNs can be underprovisioned on the storage system since multiple thin LUNs contribute to free space and also average out any surges in the space requirements of particular LUNs. So, we could have five LUNs that are 100GB in size (perpetually 30% free) in a 400GB volume/aggregate, leaving space for any temporary spikes in utilization requirements.

2.3 Need for Space Reclamation

Thin provisioning of LUNs is based on the assumption (which is not correct) that an application will never use 100% of the underlying storage. Any deletion of files on a file system does not free up space on the storage system because file systems maintain their own concept of free space using metadata. This can eventually lead to 100% space utilization on the storage system for a LUN even though the file system resident on the LUN has substantial free space. This scenario is not unique to NetApp. From a storage system perspective, the percentage of a LUN used will constantly increase and never decrease unless space reclamation is used; hence, space reclamation is required to optimize space utilization.

Figure 2) Typical host versus storage LUN space utilization.

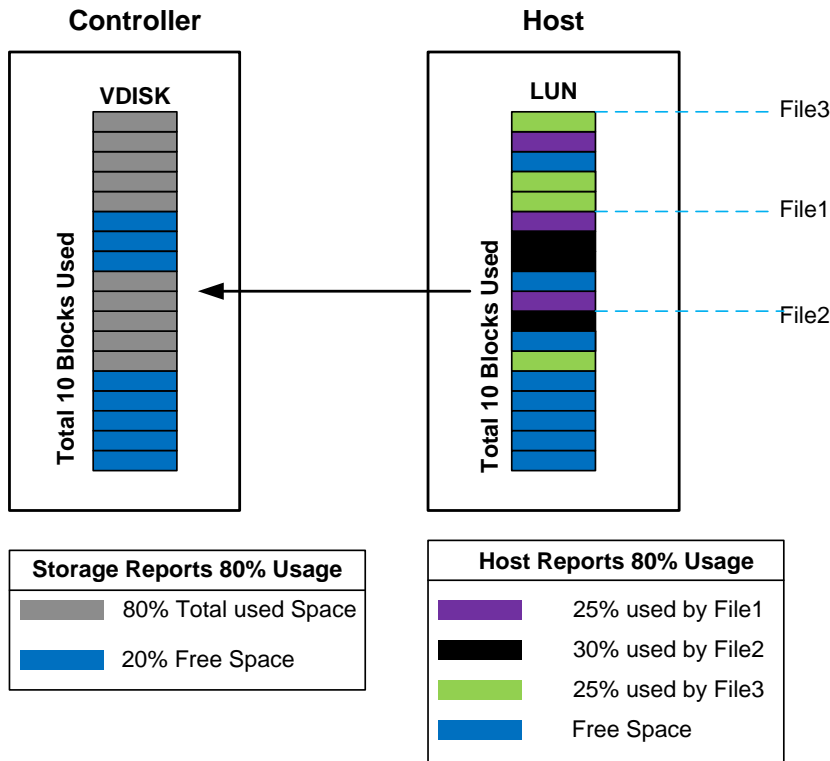


Figure 2 shows a LUN's usage from the host or file system perspective. Here, files are written and deleted from the file system or host. File1, File2, and File3 are being written by the host using almost 80% of the LUN. As predicted, the file system or host and the storage system both report the same percentage of used space, which can be seen by triggering the `df` command on both the host and storage systems.

In Figure 3, File2 is deleted by the file system, which means that the blocks being used by File2 are free and available to the file system. Since there is no technique through which the host can inform the storage about the deletion of File2, the storage will still hold the blocks used for File2. This is where the change occurs between the host and storage space utilization. Due to this, storage still reports 80% usage whereas the host reports 50% usage.

Figure 3) Typical host versus storage LUN space utilization: Delete File2.

