



Dell Storage SC Series Arrays with SUSE Linux Enterprise Server 12

Randolph Nethers, Linux/UNIX Product Specialist
February 2016

Revisions

Date	Revision	Description
August 2011	1.0	Release based on SLES 11
February 2016	2.0	Release based on SLES 12

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.
Copyright © 2011–2016 Dell Inc. All rights reserved. Dell and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective -companies.



Table of contents

1	SLES 12 and SC Series overview	6
1.1	New to SLES 12	6
1.2	Patch and service patch levels	6
1.3	Special filesystems	6
2	Fibre Channel	7
2.1	Boot from SAN	7
2.2	Installing SLES 12 to a multipath volume	7
2.3	Enabling multipath on an existing installation	10
2.4	Adding a volume	10
2.5	Expanding a volume	11
2.6	Fibre Channel HBA optimization	12
2.6.1	Optimal HBA settings	13
2.6.2	Port connectivity timeout	13
2.6.3	Queue depth	14
2.6.4	Extended logging	14
3	iSCSI	15
3.1	Host configuration	15
3.2	Adding a volume	18
3.3	Expanding a volume	19
3.4	iSCSI timeout values	19
3.5	10 GB Ethernet and iSCSI	21
4	The Linux Device Mapper	22
4.1	SC Series device definition	22
4.2	Multipath device settings	22
4.3	Working with attached volumes	23
4.3.1	Logical Volume Manager	23
4.3.2	SLES 12 Filesystems	24
4.3.3	Btrfs	24
4.3.4	XFS	24
4.3.5	Ext3/Ext4 filesystems	26
5	Performance considerations	27



5.1	Elevator algorithms	27
5.2	Multiple volumes	27
5.3	SCSI UNMAP/TRIM	28
5.4	HBA queue depth	28
5.5	SCSI queue variables	29
5.5.1	nr_requests	29
5.5.2	read_ahead_kb	29
5.5.3	The kernel I/O scheduler	29
6	Volumes and persistence	30
6.1	Creating a new filesystem and volume label	30
6.2	Adding or changing the volume label of an existing filesystem	30
6.3	Discover existing labels	31
6.4	Swap space	31
6.5	Universally unique identifiers	32
6.6	GRUB2	33
6.7	Unmounting volumes	33
6.8	SCSI UNMAP/TRIM and filesystems	34
6.9	SCSI UNMAP/TRIM and LVM configuration	34
7	Useful tools	35
7.1	lsscsi	35
7.2	lspsci	35
7.3	scsi_id	36
7.4	/proc/scsi/scsi	37
7.5	/proc/mounts	37
7.6	/sys/block/sdX/queue	37
7.7	/sys/class/fc_host/hostX	38
7.8	dmesg	38
A	Additional resources	39
A.1	Technical support and resources	39
A.2	Related documentation	39



Executive summary

SUSE™ Linux® Enterprise Server (SLES) is a versatile server operating system for deploying highly available enterprise-class information technology services in heterogeneous environments with exceptional performance and reduced risk. Using the best practices presented in this publication, the SLES operating system provides an optimized experience for use with the Dell™ Storage SC Series arrays. These best practices include guidelines for configuring volume discovery, multipath, filesystem and queue depth management.

This paper discusses some features within version 12 of SLES in conjunction with the Dell Storage Center Operating System (SCOS). This paper is presented as a reference for experienced end users and system administrators, rather than an exhaustive approach to all possible avenues of administration.

Most of the information provided is from the SLES 12 command-line interface (CLI). The CLI is available to all UNIX and Linux operating systems, and is often the most effective means to accomplish the desired result.

This paper covers both Fibre Channel and iSCSI technology for front-end connectivity, however SCOS also supports SAS and FCoE (Fibre Channel over Ethernet). For related material about SAS front-end connectivity, read [Dell Storage Center with Red Hat Enterprise Linux 7x Best Practices](#).

Note: Test any changes in the production environment in a non-production setting first if at all possible. Always follow good change management and backup practices.



1 SLES 12 and SC Series overview

SC Series arrays provide Linux-compatible and SCSI-3 compliant disk volumes that remove the complexity of allocating, administering, using and protecting mission-critical data. They eliminate the need for cumbersome physical disk configuration activities and management, as well as complex RAID configuration. The SC Series arrays also provide RAID 10 speed and reliability at the storage layer so that volumes do not need to be further RAID-managed within the Linux operating system layer. The full range of Linux utilities such as mirroring, backup, multiple filesystems, multipath, boot from SAN, and disaster recovery are available to the Administrator with SC Series volumes.

1.1 New to SLES 12

SLES 12 offers many innovative changes to the operating system including:

- Improved management capabilities with full system rollback by utilizing features found in the copy-on-write btrfs filesystem of the default filesystem for the operating system partition.
- The btrfs provides writable snapshots for easy rollback, subvolume support and online check and repair function as well as many others. A subvolume is a file namespace that can be independently mounted, unlike an LVM logical volume that is an independent block device. For details about btrfs, see the [BTRFS Wiki](#) and the [SLES 12 documentation](#).
- SLES also introduces the XFS filesystem as the default filesystem for data volumes. XFS is a high-performance 64-bit journaling filesystem that is very good at manipulating large files and performs well on SC Series arrays.
- iSCSI and FC targets are Implemented from the kernel instead of the user space.

Details about SLES are located in the [SUSE Linux Enterprise Server 12 Documentation](#) that can be downloaded from suse.com.

1.2 Patch and service patch levels

Dell strongly recommends applying all SLES 12 patches. SLES 12 SP1 became available to the public in December 2015.

1.3 Special filesystems

SLES 12 has virtual and special filesystems that provide a hierarchical structural view into kernel structures and process data. The filesystems provide a convenient and standardized method for dynamic access to process data (such as in the `/proc` filesystem) or information about kernel subsystems, hardware devices and associated device drivers. The `/sys` filesystem in Linux provides the administrator with information and configuration details for the kernel subsystems, hardware devices and device drivers. Typically, each disk block device has an entry within the `/sys/block` directory, and each Host Bus Adapter (HBA) has an entry in `/sys/class/scsi_host/hostx`, where `x` is the HBA number in the server. Access to these filesystems is useful when administering SLES 12 systems in conjunction with SC Series storage.



2 Fibre Channel

The section includes the installation of SLES 12 to a multipath volume on a SC Series arrays, working with the Device-Mapper, expanding a multipath volume, and HBA module settings.

Information about Fibre Channel connectivity and other topics beyond the scope of this document can be located at dell.com/support/. For additional information about Multipath in SLES 12, see section 4.2 "[Multipath device settings](#)".

2.1 Boot from SAN

The ability to use SC Series LUNs as bootable volumes in Linux allows administrators to leverage the strengths of SC Series Replay and Replay View volume technologies. Among other uses, Linux boot volumes Replays provide a backup/recovery mechanism to preserve the state of an operating system at a point in time before upgrades. For a LUN to be bootable, Fibre Channel HBAs expect a target LUN ID of zero. Refer to the video titled [Creating/Mapping a Volume in Enterprise Manager](#) for this procedure.

2.2 Installing SLES 12 to a multipath volume

The SLES 12 installer detects when an SC Series array volume is presented to the server over multiple paths. The installer automatically configures the Linux Device Mapper to present several paths to a single disk device.

When the installation process asks, elect to activate multipath on capable devices. This option is presented earlier in the installation compared to previous versions of SLES.

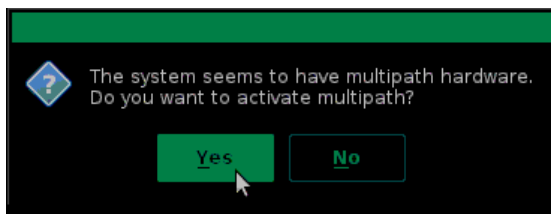


Figure 1 Activating multipath

Continue through the required steps until the **Suggested Partitioning** screen is displayed (Figure 2). Because the btrfs system permits subvolumes within a partition, the output for the pre-installation partitioning is complex.

After installing SLES 12, the output for the disk free (df) command and the entries in the /etc/fstab are on subvolumes (such as /var, /opt, or /usr/local). For example:

```
# df -h
Filesystem                                Size  Used Avail Use% Mounted on
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /
devtmpfs                                    6.9G     0   6.9G   0% /dev
tmpfs                                       6.9G   80K   6.9G   1% /dev/shm
tmpfs                                       6.9G   10M   6.9G   1% /run
tmpfs                                       6.9G     0   6.9G   0% /sys/fs/cgroup
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /.snapshots
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8%
/boot/grub2/x86_64-efi
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/tmp
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/spool
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8%
/boot/grub2/i386-pc
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/opt
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/log
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/lib/pgsql
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/lib/named
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8%
/var/lib/mailman
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /var/crash
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /usr/local
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /tmp
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /opt
/dev/mapper/36000d31000fd100000000000000000ff-part1 40G   3.1G   37G   8% /srv
/dev/mapper/36000d31000fd100000000000000000ff-part2 60G   34M   60G   1% /home
```



During installation, the **Suggested Partitioning** screen reflects this; subvolumes are created and the appropriate notations are made in the `/etc/fstab` file.

