Dell Storage PS Series Arrays with SUSE Linux Enterprise Server 12 SP1

Dell Storage Engineering
October 2016

A Dell Best Practices Guide

Revisions

Date	Revision	Description
October 2016	1.0	Release based on SLES 12 SP1

Acknowledgements

Author: Steven Lemons

Table of contents

Executive summary

SUSE® Linux® Enterprise Server (SLES) is a versatile server operating system for deploying highly available 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 Dell™ PS Series storage. These best practices include guidelines for configuring volume discovery, multipath, and filesystem management.

This paper covers iSCSI technology for front-end connectivity to the PS Series array. This paper covers two methods in configuring iSCSI connectivity to the PS Series array: the Dell EqualLogic Host Integration Tools for Linux (HIT/Linux), and native Linux iSCSI and multipathing configuration.

1 SLES 12 SP1 and PS Series overview

PS Series arrays provide Linux-compatible and SCSI-3 compliant disk volumes that remove the complexity of allocating, administering, using, and protecting mission-critical data. PS Series arrays provide multiple RAID levels (RAID 50, RAID 10, RAID 5, RAID 6, or RAID 6 Accelerated) and reliability at the storage layer so that presented volumes to the host do not need further RAID management by the Linux operating system. 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 PS Series volumes.

1.1 New to SLES 12 SP1

SLES 12 SP1 offers many innovative changes to the operating system including:

- An updated installer introduces the ability to register a system and receive all available maintenance updates during the installation process.
- Improved management capabilities with full system rollback utilize 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 SP1 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 Dell Storage arrays.
- iSCSI and FC targets are implemented from the kernel instead of the user space.
- Core technology improvements such as systemd (which replaces the System-V-based init process) and wicked (a more dynamic and modern network configuration infrastructure).

Details about SLES are located at SUSE Linux Enterprise Server 12 SP1 documentation.

1.2 Special filesystems

SLES 12 SP1 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/host*x*, where *x* is the HBA number in the server. Access to these filesystems is useful when administering SLES 12 SP1 systems in conjunction with PS Series storage.

1.3 iSCSI

iSCSI technology is a standard technology used for block storage. SLES 12 SP1 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 Dell PS Series Configuration Guide.

iSCSI protocol requires a network port to communicate with the Dell PS Series arrays. A dedicated network port or a dedicated VLAN for iSCSI traffic is critical. The type of data determines the network 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. A complete listing of all Ethernet switches and host CNAs validated with PS Series arrays can be found in the *Dell Storage Compatibility Matrix*.

2 Host configuration with HIT/Linux

The following section will focus on configuring a single SLES 12 SP1 host for access to a Dell PS Series array. A pair of 10GB Ethernet interfaces are configured to communicate with the PS Series array over iSCSI while a single 1GB interface is used for data, non-SAN traffic.

The Dell EqualLogic Host Integration Tools for Linux (HIT/Linux) provide a collection of applications and utilities to simplify the configuration and administration of Dell Storage PS Series arrays. HIT/Linux is packaged as an ISO image with the file name **equallogic-host-tools-*version*.iso**. This image contains all necessary user- and kernel-mode RPM files and an installation script for specific Linux distributions.

2.1 Download Dell EqualLogic Host Integration Tools for Linux

A Dell EqualLogic Customer Support account is required to obtain the installation kit from the EqualLogic customer support web site. Set up an account at https://eqlsupport.dell.com.

Use the support account to obtain the installation kit as follows:

1. Log into your account at https://eqlsupport.dell.com.
2. Click **Downloads** in the navigation bar and select **Host Integration Tools for Linux**.
3. Click the latest revision of the toolkit to display the web page for that revision.
4. Click the download link for the current software and accept the terms and conditions of the Dell End User License Agreement (EULA).
5. Save the ISO installation image and public GPG key to a temporary, local location. The installation requires a public key to authorize the RPM signature and run the installation.
6. Import the downloaded GPG key using the following command:

   ```
   # rpm --import file-name
   ```

   Example:

   ```
   SLES12SP1:~ # rpm --import RPM-GPG-KEY-DELLEQL
   ```

Note: The *Dell EqualLogic Host Integration Tools for Linux Installation and User's Guide* is also included in this package. It includes a complete reference about HIT/Linux.

2.2 Before installing the HIT/Linux kit

Once the HIT/Linux kit has been downloaded and the GPG key installed, there are a few steps that need to be completed before installing the actual HIT/Linux kit software.

1. Start the appropriate iSCSI unit files on the SLES 12 SP1 server. Persistent availability requires enabling the unit files through the `enable` systemctl flag.