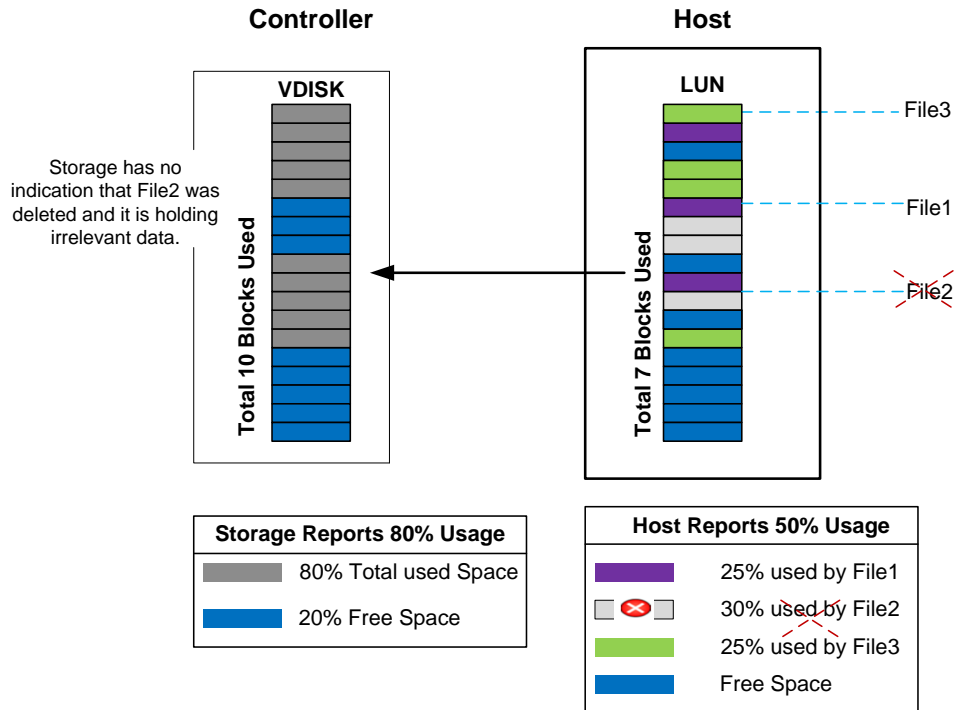


Figure 4) Typical host versus storage LUN space utilization: Write File4 using 20% of LUN.

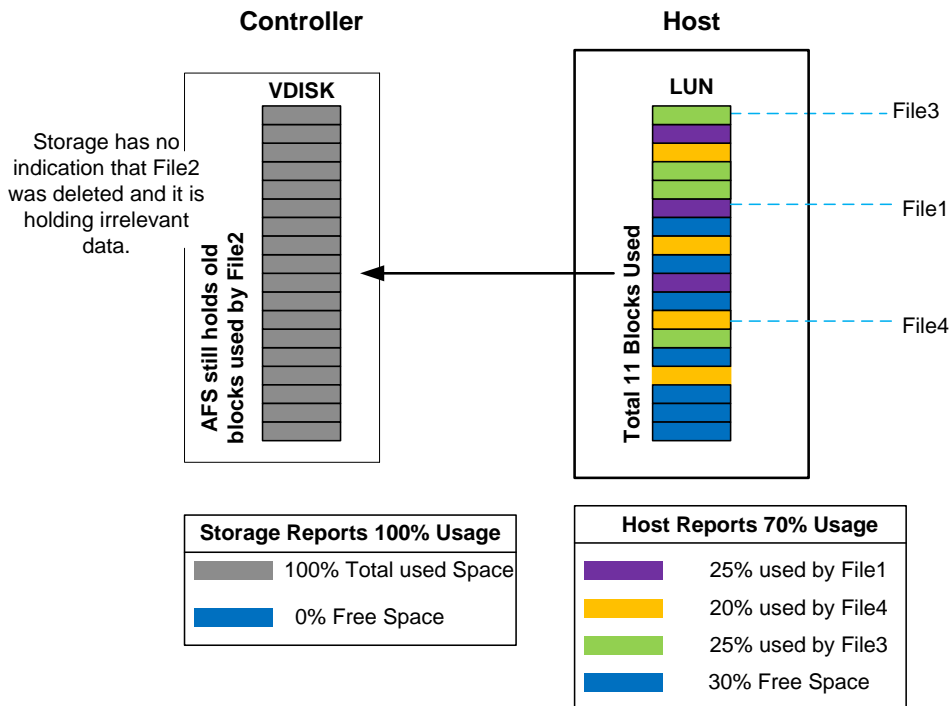


Figure 4 shows that File4 has been written by the file system using 20% of the LUN space. However, the storage shows an increase in space utilization of 20% and reports it as 100% full. This is because, as shown in Figure 3, the storage still holds the blocks used by File2, which was actually deleted by the host. This means that the storage reports 100% space usage whereas file system sees an approximate 30% free space to be used.