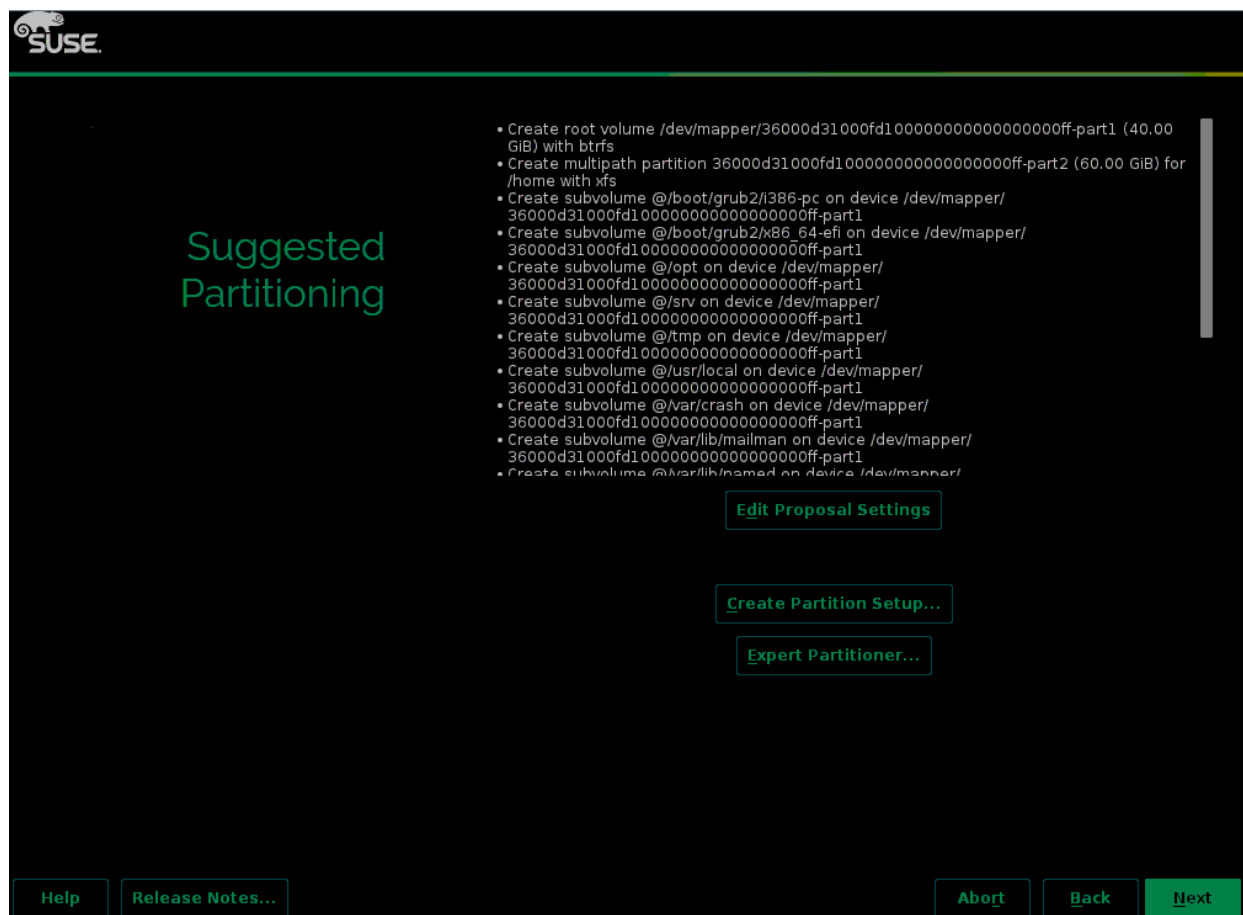


Figure 2 Suggested Partitioning screen

Note: Leave the default fstab settings suggested and mount the filesystems using the device mapper path. SLES 12 is optimized to use the btrfs filesystem; the `initrd` file contains the drivers necessary to mount and use it.

If averting from this best practice is required, the default partitioning scheme can be altered by selecting **Expert Partitioner**.

2.3 Enabling multipath on an existing installation

If Device Mapper is not running on the system, start it and ensure it remains running between boots. The Device Mapper facility in SLES 12 is the multipath daemon (`multipathd`). The `systemctl` command starts, stops and enables system daemons, services, scripts, and sockets.

```
# systemctl start multipathd.service
```

By default, the service does not start after rebooting. Use the `enable` command in `systemctl` to make the `multipathd.service` persistent between reboots.

```
# systemctl enable multipathd.service
```

To check the status of the service, use the `status` command in `systemctl`.

```
# systemctl status multipathd.service
multipathd.service - Device-Mapper Multipath Device Controller
   Loaded: loaded (/usr/lib/systemd/system/multipathd.service; enabled)
   Active: active (running) since Mon 2015-12-21 15:31:27 EST; 7 days ago
     Process: 26228 ExecReload=/sbin/multipathd reconfigure (code=exited,
status=0/SUCCESS)
     Process: 12400 ExecStartPre=/sbin/modprobe dm-multipath (code=exited,
status=0/SUCCESS)
    Main PID: 12402 (multipathd)
       Status: "running"
      CGroup: /system.slice/multipathd.service
             └─12402 /sbin/multipathd -d -s
```

Details about Device Mapper are located in section 4, "[Linux Device Mapper](#)".

2.4 Adding a volume

This section explains adding a LUN after the installation of SLES 12. For specifics regarding connecting to servers, cabling and Fibre Channel zoning, see the *Storage Center Deployment Guide*. Directions for creating a volume, assigning access control and mapping it to a server are in the [Creating and Mapping a Volume in Enterprise Manager](#) video. Additional information is available in the [Dell Enterprise Manager 2015 R3 Administration Guide](#). In this example, Dell Storage Enterprise Manager (EM) created a 20 GB volume with LUN ID 1 that was mapped to the server.

To scan the Fibre Channel HBAs in SLES 12 use the command:

```
# echo "- - -" >> /sys/class/scsi_host/host0/scan
```

Run the `echo` command for each system HBA and verify that the `sg3_utils` package is installed. The package includes the `rescan-scsi-bus.sh` script, which is useful for rescanning the HBAs.

Use either the `echo` command or the `rescan-scsi-bus.sh` script so that the kernel will detect the new LUN and add it to the list of available storage. View the new LUN by running the `dmesg` command.



Note: The number 1 in the output indicates the ID of new LUN that was just detected by the kernel.

Sample `dmesg` output for a path designated as the device `sdk`:

```
[65062.833555] scsi 0:0:0:1: Direct-Access      COMPELNT Compellent Vol    0606 PQ: 0 ANSI:
5
[65062.833866] scsi 0:0:0:1: alua: supports implicit TPGS
[65062.833871] scsi 0:0:0:1: alua: target naa.5000d31000fd1000 port group f03b rel port
3b
[65062.844023] scsi 0:0:0:1: alua: Attached
[65062.844289] sd 0:0:0:1: Attached scsi generic sg12 type 0
[65062.844350] sd 0:0:0:1: [sdk] 41943040 512-byte logical blocks: (21.4 GB/20.0 GiB)
[65062.844353] sd 0:0:0:1: [sdk] 4096-byte physical blocks
[65062.844962] sd 0:0:0:1: [sdk] Write Protect is off
[65062.844965] sd 0:0:0:1: [sdk] Mode Sense: 8f 00 00 08
[65062.845087] sd 0:0:0:1: [sdk] Write cache: disabled, read cache: enabled, doesn't
support DPO or FUA
[65062.852138] sd 0:0:0:1: alua: target naa.5000d31000fd1000 transition timeout set to 60
seconds
[65062.852142] sd 0:0:0:1: alua: target naa.5000d31000fd1000 port group f03b state A non-
preferred supports toluSNA
[65062.856958] sd 0:0:0:1: [sdk] Attached SCSI disk
```

The `multipath` command returns similar information about the new volume. The new volume has a LUN ID of 1, and is 20 GB.

```
# multipath -ll
36000d31000fd10000000000000000000100 dm-3 COMPELNT,Compellent Vol
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
   |- 0:0:0:1 sdk 8:160 active ready running
   |- 0:0:3:1 sdl 8:176 active ready running
   |- 0:0:4:1 sdm 8:192 active ready running
   |- 0:0:7:1 sdn 8:208 active ready running
   |- 6:0:0:1 sdo 8:224 active ready running
   |- 6:0:3:1 sdp 8:240 active ready running
   |- 6:0:4:1 sdq 65:0  active ready running
   `-- 6:0:7:1 sdr 65:16 active ready running
```

2.5 Expanding a volume

An SC Series volume can be expanded to accommodate more data. The SLES operating system does not automatically detect this change after the size of the volume has been changed in EM. For the change to be detected, rescan the HBAs by using the following `for` loop. It will scan each of the eight devices (`sdk` through `sdr`) displayed in the output of the previous `multipath` command.



```
# for i in sdk sdl sdm sdn sdo sdp sdq sdr
do
echo 1 > /sys/block/${i}/device/rescan
done
```

Sample dmesg output from the device sdk. The kernel detected the size change from 20 GB to 25 GB.

```
[67497.667569] sd 0:0:0:1: [sdk] 52428800 512-byte logical blocks: (26.8
GB/25.0 GiB)
[67497.667576] sd 0:0:0:1: [sdk] 4096-byte physical blocks
[67497.668359] sdk: detected capacity change from 21474836480 to
26843545600
```