   ```
   SLES12SP1:~ # systemctl start {iscsi.service,iscsid.socket}
   SLES12SP1:~ # systemctl enable {iscsi.service,iscsid.socket}
   ```

2. Confirm that both the `iscsi.service` and `iscsid.socket` services are active (highlighted in yellow) and set as `enabled` (highlighted in green) for persistent availability across reboots.

```
SLES12SP1:~ # systemctl status {iscsi.service,iscsid.socket} | egrep -B1
"active"
   Loaded: loaded (/usr/lib/systemd/system/iscsi.service; enabled)
   Active: active (exited) since Tue 2016-09-13 09:29:25 CDT; 4h 27min ago
--
   Loaded: loaded (/usr/lib/systemd/system/iscsid.socket; enabled)
   Active: active (running) since Tue 2016-09-13 09:29:20 CDT; 4h 27min
ago
```

3. Provide additional configuration details for the two 10GB Ethernet interfaces to ensure optimal performance with the PS Series array. Each 10GB interface should be configured with an IPv4 address within the same network as the PS Series array. For the purposes of this paper, the interfaces described as follows are configured with 10.10.10.112 and 10.10.10.113, respectively. This will allow these 10GB interfaces to discover volumes from the PS Series array configured at 10.10.10.10.

   a. Set the frame size (MTU) to Jumbo frames (9000) for each 10GB interface on the host that will be connected to the PS Series array. Edit the **/etc/sysconfig/network/ifcfg-nic** file for each respective port of the 10GB interface. For the purposes of this paper, the ports of the 10GB interface are p3p1 and p3p2. The following command reference will quote out any existing MTU size in the p3p1 and p3p2 file(s) and replace with the proper Jumbo frames value.

```
SLES12SP1:~ # sed -i -e 's/^MTU=/#&/' -e '$aMTU='"\'"'9000'"\'"''
/etc/sysconfig/network/ifcfg-p3p*
```

Note: The MTU (Jumbo Frame) size must be configured with the same value on each port in the Ethernet switch configuration, any additional NIC on the host that will be connected to the SAN as well as the Dell Storage PS Series array. If the MTU size in all three components (storage, switch and host) do not match, anomalous behavior can occur within the SAN.

   b. Set the flow control to RX=On and TX=On for each port of the 10GB interface that will be connected to the PS Series array. Use the command **`ethtool -a`** to check these values. Use the command **`ethtool -A`** if these values need to be changed to the value of `on`.

```
SLES12SP1:~ # for x in {p3p1,p3p2}; do ethtool -a $x; done
Pause parameters for p3p1:
Autonegotiate:  off
RX:             on
TX:             on

Pause parameters for p3p2:
Autonegotiate:  off
RX:             on
TX:             on
```

4. Once all of the networking configuration parameters have been defined, restart the networking service for the new values to take effect. With SLES 12, a new network management tool for Linux was introduced named Wicked. Use the **wicked** command to dynamically change the configuration of these interfaces as well as check the status of each interface confirming the new configuration values.

```
SLES12SP1:~ # systemctl restart wicked
SLES12SP1:~ # systemctl status wicked
wicked.service - wicked managed network interfaces
   Loaded: loaded (/usr/lib/systemd/system/wicked.service; enabled)
   Active: active (exited) since Mon 2016-09-19 10:14:46 CDT; 6s ago
  Process: 26129 ExecStop=/usr/sbin/wicked --systemd ifdown all
(code=exited, status=0/SUCCESS)
  Process: 26653 ExecStart=/usr/sbin/wicked --systemd ifup all
(code=exited, status=0/SUCCESS)
 Main PID: 26653 (code=exited, status=0/SUCCESS)

Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: lo             up
Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: em1            up
Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: p3p1           up
Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: p3p2           up
Sep 19 10:14:46 SLES12SP1-Intel systemd[1]: Started wicked managed network
interfaces.
SLES12SP1:~ # wicked ifstatus {p3p1,p3p2}
p3p1            up
      link:     #4, state up, mtu 9000
      type:     ethernet, hwaddr a0:36:9f:03:a4:98
      config:   compat:suse:/etc/sysconfig/network/ifcfg-p3p1
      leases:   ipv4 static granted
      addr:     ipv4 10.10.10.112/24 [static]

p3p2            up
      link:     #7, state up, mtu 9000
      type:     ethernet, hwaddr a0:36:9f:03:a4:9a
      config:   compat:suse:/etc/sysconfig/network/ifcfg-p3p2
      leases:   ipv4 static granted
      addr:     ipv4 10.10.10.113/24 [static]
```

5. Create a backup of the system with the **snapper** command prior to installing HIT/Linux so you can revert any or all changes if necessary.

```
SLES12SP1:~ # snapper create -d "pre HIT/Linux install"
```

2.3 Install and configure the HIT/Linux kit

Perform the following steps to install and configure the HIT/Linux kit:

1. Mount the HIT/Linux kit ISO image using one of the following methods:
   - Use the Dell PowerEdge iDRAC Virtual Media feature to map the image to the server.
   - Burn the image to a CD and insert it in the server optical drive.
   - Copy the image to a USB drive and insert it into one of the available server USB ports.
   - Mount the ISO image file (shown in this example).

   Use the following commands, executed from within the directory containing the ISO image file, to mount the ISO image file:

   ```
   SLES12SP1:~ # mkdir -p /media/iso
   SLES12SP1:~ # mount -o loop equallogic-host-tools-version.iso /media/iso
   ```

   After mounting the image, the following files and directories are available.