Even if a user writes one more file, File5, using 20% of the LUN space, space utilization from the storage perspective still shows 100% full. Whether a write or a delete action is performed now by a file system, you will observe that the host space utilization reflects the user deletion and creation of the file, whereas the storage will report as 100% full because it is unaware of the host's deletion activity.

Applications typically use and free blocks within a LUN, but do not propagate this information back to the storage on a per-block basis since the blocks that are freed are not written directly; instead, the metadata tracking block usage is updated. Unwanted obsolete data that is not required by the application still resides in the LUN on the storage system and consumes blocks. If a Snapshot™ copy is created in the storage volume that contains the LUN, these blocks end up being used in Snapshot copies as well, worsening space utilization by forcing the storage system to store unwanted application data for a longer period of time. The space utilization of the LUN on the storage exceeds the space utilization perceived by the application or file system.

3 Red Hat Enterprise Linux Operating System

3.1 Overview

The RHEL 6.2 operating system supports space reclamation in settings with NetApp thin provisioning support. As mentioned in the preceding section, space reclamation is a vital aspect of the full data management lifecycle for NetApp customers. This standards-based capability resonates with NetApp's storage efficiency strategy by enabling customers to reclaim space on the storage array that has already been freed by the host operating system.

3.2 Discard

Red Hat has implemented space reclamation on RHEL 6.2 through the ext4 "discard" feature. Discard is the mechanism through which the host operating system informs the storage array that a given range of blocks is no longer in use. Discard requests are issued to the storage array using the negotiated `discard` command specific to the storage protocol (TRIM for ATA, `WRITE_SAME` with `UNMAP` set, or the `UNMAP` command for SCSI).

At present, ext4 is the only fully supported file system that supports discard. To enable `discard` commands on a device, use the mount option `discard`.

For example, to mount the Device-Mapper Multipath (`dm-multipath`) device using the `discard` option, run:

```
mount -t ext4 -o discard /dev/mapper/360a98000572d4461524a64793868696d /mnt
```

Alternately, you can enable discard without using the automatic "`-o discard`" mount option, by using the `fstrim` command which would manually trigger space reclaim. For this, first mount the `dm-multipath` device using ext4:

```
mount -t ext4 /dev/mapper/360a98000572d4461524a64793868696d /mnt
```

After writing to this device and then removing the data, you can manually trigger space reclaim on this device by running the `fstrim` command as shown below:

```
fstrim /mnt
```