Next, reload multipathd so the daemon acknowledges the change.

```
# systemctl reload multipathd
# multipath -ll
36000d31000fd100000000000000000100 dm-3 COMPELNT,Compellent Vol
size=25G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 0:0:0:1 sdk 8:160 active ready running
  |- 0:0:3:1 sdl 8:176 active ready running
  |- 0:0:4:1 sdm 8:192 active ready running
  |- 0:0:7:1 sdn 8:208 active ready running
  |- 6:0:0:1 sdo 8:224 active ready running
  |- 6:0:3:1 sdp 8:240 active ready running
  |- 6:0:4:1 sdq 65:0 active ready running
  `- 6:0:7:1 sdr 65:16 active ready running
```

2.6 Fibre Channel HBA optimization

Each HBA vendor has their own set of best practices that should be consulted prior to making changes. SLES 12 host stability and usability may require changes to the default HBA settings. Refer to the vendor documentation for tunable parameters of each device. A brief description can be obtained by using the modinfo command. The following modinfo output sample is for a Qlogic HBA.

```
# modinfo qla2xxx
<output truncated>
parm: ql2xlogintimeout:Login timeout value in seconds. (int)
parm: qlport_down_retry:Maximum number of command retries to a
port that returns a PORT-DOWN status. (int)
<output truncated>
```



2.6.1 Optimal HBA settings

There are several ways to store the Fibre Channel module settings on a SLES 12 system. For boot-from-SAN systems, these changes must be incorporated in the ramdisk. For non-boot-from-SAN systems, they can be placed in system files.

SLES 12 uses version 2 of the Grand Unified Bootloader. GRUB2 possesses characteristics such as a new shell-like syntax that allows advanced scripting capabilities.

Note: Unified extensible firmware Interface systems require GRUB2-EFI.

When altering the GRUB2 boot options, specify the module settings prior to loading the kernel with the `yast2` wizard **System > Boot Loader > Edit** menu command. Add the queue depth and port down retry settings by clicking **Kernel Parameters > Optional Kernel Command Line Parameters**. The new ramdisk is created and installed automatically.

For non-boot-from-SAN systems, change the module settings in the `/etc/modprobe.d/99-local.conf` or `/etc/modprobe.d/XX-DRIVENAME.conf` file and then unload and reload the module with the `rmmod` and `modprobe` commands.

2.6.2 Port connectivity timeout

Port connectivity timeout can be critical in cases of an SC Series controller failover. In such an instance, the World Wide Name (WWN) for the active port momentarily disappears from the fabric before it is available on the controller. A port that is moving between controllers can be unavailable throughout the fabric for as little as five seconds to as much as 60 seconds. The default time out value for either Qlogic or Emulex HBAs is 30 seconds. However, because 30 seconds is not always sufficient, increase the value to 60 seconds.

An exception to this best practice is an environment using multipath. See the section 4.2, "[Multipath device settings](#)" for more information.

To view the current timeout value, use the HBA timeout value file in the `/sys` filesystem.

- For Qlogic:

```
# cat /sys/module/qla2xxx/parameters/ql2xlogintimeout
60
```

- For Emulex:

```
# cat /sys/class/scsi_host/hostX/lpfc_nodev_tmo
60
```



2.6.3 Queue depth

Adjusting the queue depth to make I/O performance more efficient will also increase latency and throughput. Modify and monitor the settings to obtain an acceptable balance. Keep in mind that one SC Series array connected to multiple servers has a finite capacity to queue incoming I/O. An increased queue capacity on one system may negatively impact others.

Queue depth settings for Fibre Channel HBAs are in the BIOS and configuration files used by the Linux kernel that are managed by the modprobe facility. If the queue depth value is set differently in the two places, the lesser of the two values takes precedence. As a best practice, use the BIOS and set the queue depth to a high value in the operating system. A recommended starting point for the queue depth setting in the operating system is 128. From there, test the value and adjust accordingly.

`ql2xmaxqdepth` is the value that controls the maximum queue depth per LUN when interacting with the OS. Display the current `ql2xmaxqdepth` value with the command:

```
# cat /sys/module/qla2xxx/parameters/ql2xmaxqdepth
32
```

2.6.4 Extended logging

While logging is helpful during a debugging process, it can impact performance during normal operations. To enable extended logging, `ql2xextended_error_logging` can be used for Qlogic HBAs and `lpfc_log_verbose` for Emulex HBAs.

- To view the current value:

Qlogic:

```
# cat /sys/module/qla2xxx/parameters/ql2xextended_error_logging
0
```

Emulex HBA

```
cat /sys/class/scsi_host/hostX/lpfc_log_verbose
```

- To turn logging on:

```
# echo 1 > /sys/module/qla2xxx/parameters/ql2xextended_error_logging
```

- To turn logging off:

```
# echo 0 > /sys/module/qla2xxx/parameters/ql2xextended_error_logging
```



3 iSCSI

iSCSI technology is a standard technology used for enterprise block storage. SLES 12 uses an updated implementation of the RFC 3720 open-iSCSI stack. This technology permits organizations to scale their block storage infrastructure while leveraging existing infrastructure. The open-iSCSI implementation is the only implementation discussed here. For other vendor-provided implementations, refer to the vendor specific documentation. Information about iSCSI topologies can be found in the SC Series connectivity instructions.

iSCSI protocol requires a network port to communicate with the SC Series arrays. A dedicated network port or a dedicated VLAN for iSCSI traffic is critical. The type of data determines the topology. Data that is sensitive or confidential is treated differently than data that requires immediate, high availability and low latency. These needs drive architecture configurations such as the dedication of ports, VLAN usage, multipathing, and redundancy.

Routine TCP/IP data ideally uses separate paths from iSCSI traffic. Even better, are 10 GB switches dedicated to iSCSI traffic, distinct from 1 GB switches for other server traffic. Examples of constrained architectures might include the use of VLAN-tagged traffic, with iSCSI network traffic tagged differently from general traffic. Whenever possible, use multipath for iSCSI data for redundancy. In the absence of VLAN tagging, different traffic can be routed to different destinations with static routing or at the iSCSI level in the configuration.

3.1 Host configuration

Consider a scenario of a single host and a single SC8000 array. A pair of 10 GB Ethernet interfaces are configured to communicate with the SC8000, and a single 1-GB interface is used for routine traffic.

First, the appropriate iSCSI unit files on the SLES 12 server need to be started. Persistent availability requires enabling the unit files.

```
# systemctl start iscsi.service
# systemctl start iscsid.socket
# systemctl enable iscsi.service
# systemctl enable iscsid.socket
```

Configure the Network Interfaces in yast2. In the following example, the interfaces are configured to be on the same network as the SC8000 using discovery addresses 10.10.6.82 and 10.10.6.83.

```
# ifconfig -a
<output truncated>
eth1      Link encap:Ethernet  HWaddr 00:21:9B:8A:E2:85
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```



```

eth2      Link encap:Ethernet  HWaddr A0:36:9F:7E:0E:8C
          inet addr:10.10.6.176  Bcast:10.10.255.255  Mask:255.255.0.0
          inet6 addr: fe80::a236:9fff:fe7e:e8c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:86781 errors:0 dropped:0 overruns:0 frame:0
          TX packets:45194 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:9292103 (8.8 Mb)  TX bytes:38934256 (37.1 Mb)

eth3      Link encap:Ethernet  HWaddr A0:36:9F:7E:0E:8E
          inet addr:10.10.6.177  Bcast:10.10.255.255  Mask:255.255.0.0
          inet6 addr: fe80::a236:9fff:fe7e:e8e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:62333 errors:0 dropped:0 overruns:0 frame:0
          TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:5884325 (5.6 Mb)  TX bytes:816 (816.0 b)

```

```
# netstat -nr
```

```
Kernel IP routing table
```

Destination	Gateway	Genmask	Flags	MSS Window	irtt	Iface
0.0.0.0	10.124.4.1	0.0.0.0	UG	0 0	0	eth0
10.10.0.0	0.0.0.0	255.255.0.0	U	0 0	0	eth2
10.10.0.0	0.0.0.0	255.255.0.0	U	0 0	0	eth3
10.124.4.0	0.0.0.0	255.255.252.0	U	0 0	0	eth0

With the two 10 GB interfaces configured, ping the discovery IP addresses to test connectivity. The `-c` switch limits the number of pings to one, and the `-w` switch limits the wait time for a return packet to one second (plenty of time on the single-hop network used in this example).

```
# ping -c 1 -w 1 10.10.6.82
```

```
PING 10.10.6.82 (10.10.6.82) 56(84) bytes of data.
64 bytes from 10.10.6.82: icmp_seq=1 ttl=64 time=0.045 ms
```

```
--- 10.10.6.82 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.045/0.045/0.045/0.000 ms
```

```
# ping -c 1 -w 1 10.10.6.83
```

```
PING 10.10.6.83 (10.10.6.83) 56(84) bytes of data.
64 bytes from 10.10.6.83: icmp_seq=1 ttl=64 time=0.042 ms
```

```
--- 10.10.6.83 ping statistics ---
```

```
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.042/0.042/0.042/0.000 ms
```



The `iscsiadm` command is used to request the SC8000 target IQNs and require the Linux host login to these targets.

```
# iscsiadm -m discovery -t sendtargets -p 10.10.6.82:3260
10.10.6.82:3260,0 iqn.2002-03.com.compellent:5000d31000fd1018
10.10.6.82:3260,0 iqn.2002-03.com.compellent:5000d31000fd102c







# iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd1018, portal: 10.10.6.82,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd102c, portal: 10.10.6.82,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd102b, portal: 10.10.6.85,3260] (multiple)
Logging in to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd1017, portal: 10.10.6.85,3260] (multiple)
Login to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd1018, portal: 10.10.6.82,3260] successful.
Login to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd102c, portal: 10.10.6.82,3260] successful.
Login to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd102b, portal: 10.10.6.85,3260] successful.
Login to [iface: default, target: iqn.2002-
03.com.compellent:5000d31000fd1017, portal: 10.10.6.85,3260] successful.
```

The iSCSI HBA should now be identifiable on the SC8000, with the same iSCSI name as in `/etc/iscsi/initiatorname.iscsi`.

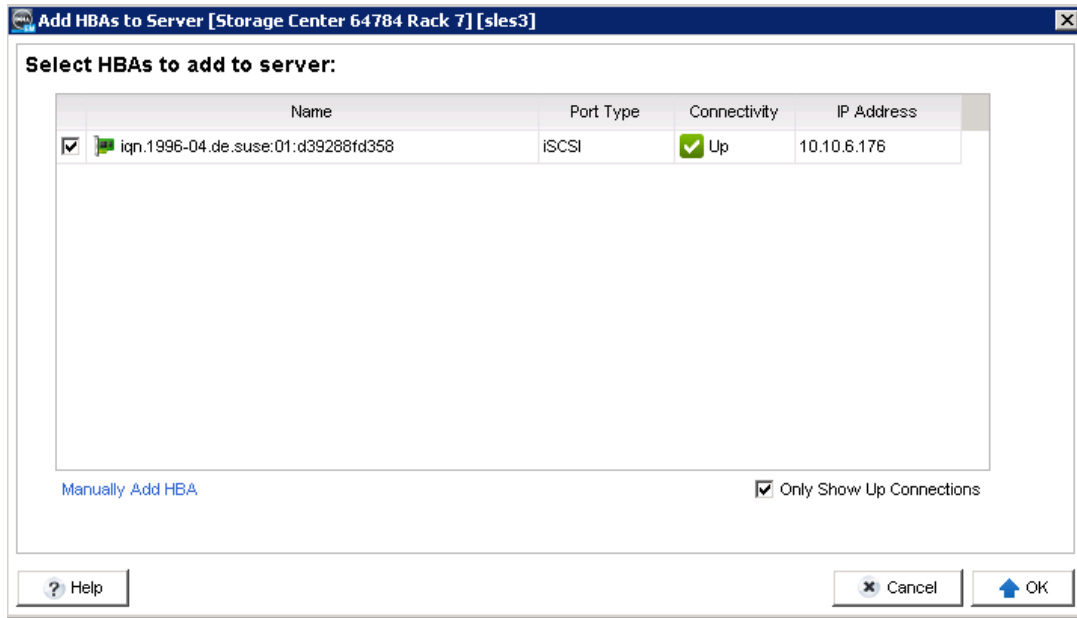
```
# cat /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1996-04.de.suse:01:d39288fd358
```

In EM, adding a new HBA to the server results in an HBA with the same name.

1. Open the **Storage** tab.
2. Expand **Servers** in the middle column.
3. Select the server corresponding to the SLES 12 host where the `iscsiadm` command was run.
4. The list of **Server HBAs** is displayed in the middle of the screen on the right side.

Server HBAs		
Name	Port Type	Connectivity
 2001000E1E09E24A	Fibre Channel	 Up
 2001000E1E09E24B	Fibre Channel	 Up
 iqn.1996-04.de.suse:01:d39288fd358	iSCSI	 Up

5. Click **Add HBAs to Server**.
6. Select the available HBA and click **OK**.



3.2 Adding a volume

For this example, a 10 GB volume was created with EM and mapped to the example server. Information about creating a volume, assigning access control, and mapping it to a server are explained in [Creating and Mapping a Volume in Enterprise Manager](#). The 10 GB volume was created on the SC8000 and mapped to the server with the name `iqn.1996-04.de.suse:01:d39288fd358`. Rescan the server iSCSI bus using the `iscsiadm` command.

```
# iscsiadm -m node -R
Rescanning session [sid: 13, target: iqn.2002-
03.com.compellent:5000d31000fd102b, portal: 10.10.6.85,3260]
Rescanning session [sid: 14, target: iqn.2002-
03.com.compellent:5000d31000fd1017, portal: 10.10.6.85,3260]
Rescanning session [sid: 15, target: iqn.2002-
03.com.compellent:5000d31000fd1018, portal: 10.10.6.82,3260]
Rescanning session [sid: 16, target: iqn.2002-
03.com.compellent:5000d31000fd102c, portal: 10.10.6.82,3260]
```

Running the multipath command returns the new 10 GB volume that is available.

```
# multipath -ll
<output truncated>
36000d31000fd100000000000000000101 dm-4 COMPELNT,Compellent Vol
size=10G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
  |- 20:0:0:2 sdt 65:48 active ready running
  |- 23:0:0:2 sdx 65:112 active ready running
  |- 22:0:0:2 sdy 65:128 active ready running
```

```
`- 21:0:0:2 sdz 65:144 active ready running
```

At this point, a filesystem can be added to the iSCSI LUN, a mount point created, and an entry written to the `/etc/fstab` file. It is a best practice to use the entire drive without partition for all non-boot drives. Doing so leverages the SC Series array native strengths of wide striping the volume across all disks in the tier provisions.

```
# mkfs.xfs /dev/dm-4
meta-data=/dev/dm-4          isize=256    agcount=17, agsize=163328
blks
        =                    sectsz=4096   attr=2, projid32bit=1
        =                    crc=0         finobt=0
data      =                    bsize=4096   blocks=2620928, imaxpct=25
        =                    sunit=512     swidth=512 blks
naming    =version 2          bsize=4096   ascii-ci=0 ftype=0
log       =internal log      bsize=4096   blocks=2560, version=2
        =                    sectsz=4096   sunit=1 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
# xfs_admin -u /dev/dm-4
UUID = 1eeea9ea-3b20-42e2-8b1d-17c272f8202c
```

Verify that the entry in the `/etc/fstab` file is similar to:

```
UUID=1eeea9ea-3b20-42e2-8b1d-17c272f8202c /opt/new xfs defaults,_netdev 1 2
```

Adding the `_netdev` flag in `/etc/fstab` ensures the filesystem is mounted after the network is up and running.

3.3 Expanding a volume

After a volume is grown (See section 2.5, “Expanding a volume”), use the `iscsiadm` command to rescan and detect the size change.

```
# iscsiadm -m node -R
```

Once the iSCSI subsystem has been resized, the `multipathd` daemon must be reloaded to detect the change and apply the size increase to the multipath device.

```
# systemctl reload multipathd.service
```

3.4 iSCSI timeout values

If an SC Series controller fails in a single-path volume, the functionality of the failed unit is moved to a functioning unit. Failover requires approximately 30 seconds to complete, therefore configure the iSCSI daemon to wait 60 seconds before failing the connection completely.

A multipath environment is much more resilient. The iSCSI daemon can be configured to fail a path very quickly. Once a path fails, outstanding I/O is resubmitted to the other active routes. If all paths are down,



then I/O queues until a route become available. This permits the storage environment to sustain network-level and storage-level failures.

The following iSCSI daemon configuration settings directly affect iSCSI connection timeouts. The configurable values are in `/etc/iscsi/iscsid.conf`.

- To control the frequency of NOP-Out requests sent to each target, adjust the variable below, where `X` is in seconds; the default is five seconds.

```
node.conn[0].timeo.noop_out_interval = X
```

- To control the time out for the NOP-Out, use the following variable. This default is also five seconds. After the specified time has passed, the I/O fails back to the SCSI layer. When dm-multipath is in use, the I/O fails back to the multipath layer.

```
node.conn[0].timeo.noop_out_timeout = X
```

- To specify the length of time to wait for session reestablishment before failing SCSI commands back to the application when running the Linux SCSI Layer error handler, edit the `node.session.timeo.replacement_timeout` variable. The default is 120 seconds.

```
node.session.timeo.replacement_timeout = X
```

- If a network problem is detected, the running commands are failed immediately. The exception to this rule is when the error handler for the SCSI layer is running. The `iscsiadm` command can be used to check if the error handler is running.

```
# iscsiadm -m session -P 3
```

The amount of output depends on the number of iSCSI connections there are to the array, but there will be an entry for each link in the `/sys/class/iscsi_host` directory. The output might be as follows:

```
Host Number: X State: Recovery
```

- When the SCSI error handler is running, commands will not be failed until the `node.session.timeo.replacement_timeout` value has passed. This value is in seconds and the default is 120.

The timer can be directly altered with the value in the sysfs file (`X` is in seconds, and `Y` is the device file).

```
# echo X > /sys/block/sdY/device/timeout
```

In the example below the timeout value for `/dev/sdf` is set to 180 seconds.

```
# echo 180 > /sys/block/sdf/device/timeout
```

- There is an udev rule that can also be set. In the `/etc/udev/rules.d/60-raw.rules` add the following line to make the change to 180 seconds persistent.