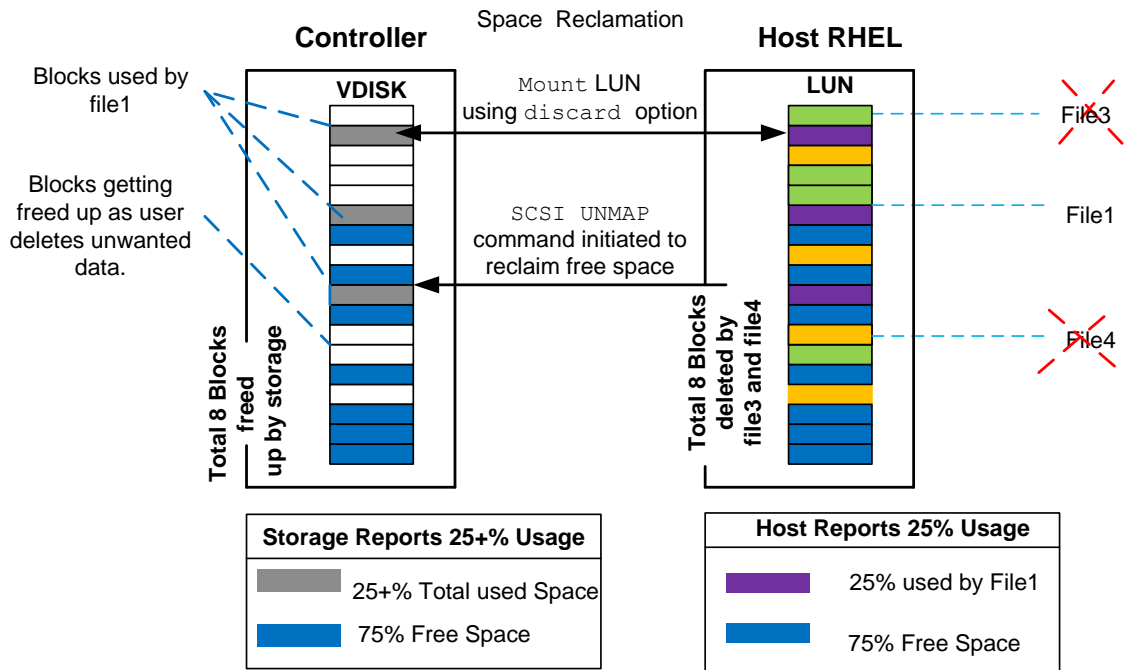
3.3 SCSI UNMAP

RHEL 6.2 supports the discard feature using the `UNMAP` and `WRITESAME` commands; by default, `UNMAP` is used. For RHEL 6.2 NetApp thin-provisioned LUNs connected to NetApp arrays running a minimum of Data ONTAP 8.1 operating in 7-Mode, `discard` uses the default command `UNMAP` for space reclamation. The `SCSI UNMAP` command enables the host to "reclaim space"; that is, to reclaim blocks that are no longer used in a file system.

Example:

After deleting a file, a `SCSI UNMAP` command is sent to the LUN to reclaim the space used by the file in the storage system. This process is automatic and is not controllable.

Figure 5) Host versus storage space utilization using space reclamation.



4 Best Practices for Deploying NetApp Thin LUNs with RHEL 6.2 Using Discard

4.1 Hardware and Software Prerequisites

Verify that the following configuration is set on both the host and controller end to facilitate complete configuration.

Host Requirement

- Operating system RHEL 6.2 or later.
- NetApp recommends installing the NetApp Host Utilities Kit—Linux HU 5.3 Kit to complete the setup. Refer to the "NetApp Host Utilities Installation and Setup Guide" at <http://now.netapp.com/NOW/cgi-bin/software>.
- Before installing the Linux Host Utilities Kit, verify that the following 32-bit packages are installed on it for proper sanlun functioning.
 - glibc
 - nss-softokn-freebl
 - libgcc
 - device-mapper-libs
 - libsepol
 - libselinux
 - libudev

For details on the host, refer to the [NetApp Interoperability Matrix](#).

NetApp Storage Controller Requirement

- Hardware: All series
- Minimum: Data ONTAP 8.1 operating in 7-Mode or later
- Supported protocols: FCP, iSCSI, and FCOE

4.2 Data ONTAP Parameter Prerequisites

The volume settings that are primarily recommended to facilitate appropriate configurations are as follows.

Best Practices

Best practices to configure volumes and LUNs:

- Enable autosize on the volume to grow volume as needed.
- Enable provisioning thresholds by specifying `space_alloc` during LUN creation.

Table 1) Recommended settings for thin-provisioned LUNs on RHEL 6.2.

Parameter	Value
Volume Options	
<code>autosize</code>	<code>on</code>
LUN Options	
<code>space reservation</code>	<code>disabled</code>
<code>space_alloc</code>	<code>on</code>

5 Limitations

The limitations are as follows:

- The current RHEL 6 storage stack does not distinguish between driver, device, and transport errors; all errors are treated as input/output errors. In addition, provisioning threshold errors are treated as I/O errors on the RHEL 6.0 and 6.1 host. This limitation has been fixed in RHEL 6.2, wherein all of these types of errors are properly propagated up through the storage stack to the upper layers.
- In RHEL 6.0, `ext4` “discard” is executed every time blocks are freed, which results in performance degradation. Starting from RHEL 6.1, `ext4` implements batched discard—that is, batching of the various discards—to improve performance.