```
ACTION=="add", SUBSYSTEM=="scsi" , SYSFS{type}=="0|7|14", \  
RUN+="/bin/sh -c 'echo 180 > /sys$DEVPATH/timeout'"
```

Note: When modifying iSCSI timeout values, it is important to test and verify as many failure scenarios as possible before entering a system into production. Timeout and other connection settings are statically created during the discovery step and written to configuration files in `/var/lib/open-iscsi/*`.

3.5 10 GB Ethernet and iSCSI

Ten gigabyte Ethernet technology has been available for servers many years and is included standard with many Dell PowerEdge offerings. The technology has matured to the point that 10GB Ethernet adapters and switches are commodity hardware. To accommodate the demands of 10-GB Ethernet, the Linux kernel version 2.6.17 introduced full TCP buffer autotuning with a 4 MB maximum buffer size.

Information about tuning is available in the [System Analysis and Tuning Guide](#) from the [SuSE Linux Enterprise Server 12 documentation site](#).

Of all the tunable parameters jumbo frames, whether used with 1 GB or 10 GB Ethernet, are likely to provide the most performance difference. The customary jumbo frame size is 9000 bytes. Check with the HBA and switch vendor documentation to determine if the larger frame size is an option. Whatever the frame size is, set it to the exact same value on each piece of equipment being used, especially the server HBA, the Ethernet switches and the SC Series Ethernet ports.

Note: Be certain the SC Series, switches and server HBA support the specified frame size, and are all set to the same frame size.

The network configuration file in `/etc/sysconfig/network/ifcfg-ethX` houses the configurable settings for each network interface including the maximum transmission unit (MTU). The YaST setup and configuration tool is also available for updating the MTU value.

The tunable kernel parameters are accessible in `/proc/sys/net/core` and `/proc/sys/net/ipv4`. Store the optimized values in `/etc/sysctl.conf`. Some default values are set conservatively low; parameters to consider are:

- Linux Auto-tuning buffer limits:
 - `net.ipv4.tcp_sack`
 - `net.ipv4.tcp_window_scaling`
 - `net.ipv4.tcp_timestamps`
- TCP Max Buffer Sizes
 - `net.core.rmem_max`
 - `net.core.wmem_max`



4 The Linux Device Mapper

The Linux Device Mapper is a flexible yet generic framework that provides interconnected virtual block devices on top of physical storage devices. One of its most powerful feature sets (DM-multipath) is the ability to detect, create and monitor block devices with multiple paths to a backing storage device. DM-multipath, dm_mpath or multipath are frequent abbreviations. Device Mapper works with a set of default parameters that can be adjusted in the `/etc/multipath.conf` configuration file.

4.1 SC Series device definition

Device Mapper is not vendor or transport specific. It can manage devices that are directly attached as well as those using Fibre channel or iSCSI devices. The kernel version used by SLES 12 includes the SC Series device definition by default. Little effort is required to configure simple multipath environments.

After the host SCSI subsystem detects an SC Series volume, SLES multipath automatically creates a multipath device based on the SCSI ID of the volume. Settings in the `/etc/multipath.conf` file pertinent to the volume and any appropriate defaults for a multipath volume are automatically applied.

There are two important default Device Mapper volume settings:

- `path_checker tur`

Ensures the SCSI `Test Unit Ready` command is used to monitor end-device health.

- `no_path_retry queue`

Works when multipath has no healthy paths to a LUN. In this case, I/O is queued until a path is available.

4.2 Multipath device settings

Multipath creates a device for every unique SCSI ID. Each device that multipath creates has settings that differentiate it from the others and stores them in the `/etc/multipath.conf` configuration file. The operating system installation process does not create a configuration file because there are default settings in the kernel. Create a multipath device with a user-friendly name in a `/etc/multipath.conf` file; the example below provides a good starting point. There are excellent example files stored in `/usr/share/doc/packages/multipath-tools/` that explain the options available for the Device Mapper. The below simple multipath configuration file assigns an easy-to-read alias to the device instead of the long SCSI ID string.

```
multipaths {
    multipath {
        wwid 36000d31000fd100000000000000000ff
        alias FileShare
    }
}
```



After adding the alias entry, reload the multipath daemon for the change to take effect.

```
# systemctl reload multipathd.service
```

Before changing the /etc/multipath.conf file, the output from `multipath -ll` shows the name.

```
# multipath -ll
36000d31000fd100000000000000000100 dm-3 COMPELNT,Compellent Vol
size=250G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
    |- 0:0:0:1   sdk 8:160   active ready  running
    |- 0:0:3:1   sdl 8:176   active ready  running
    |- 0:0:4:1   sdm 8:192   active ready  running
    |- 0:0:7:1   sdn 8:208   active ready  running
    |- 6:0:0:1   sdo 8:224   active ready  running
    |- 6:0:3:1   sdp 8:240   active ready  running
    |- 6:0:4:1   sdq 65:0    active ready  running
    `-- 6:0:7:1   sdr 65:16   active ready  running
```

After the change to the multipath.conf file and reloading the multipath daemon, change the name of the LUN and /dev/mapper associated files to the alias listed in the configuration file.

```
# multipath -ll
FileShare (36000d31000fd100000000000000000100) dm-3 COMPELNT,Compellent Vol
size=250G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
    |- 0:0:0:1   sdk 8:160   active ready  running
    |- 0:0:3:1   sdl 8:176   active ready  running
    |- 0:0:4:1   sdm 8:192   active ready  running
    |- 0:0:7:1   sdn 8:208   active ready  running
    |- 6:0:0:1   sdo 8:224   active ready  running
    |- 6:0:3:1   sdp 8:240   active ready  running
    |- 6:0:4:1   sdq 65:0    active ready  running
    `-- 6:0:7:1   sdr 65:16   active ready  running
```

4.3 Working with attached volumes

While using SC Series volumes, consider whether or not to use tools available in SLES 12 for working with attached volumes and decide on the filesystem type for the given volume.

4.3.1 Logical Volume Manager

Logical Volume Manager (LVM) places a layer of abstraction between the filesystem and the hardware to enable more advanced disk management in a Linux environment. However, SC Series arrays provide a similar functionality that also offloads the overhead associated with LVM from the Linux system. For this reason, do not use LVM with SC Series arrays.



4.3.2 SLES 12 Filesystems

SLES 12 introduced the b-tree filesystem (btrfs) as the default filesystem for the root volume while data volumes are formatted with the XFS filesystem by default. Btrfs and snapshot support for the root/boot partition are default tools used to set up SLES 12.

4.3.3 Btrfs

Btrfs includes writable snapshots, subvolume support, online check and repair functionality, and offline migration from existing ext2, ext3, or ext4 filesystems. There is also bootloader support for /boot, which permits booting from a btrfs partition. Use the btrfs filesystem for the root/boot partition as a best practice.

Creating a btrfs filesystem is very similar to other types of Linux filesystems. Any SLES 12 filesystem should occupy an entire block device in every case except the root/boot volume, which does require partitions.

```
# mkfs.btrfs /dev/sdk
```

The filesystem size can be expanded using the btrfs utility. For example, if the LUN FileShare above was resized from 250 GB to 300 GB, the space would be increased by adding the 50 GB.

```
# btrfs filesystem resize +50G /fileshare
Resizing '/fileshare' of '+50G'
```

To see the change, use the show subcommand within the btrfs utility as shown below.

```
# btrfs filesystem show /fileshare
Label: none  uuid: f04c511f-4c17-4b5e-922a-4c098aad11a6
    Total devices 1 FS bytes used 21.90GiB
    devid    1 size 300.00GiB used 21.90GiB path /dev/mapper/FileShare

Btrfs v3.16+20140829
```

4.3.4 XFS

Enterprise Manager was used to create and map another 500 GB volume (LUN ID 3) to the example server from the **FileShare2** SC Series array.

XFS is the preferred filesystem for data volumes on SLES 12. It is a stable, journaling filesystem specifically designed to work with high-performance storage and massive file sets. XFS is excellent for manipulating large files and performs well on high-end hardware such as SC Series arrays. It is the default filesystem for data partitions in SLES.

An XFS filesystem can be created on a volume using the `mkfs.xfs` command.

The multipath command output for the FileShare2 volume might look like the example below after creating it in Enterprise Manager and adding it to the server.

```
# multipath -ll
```



```

<output truncated>
FileShare2 (36000d31000fd10000000000000000000102) dm-6 COMPELNT,Compellent
Vol
size=500G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='service-time 0' prio=1 status=active
    |- 0:0:1:3 sdab 65:176 active ready running
    |- 0:0:2:3 sdac 65:192 active ready running
    |- 0:0:5:3 sdaf 65:240 active ready running
    |- 0:0:6:3 sdag 66:0 active ready running
    |- 6:0:1:3 sdaj 66:48 active ready running
    |- 6:0:2:3 sdak 66:64 active ready running
    |- 6:0:5:3 sdan 66:112 active ready running
    `-- 6:0:6:3 sdao 66:128 active ready running

```

Then the XFS is created without partitioning on the entire volume.

```

# mkfs.xfs /dev/mapper/FileShare2
meta-data=/dev/mapper/FileShare2 isize=256      agcount=17, agsize=8191488
blks
          =                               sectsz=4096 attr=2, projid32bit=1
          =                               crc=0      finobt=0
data      =                               bsize=4096 blocks=131072000, imaxpct=25
          =                               sunit=512   swidth=512 blks
naming    =version 2                     bsize=4096   ascii-ci=0 ftype=0
log        =internal log                  bsize=4096   blocks=64000, version=2
          =                               sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none                           extsz=4096   blocks=0, rtextents=0

```

If the volume is expanded, rescan the Fibre Channel bus and then use the `xfs_growfs` command.

```

# xfs_growfs /fileshare2
meta-data=/dev/mapper/FileShare2 isize=256      agcount=17, agsize=8191488
blks
          =                               sectsz=4096 attr=2, projid32bit=1
          =                               crc=0      finobt=0
data      =                               bsize=4096 blocks=131072000, imaxpct=25
          =                               sunit=512   swidth=512 blks
naming    =version 2                     bsize=4096   ascii-ci=0 ftype=0
log        =internal                     bsize=4096   blocks=64000, version=2
          =                               sectsz=4096  sunit=1 blks, lazy-count=1
realtime  =none                           extsz=4096   blocks=0, rtextents=0
data blocks changed from 131072000 to 144179200

```



4.3.5 Ext3/Ext4 filesystems

While the ext2, ext3 and ext4 filesystems have been supplanted by btrfs and XFS, they are still supported. These filesystems are easily transported between Linux systems, especially from SLES 12 to older versions. Between the ext3 and ext4 filesystems, ext4 offers more options and provides better performance.

As with other filesystems, the ext4 filesystem should be placed on top of the entire block device. For example:

```
# mkfs.ext4 -L FileShare3 /dev/sdba
```

Growing or shrinking the volume can be done with the ext4 filesystem mounted and online. However, the SC Series volume must first be expanded and then the Linux system devices rescanned.

```
# resize2fs /dev/sdba
```

5 Performance considerations

Each environment and workload set dictates if there are any tuning requirements. This section provides general direction regarding standard tuning options that are available in Linux as a starting point for determining how to achieve optimal performance. Understanding the applications that use the SC Series arrays and their individual and combined demands is key to beginning the process. This tuning process is often iterative using a few methods to evaluate effectiveness. Perception, application metrics, and the Dell Performance Analysis Collection Kit (DPACK) are all useful during tuning. DPACK is free and can be obtained by sending an email to DPACK_Support@Dell.com.

Performance variables are often common without regard to the transport mechanism, whether Fibre Channel, iSCSI, or otherwise. However, because iSCSI uses Ethernet, considerations for file storage and network tuning, as opposed to block storage, are important.

Note: Make manual Fibre Channel and iSCSI changes individually and incrementally. Evaluate them against multiple workload types to understand the effects on overall performance. iSCSI tuning is often more time consuming because of the block-level subsystem tuning considerations in conjunction with the network (Ethernet) tuning. A solid understanding of the various Linux subsystem layers and how they interact is necessary to tune the complex interaction between SCSI calls and network optimization.

5.1 Elevator algorithms

SLES 12 maintains thorough documentation about the system and I/O performance tuning. SLES 12 I/O performance tuning includes using scheduling controls that determine priority for submitting input and output operations to and from storage. SLES 12 offers various I/O algorithms (called elevators) fitted to differing workloads. The purpose of elevators is to reduce the number of seek operations, to prioritize requests and ensure I/O request completion before a specified deadline. Choosing the best I/O elevator depends on workload and hardware. For a complete discussion, see chapter twelve in [SLES 12 System Analysis and Tuning](#).

5.2 Multiple volumes

An SC Series volume is only active and presented on one controller at a time. When using multiple volumes on a Linux host, alternately present volumes to the Linux host across both controllers to balance I/O processing and load. Evaluate, adapt and test this method to meet the individual varied needs. A volume can be pinned to a specific controller in Dell Storage Enterprise Manager as instructed below.

1. Select the **Storage** tab on the top menu.
2. Expand **Volumes** and navigate to the appropriate volume.
3. Select **Mappings** in the middle tab bar on the right.
4. Select **Edit Settings** on the far right.
5. Read the modifying mappings caution and click **Continue**.
6. In the **Restrict Mapping Paths** section, deselect **Allow the Storage Center to automatically determine the best Controller to activate Volume on** and choose the appropriate controller.



5.3 SCSI UNMAP/TRIM

SC Series arrays use thin provisioning and often solid-state drives (SSDs). The ability to discard unused blocks is an efficient use of the equipment. The SCSI UNMAP/TRIM feature is a highly effective way of managing the storage and its cost of use in an enterprise environment. However, there are other considerations before implementing this function.

The discard mount parameter enables the filesystem to perform real-time, on-the-fly SCSI UNMAP commands to the SC Series array. While this is ideal in situations of high I/O, executing UNMAP commands will introduce additional I/O and CPU load. Frequent use of UNMAP command can also decrease the lifetime of SSDs because it increases the number of write operations used against the SSD.

The alternative to using the discard mount parameter is to use the `fstrim` command, which is part of the `util-linux` package. Using the `fstrim` command permits the asynchronous use of the UNMAP/TRIM feature that can occur during times of relative I/O calm, thereby avoiding performance degradation. The infrequent removal of unused blocks can also drastically reduce the number of write operations to SSDs in situations where a lot of data is written and deleted from the drives.

Use of the `fstrim` command can be scripted and added as a cron job. For example, if a server had filesystems `u01`, `u02`, `u03`, `u04` and `u05`, then the following script would perform UNMAP using the `fstrim` command on a nightly basis.

```
#!/bin/bash
FSTrim=/usr/bin/fstrim
MntPoints="u01 u02 u03 u04 u05"
for i in ${MntPoints}
do
echo "INFO: Applying ${FSTrim} to mount point ${i}"
${FSTrim} -v /${i}
done
```

Then the crontab entry would look like the following, assuming the script path was `/usr/local/bin/FStrimmer.sh`. (To view the cron table use `crontab -l`, to edit it use `crontab -e`.)

```
30 1 * * * /usr/local/bin/FStrimmer.sh
```

In this example, the script runs daily at 1:30 AM.

5.4 HBA queue depth

The queue depth is configurable in the HBA firmware or the Linux kernel module for the HBA. Keep in mind that if these two settings have different values, the lower value takes precedence. A good strategy to consider is setting the HBA firmware to the highest number allowable and then tuning this value downward from within the Linux kernel module (see section 0).



5.5 SCSI queue variables

SCSI device queue settings are tunable. Tuning them can improve performance. Discussed below are some common tunable settings. For multipath devices, the tunable values are in `/sys/block/dm-X/queue`. Block devices tunable settings are in `/sys/block/sdX/queue`. Performance enhancement is possible by modifying the values in the queue directories, with the caveat that changing one setting will affect others. There are 34 total tunable settings; most of these are beyond the scope of this paper. However, the `nr_requests`, `read_ahead_kb` and `scheduler` settings are described in the following sections.

5.5.1 `nr_requests`

The `nr_requests` setting used by the SLES 12 Linux kernel determines the depth of the request queue. The default value is 128 for SLES 12. The `nr_requests` setting compliments the HBA queue depth. The larger the `nr_requests` value, the greater the number of scheduled requests. The higher number keeps the I/O subsystem moving in one direction longer, potentially making handling disk I/O more efficient. A value of 512 or 1024 is a good place to start. Adjust up or down from there according to the resulting performance.

5.5.2 `read_ahead_kb`

The `read_ahead_kb` parameter defines the number of kilobytes of I/O the kernel reads from a block device when the read is sequential. In situations of frequent, or large sequential read workloads, a noticeable performance increase is probable. SLES 12 uses 512 as a default value. Using 2048 or 4096 is a good place to start for tuning this parameter.

5.5.3 The kernel I/O scheduler

The `schedule` parameter defines scheduling behavior for SCSI devices. Application vendors often have specific recommendations for setting the parameter for optimal performance with the program. SLES 12 has a default setting of `noop deadline [cfg]`.

For a detailed discussion, see [SLES 12 System Analysis and Tuning](#), specifically chapter 12, "Tuning I/O Performance".



6 Volumes and persistence

SLES 12 is capable of discovering multiple volumes from an SC Series array. The operating system will give new disks a device designation of `/dev/sdX`, where `X` can be any single or two letter combination (for example `a` or `zz`). Designation depends on the method of discovery, using the various interfaces connecting the server to storage.

Device names (such as `/dev/sda`) are used to designate the volumes in a myriad of commands and files. Probably the most important use of the device name is in the filesystem table (`/etc/fstab`) in Unix-like operating systems. The filesystem table correlates device names with mount points. In a static disk environment, the `/dev/sdX` device name works well in the filesystem table.

Disk labels are exceptionally useful when scripting replays for SC Series arrays. If a backup server maps a production volume, it is not necessary to know what drive letter the view volume is assigned. The program that created the filesystem can write a label to the filesystem making mounting and manipulating it easy.

Note: Disk labels will not work in a multipath environment. Multipath device names are persistent by default. Multipath does support aliases, which permits human-readable names. Labels are noted in this paper because they are a common part of how SLES systems are configured.

6.1 Creating a new filesystem and volume label

Note: This will format your volume. Formatting destroys all of the data on a volume.

The `mkfs.btrfs`, `mke2fs`, or `mkfs.xfs` commands take the option, `-L LabelName`. The commands create a new filesystem on the device and erase any previous filesystem tables. This destroys pointers to the existing files in order to create a new filesystem and label on the disk. The examples below cover several major filesystem types; each creates a filesystem with the label `FileShare`.