Appendix

NetApp Thin-Provisioned LUNs on RHEL 6.2

With Data ONTAP 8.1 operating in 7-Mode, NetApp supports the auto space reclamation on the RHEL 6.2 `ext4` file system. The following example shows how to use space reclamation using the “discard” mechanism for a NetApp thin-provisioned LUN on RHEL 6.2.

In the example, we are creating a 12GB volume and a 10GB thin LUN for the specified volume. Autosize the volume to grow a maximum of 15GB in increments of 50MB and enable provisioning thresholds by using `space_alloc` to trigger the threshold limit when the LUN is nearly full.

Steps to Execute at the Storage Side

1. Create the test_vol volume of size 12GB on aggregate "aggr1."

```
vol create test_vol -s volume aggr1 12g
```

2. Enable autosize on test_vol to grow to a maximum GB specified by the user; here we set to 15GB in 50MB increments.

```
Vol autosize test_vol -i 50m -m 15g on
```

3. Create a thin-provisioned LUN with space reservation disabled. Enable provisioning thresholds to the LUN by using the space_alloc feature.

```
Lun create -s 10g -t linux -o noreserve -e space_alloc /vol/test_vol/thin_lun
```

Steps to Execute at the Host Side

1. Verify that the Linux Host Utilities Kit 5.3 has been installed.
2. Map the LUN and configure dm-multipath.
3. Enable space reclaim on the dm-multipath device by the following methods:
 - **Using automatic discard.**

This can be done by using ext4's `-o discard` mount option as follows:

- a. Mount the dm-multipath device with the `-o discard` option:

```
mount -t ext4 -o discard /dev/mapper/360a98000572d4461524a64793868696d /mnt
```

- b. Write to this multipath device and then delete the same data. This will automatically trigger discard on the device. You can verify this on the target by running the `df` command that will display the LUN size as first increasing (during the write) and then decreasing (during the delete), which indicates that space reclaim is active.

- **Using manual discard.**

This can be done by running the `fstrim` command as follows:

- a. Mount the dm-multipath device:

```
mount -t ext4 /dev/mapper/360a98000572d4461524a64793868696d /mnt
```

- b. Write to this multipath device and then delete the same data. To manually trigger discard on the device, run the `fstrim` command:

```
fstrim /mnt
```

- c. You can run the `df` command on the target to verify that space reclaim is active; the LUN size will first increase (during the write), and then decrease (when `fstrim` is run).

References

- SCSI Block Commands-3 (SBC-3)
<http://www.t10.org/>
- Red Hat Enterprise Linux 6 Storage Administration and Deployment Guide
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/
- TR-3975: NetApp Thin Provisioning Deployment and Implementation Guide Data ONTAP 8.1 7-Mode
<http://media.netapp.com/documents/tr-3965.pdf>
- RHEL 6 DM-Multipath Configuration and Administration Guide
http://docs.redhat.com/docs/en-US/Red_Hat_Enterprise_Linux/6/html/DM_Multipath/index.html

Refer to the [Interoperability Matrix Tool](#) (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NetApp provides no representations or warranties regarding the accuracy, reliability, or serviceability of any information or recommendations provided in this publication, or with respect to any results that may be obtained by the use of the information or observance of any recommendations provided herein. The information in this document is distributed AS IS, and the use of this information or the implementation of any recommendations or techniques herein is a customer's responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. This document and the information contained herein may be used solely in connection with the NetApp products discussed in this document.

[Go further, faster®](#)



www.netapp.com

© 2012 NetApp, Inc. All rights reserved. No portions of this document may be reproduced without prior written consent of NetApp, Inc. Specifications are subject to change without notice. NetApp, the NetApp logo, Go further, faster, Data ONTAP, FlexCache, FlexClone, FlexVol, SnapMirror, and Snapshot are trademarks or registered trademarks of NetApp, Inc. in the United States and/or other countries. Linux is a registered trademark of Linus Torvalds. All other brands or products are trademarks or registered trademarks of their respective holders and should be treated as such. TR-4046-0312