```
# mkfs.btrfs -L FileShare /dev/sdgt2
# mke2fs -j -L FileShare /dev/sdgt2
# mkfs -t ext3 -L FileShare /dev/sdgt2
# mkfs -t ext4 -L FileShare /dev/sdgt2
# mkfs.xfs -L FileShare /dev/sdgt2
```

6.2 Adding or changing the volume label of an existing filesystem

The following command string changes the label on a btrfs volume.

```
# btrfs filesystem label /dev/sdX FileShare
```

For the various extended filesystems (`ext2`, `ext3` and `ext4`) issue the following command:

```
# e2label /dev/sdX FileShare
```

Both of these commands will change the label of the device `/dev/sdX` to `FileShare`.



6.3 Discover existing labels

The label for the filesystem can be displayed using the command shown below. In this example, the filesystem label is FileShare.

```
# btrfs filesystem show /dev/sdgt2
Label: FileShare  uuid: 065611ee-4c5d-459b-a9f9-3b59262475e8
      Total devices 1 FS bytes used 9.07GiB
      devid    1 size 36.57GiB used 10.32GiB path /dev/sdgt2

Btrfs v3.16+20140829
```

If the filesystem is an ext2, ext3 or ext4 filesystem, use:

```
# e2label /dev/sdgt2
Fileshare
```

It is possible to discover a partition label using `/etc/fstab` for a system using an ext4 filesystem. The `/etc/fstab` file contains descriptive information about the various filesystems mounted on a SLES system. The first field describes the block device or remote filesystem mounted. The second field is the mount point for the filesystem, or in the case of swap, **none**. The third field describes the type of filesystem being mounted. The fourth field lists the options used in association with the filesystem mounting operation. The fifth field is a number, used by the `dump` command to determine which filesystems need to be dumped. The sixth field is used by the `fsck` command to determine the filesystem check order at boot.

<code>LABEL=/</code>	<code>/</code>	<code>ext4</code>	<code>defaults</code>	<code>1</code>	<code>1</code>
<code>LABEL=/boot</code>	<code>/boot</code>	<code>ext4</code>	<code>defaults</code>	<code>1</code>	<code>2</code>
<code>LABEL=FileShare</code>	<code>/share</code>	<code>ext4</code>	<code>defaults</code>	<code>1</code>	<code>2</code>

The `LABEL=` syntax can be used in a variety of places including mount commands and the GRUB boot loader configuration. Disk labels can also be referenced as a path for applications that do not recognize the `LABEL=` syntax. For example, the volume designated by the label `FileShare` can be accessed at the path `/dev/disk/by-label/FileShare`. Typically for the default `btrfs` root volume, labels are not assigned while an operating system is being installed.

6.4 Swap space

Labeling swap space can be useful, and is only possible when it is created. Swap partitions can be recreated however without data loss, as they do now contain static data and filesystem. For example:

```
# swapoff /dev/sda2
# mkswap -L swapLabel /dev/sda2
# swapon LABEL=swapLabel
```

Like other labels, the new swap label can be added to `/etc/fstab`.

6.5 Universally unique identifiers

UUIDs are an alternative to disk labels that are static and safe for use anywhere. However, their length can make them awkward to work with. The filesystem creation assigns a UUID to the volume as part of the process.

As a best practice, use of UUIDs for `/etc/fstab` entries with SLES 12, including for boot, root and swap.

The UUID of a specific partition can be determined using differing tools based on the type of filesystem.

Discovering the UUID using `btrfs`:

```
# btrfs filesystem show /dev/sdgt2
Label: none  uuid: 065611ee-4c5d-459b-a9f9-3b59262475e8
      Total devices 1 FS bytes used 9.07GiB
      devid    1 size 36.57GiB used 10.32GiB path /dev/sdgt2

Btrfs v3.16+20140829
```

The XFS filesystem has a similar command.

Discovering the UUID using `xfs_admin -u`

```
# xfs_admin -u /dev/sdgt3
UUID = 890481f7-b343-456e-aaff-64d7673b02f6
```

The `tune2fs` command can be used for the `ext2`, `ext3`, and `ext4` filesystems.

The UUID listed using `tune2fs`

```
# tune2fs -l /dev/sdc5
tune2fs 1.42.11 (09-Jul-2014)
Filesystem volume name:   /FileShare2
Last mounted on:          <not available>
Filesystem UUID:          bf9d6e88-bb9b-448c-8341-4b28169aa8fd
<output truncated>
```

Another means for obtaining the UUID of a partition or device is to do a long list of the `/dev/disk/by-uuid` directory.

UUID discovery in the `/dev/disk/by-uuid` directory:

```
# ls -l /dev/disk/by-uuid
total 0
lrwxrwxrwx 1 root root 11 Dec  9 16:15 065611ee-4c5d-459b-a9f9-3b59262475e8 -> ../../sdgt2
lrwxrwxrwx 1 root root 11 Dec  9 16:15 1177c51c-714c-4fa6-994e-1a70489cff41 -> ../../sdgt1
```



```
lrwxrwxrwx 1 root root 11 Dec  9 16:15 890481f7-b343-456e-aaff-
64d7673b02f6 -> ../../sdgt3
lrwxrwxrwx 1 root root 11 Dec 18 13:46 bf9d6e88-bb9b-448c-8341-
4b28169aa8fd -> ../../sdc5
```

The output shows the UUIDs for the various filesystems. Disk UUIDs are useful in the `/etc/fstab` or any other place persistent mappings are required. For example, here is a partial list of filesystems in the `/etc/fstab` file:

```
UUID=1177c51c-714c-4fa6-994e-1a70489cff41 swap swap defaults 0 0
UUID=065611ee-4c5d-459b-a9f9-3b59262475e8 / btrfs defaults 0 0
UUID=065611ee-4c5d-459b-a9f9-3b59262475e8 /boot/grub2/i386-pc btrfs
subvol=@/boot/grub2/i386-pc 0 0
UUID=065611ee-4c5d-459b-a9f9-3b59262475e8 /boot/grub2/x86_64-efi btrfs
subvol=@/boot/grub2/x86_64-efi 0 0
[output truncated]
UUID=890481f7-b343-456e-aaff-64d7673b02f6 /home xfs defaults 1 2
```

In situations where an absolute path is necessary, the links created in the `/dev/disk/by-uuid` directory work in almost all situations.

6.6 GRUB2

The configuration for GRUB2, the second version of the grand unified boot loader, makes use of the UUID by default. For example, it is part of the line from the `/boot/grub2/grub.cfg` file. This script is expecting the UUID as opposed to the LABEL or device file path.

Use of UUIDs in the `/boot/grub2/grub.cfg` file:

```
if [ x$feature_platform_search_hint = xy ]; then
  search --no-floppy --fs-uuid --set=root --hint-bios=hd0,msdos2 \
  --hint-efi=hd0,msdos2 --hint-baremetal=ahci0,msdos2 \
  --hint='hd0,msdos2' 065611ee-4c5d-459b-a9f9-3b59262475e8
else
  search --no-floppy --fs-uuid --set=root \
  65611ee-4c5d-459b-a9f9-3b59262475e8
fi
```

6.7 Unmounting volumes

The Linux operating system stores data regarding each volume, including information about an unmapped volume. The system will retain this information until it is restarted. If another volume using the same target information is presented, the Linux system will reuse the old data from the volume. In other words, it will attempt to connect and unmount the volume. The reuse of old volume data will undoubtedly lead to complications.



Removing the previous, unmapped volume information on the Linux server side is a best practice. Removing the old volume information will leave the data on the volume itself untouched. Only the locally stored information on the Linux system is deleted.

1. Determine the device name of the volume to unmount. For instance `/dev/sdXX`.
2. Unmap the volume in Enterprise Manager.
3. Delete the volume information on the Linux OS. Use the command below, replacing `sdXX` with the correct device file name

```
# echo 1 > /sys/block/sdXX/device/delete
```
4. The echo command, if it works, will provide no output.

Note: Be absolutely certain that the correct device file information is being removed! It is possible to remove the incorrect device file information, making that device unavailable to the Linux system, even if it is being used.

6.8 SCSI UNMAP/TRIM and filesystems

SLES 12 supports SCSI UNMAP/TRIM for filesystems standard to it, including ext4, XFS and btrfs. Files and directories removed from a filesystem result in the release of pages on the array, which are returned to the page pool using the UNMAP/TRIM feature of the SCSI-3 protocol.

Enabling SCSI UNMAP/TRIM functions on the filesystem happens through the mount command. The mount command takes the `-o` switch, which permits inclusion of various parameters in the command line. The `discard` parameter invokes the UNMAP/TRIP feature for ext4, XFS or btrfs filesystems.

```
# mount -o discard /dev/mapper/<volume_name> /<mountpoint_name>
```

The `discard` mount parameter is persistent across reboots by adding it to the `/etc/fstab` file for the filesystems. For example, both the `/home` and `/fileshare2` filesystems have the `discard` parameter in fourth column.

```
# cat /etc/fstab
<output truncated>
UUID=065611ee-4c5d-459b-a9f9-3b59262475e8 /home xfs defaults,discard 1 2
/dev/mapper/FileShare2 /fileshare2 xfs defaults,discard 1 2
```

6.9 SCSI UNMAP/TRIM and LVM configuration

While using LVM with SC Series arrays is not a best practice, there might be a reason for using it. The LVM subsystem can be configured to leverage the SCSI UNMAP/TRIM commands, in turn passing the commands back to the array. When a logical volume with LVM is removed from a volume group, then that space returns to the page pool as freed space on the SC Series array. The LVM setting resides in the `/etc/lvm/lvm.conf` configuration file. The key value pair is `issue_discards`. The default setting is 0 for off. Change the value to 1 to configure LVM to issue discard commands that affect this setting.



7 Useful tools

The base installation for SLES includes tools helpful for storage administration. This section explores some of those tools.

7.1 lsscsi

The `lsscsi` tool obtains storage information from the `/proc`, and `/sys` filesystems and displays the data in human readable format.

Example output from the `lsscsi` command, with disk and volume information:

```
# lsscsi
[4:0:0:0]    disk      COMPELNT Compellent Vol   0605  -
<output truncated>
[4:0:0:10]   disk      COMPELNT Compellent Vol   0605  /dev/sdib
[4:0:1:11]   disk      COMPELNT Compellent Vol   0605  /dev/sdij
[4:0:2:12]   disk      COMPELNT Compellent Vol   0605  /dev/sdir
[4:0:3:13]   disk      COMPELNT Compellent Vol   0605  /dev/sdiz
```

This output shows five drives from the SC Series array, among other truncated output. It also shows that one front-end port is visible but is not presenting a LUN 0, as identified by the lines with a dash (-) in the final column; this is the expected behavior. There are multiple options for `lsscsi` that provide even more detailed information.

Notice in the output above, the four columns on the left within the brackets, delimited by colons [host:channel:target:lun]. The host number represents the local HBA. The host number from the hostX designation is assigned to the HBA port. The Linux operating system mapped the volume to this port. The channel is the SCSI bus address, which will always be a zero. The third number, the target, corresponds to the front-end ports of the SC Series array. Finally, the last number is the LUN number or address for the volume listed.

7.2 lspci

Currently attached PCI devices can be listed using `lspci`. Identifying what the kernel has available to it and what features are associated with the device can be helpful.

Example output from `lspci`:

```
# lspci
<output truncated>
04:00.0 Fibre Channel: QLogic Corp. ISP8324-based 16Gb Fibre Channel to
PCI Express Adapter (rev 02)
04:00.1 Fibre Channel: QLogic Corp. ISP8324-based 16Gb Fibre Channel to
PCI Express Adapter (rev 02)
05:00.0 Fibre Channel: QLogic Corp. ISP8324-based 16Gb Fibre Channel to
PCI Express Adapter (rev 02)
```



05:00.1 Fibre Channel: QLogic Corp. ISP8324-based 16Gb Fibre Channel to
PCI Express Adapter (rev 02)

7.3 scsi_id

The World-Wide Identifier (WWID) of a volume is available in a number of ways. Among them is the `scsi_id`, found in the `/lib/udev` directory. `Scsi_id` is part of the UDEV package. The WWID matches the volume serial number reported Dell Storage Enterprise Manager.

Output from `scsi_id`:

```
# /lib/udev/scsi_id -u -g /dev/sdcs
36000d31000fd100000000000000000ff
```

To see the device ID on the SC Series array, go to the Dell Storage Enterprise Manager, select the appropriate Storage Center, click the **Storage** tab on top. In the menu of folders beneath the tab find the LUN of interest. The last twenty-four numbers in the `scsi_id` output will always match the last twenty-four numbers the Device ID in Enterprise Manager.

General



Serial Number	0000fd10-000000ff
Device ID	6000d31000fd10000000000000000ff
Replay Profile List	Daily
Storage Profile	 Recommended (All Tiers)
Storage Type	 Assigned - Redundant - 2 MB
Compression Enabled	No
Host Cache Enabled	No

Figure 3 From the summary tab of a test LUN in EM

The WWID of the LUN consists of three parts. The first eight characters are the unique ID for the Storage Center. The middle section is the controller number in hexadecimal. Finally, the last numbers are the serial number for the volume. Use these numbers to correlate the SC Series array with the controller number. In an environment with multiple SC Series arrays or other storage systems, it is entirely likely another storage system will have the same volume serial number. When taking all the three parts together, however, the WWID will be unique.

It is possible the two serial numbers do not correlate. If the volume had been copy migrated, A new serial number is assigned to the volume on the SC Series array. The previous WWID is presented to the server, preventing service disruption.

7.4 /proc/scsi/scsi

As mentioned above, the `lsscsi` tool parses information from within the `/proc` and `/sys` pseudo-files systems. The contents of `/proc/scsi/scsi` provide information about LUNs and targets. Correlating the output with a specific device is difficult.

```
# cat /proc/scsi/scsi
<output truncated>
Host: scsi0 Channel: 00 Id: 02 Lun: 10
  Vendor: COMPELNT Model: Compellent Vol   Rev: 0605
  Type:   Direct-Access                    ANSI  SCSI revision: 05
<output truncated>
```

7.5 /proc/mounts

Within the pseudo-filesystem `/proc` is the symbolic link `mounts`, which points at a construct within the kernel. The referenced construct contains a great deal of information about mounts. The output is similar to that found in the `/etc/mtab` file, but is current.

```
# cat mounts
rootfs / rootfs rw 0 0
sysfs /sys sysfs rw,nosuid,nodev,noexec,relatime 0 0
proc /proc proc rw,nosuid,nodev,noexec,relatime 0 0
devtmpfs /dev devtmpfs rw,nosuid,size=12296164k,nr_inodes=3074041,mode=755 0 0
securityfs /sys/kernel/security securityfs rw,nosuid,nodev,noexec,relatime 0 0
<output truncated>
/dev/mapper/36000d31000fd10000000000000000079_p1 /FileShare xfs ↻
rw,relatime,attr2,inode64,noquota 0 0
```

7.6 /sys/block/sdX/queue

Each Linux disk device, distinguished by `sdX`, has a set of separate parameters and settings, found in the `/sys` pseudo-filesystem in the `/sys/block/sdX/queue`.

```
# ls /sys/block/sdcs/queue
add_random                max_hw_sectors_kb        optimal_io_size
discard_granularity        max_integrity_segments   physical_block_size
discard_max_bytes          max_sectors_kb            read_ahead_kb
discard_zeroes_data        max_segments              rotational
hw_sector_size             max_segment_size          rq_affinity
iosched                    minimum_io_size            scheduler
iostats                    nomerges                   write_same_max_bytes
logical_block_size         nr_requests
```



7.7 /sys/class/fc_host/hostX

Fibre Channel class HBA has a corresponding entry within the /sys pseudo-filesystem in /sys/class/fc_host/hostX. It is an incremental value of the HBAs available to the kernel.

The HBA World-Wide Name (WWN) can be displayed from /sys/class/fc_host/hostX

```
# cat /sys/class/fc_host/host0/port_name
0x21000024ff6d48d0
```

7.8 dmesg

The driver message command (dmesg) is available in Linux that prints the kernel message buffer. The output of this program is useful for finding what device name the kernel assigned to a recently discovered volume.

Below is a sample of output from the dmesg command showing a LUN with a total of about 250 GB. This output is also found in the /var/log/messages file after a reboot. The output could be found if the LUN were newly attached to the host, using a rescan of the SCSI bus.

```
SCSI device sdc: 513204680 512-byte hdwr sectors (250587 MB)
sdc: Write Protect is off
sdc: Mode Sense: 87 00 00 00
SCSI device sdc: drive cache: write through
SCSI device sdc: 513204680 512-byte hdwr sectors (250587 MB)
sdc: Write Protect is off
sdc: Mode Sense: 87 00 00 00
SCSI device sdc: drive cache: write through
sdc: unknown partition table
sd 4:0:3:9: Attached scsi disk sdc
sd 4:0:3:9: Attached scsi generic sg13 type 0
```



A Additional resources

This appendix provides contact information and other documentation available to assist with administering a SLES 12 system or systems in conjunction with SC Series storage.

A.1 Technical support and resources

Dell.com/support is focused on meeting customer needs with proven services and support.

For additional support information on specific array models, see the following table.

Dell Storage	online support	Email	phone support (US only)
SC Series and Compellent	https://portal.compellent.com	support@compellent.com	866-EZ-STORE (866-397-8673)
SCv Series	http://www.dell.com/support	Specific to service tag	800-945-3355
PS Series (EqualLogic)	http://eqlsupport.dell.com	eqlx-customer-service@dell.com	800-945-3355

[Dell TechCenter](#) is an online technical community where IT professionals have access to numerous resources for Dell software, hardware and services.

[Storage Solutions Technical Documents](#) on Dell TechCenter provide expertise that helps to ensure customer success on Dell Storage platforms.

A.2 Related documentation

For SLES 12 documentation from the vendor, please see:

Item	Resource
Admin Guide	SLES 12 Administration Guide
Deployment Guide	SLES 12 Deployment Guide
Storage Admin Guide	SLES 12 Storage Administration
System Tuning	SLES 12 System Analysis and Tuning

Some pertinent Dell Storage information that might be useful, please see:

Dell Resource
Creating/Mapping a Volume in Enterprise Manager



Dell Enterprise Manager 2015 R3 Administration Guide
Best practices for sharing an iSCSI SAN Infrastructure with Dell PS Series and Dell SC Series Storage using Linux Hosts
Dell Storage Center with Red Hat Enterprise Linux 7x Best Practices