   ```
   SLES12SP1:~ # ls /media/iso
   EULA install LICENSES packages README support welcome-to-HIT.pdf
   ```

Note: To simplify this guide, it is assumed that the HIT/Linux ISO image content is in the /media/iso/ directory. Replace this directory in the following commands if another directory is used (such as /media/CDROM/).

2. Run the installation script by entering the following command.

   ```
   SLES12SP1:~ # /media/iso/install
   ```

3. The Dell End User License Agreement is displayed; type **`Accept`** to continue.
4. Type **`y`** to install the equallogic-host-tools and dell-dm-switch-kmp-default packages.
5. After the **`Install succeeded`** message appears, the installation procedure invokes the script **eqlconfig** to perform the initial configuration.
6. When prompted, Would you like ehcmd to actively manage MPIO and iSCSI sessions (Yes/No) [Yes]?**,** press [**Enter**].
7. When prompted, `Choose address protocol (IPv4/IPv6) [IPv4]:`, press [**Enter**] to choose IPv4, or type **`IPv6`** and press [**Enter**].

Note: While Dell Storage PS Series arrays support IPv6, the following example shows IPv4 configuration.

8. When prompted with the subnets discovered for ehcmd to actively manage MPIO, select the appropriate value to match your SAN's subnet. For the purposes of this paper, the subnet is 10.10.10.0/24.

```
Found the following subnets for MPIO:

        1.) 10.211.15.192/26
        2.) 10.10.10.0/24
        3.) Choose individual NICs

Enter a comma-separated list of subnets that you want to use for MPIO,
e.g., 1, 2 or select individual NICs (1, 2, 3) [2]: 2
```

9. When prompted for the document directory where ASM (Auto-Snapshot Manager) stores Smart Copy backup documents, press [**Enter**] to confirm the default directory or type a custom location.

Note: The ASM feature of HIT/Linux is installed by default during HIT/Linux installation.

10. If credentials are required to access the group array found at subnet 10.10.10.0/24, enter **`Yes`** at next prompt otherwise enter **`No`** to provide these at a later time during setup.
11. Next, the installation procedure invokes the script **eqltune** to verify that the configurable parameters conform to the Dell recommended values. After **eqltune** identifies critical system issues in a summary table, it repairs each category of critical issue. Press [**Enter**] to fix the detected errors.
12. After all of the settings have been optimized, the message, `Installation complete` is displayed.
13. Run the following to take advantage of BASH completion for most Dell EqualLogic tools.

```
SLES12SP1:~ # . /etc/bash_completion.d/equallogic
```

14. Once installation configuration has completed, use the ehcmcli utility to confirm adapter configuration:

```
SLES12SP1:~ # /usr/sbin/ehcmcli status
Generating diagnostic data, please wait...

============================================================
Adapter List
============================================================
  Name: p3p1
    IP Address: 10.10.10.112
    HW addr: A0:36:9F:03:A4:98

  Name: p3p2
    IP Address: 10.10.10.113
    HW addr: A0:36:9F:03:A4:9A

============================================================
Volume list
============================================================
============================================================
```

```
Summary
============================================================
  Adapters:         2
  Managed Volumes:  0
  iSCSI Sessions:   0
  Errors:           0
  Warnings:         0
  Suggestions:      0
```

The **ehcmcli** command (with status flag) shows the current status of the Dell EqualLogic Host Connection Manager daemon (ehcmd) as well as useful information such as the adapter list, volume list, number of iSCSI sessions, errors, and warnings.

2.4 Connect to the PS Series group

Once the HIT/Linux installation is complete, the rswcli (Dell EqualLogic Remote Setup Wizard CLI) and ehcmcli (Dell EqualLogic Host Connection Manager CLI) utilities are used to configure host access to the PS Series arrays.

To configure host access to a PS Series group, use the `rswcli` utility as such:

```
# rswcli –a –gn=group name –gip=group IP address
```

Example:

```
SLES12SP1:~ # /usr/sbin/rswcli –a –gn=Coexistence-10GbE –gip=10.10.10.10
```

To verify the group is accessible from the host, use the `rswcli` utility as such:

```
# rswcli –l
```

Example:

```
SLES12SP1:~ # /usr/sbin/rswcli -l
Processing list-group command...

Groups accessible from this computer:

Group Name: Coexistence-10GbE
Group IP Address: 10.10.10.10


The list-group command succeeded.
```

2.5 Edit volume access permissions for Linux host

Three different types of access can be established to a Dell Storage PS Series SAN volume: using CHAP account information, using iSCSI initiator name, or using the initiator IP address. Each access type can be used or a combination of these methods (for example, CHAP and the iSCSI initiator name) for host access.

On the Linux host that will access a volume on the storage array, perform the following steps:

1. Review the iSCSI initiator name stored in the /etc/iscsi/initiatorname.iscsi file as shown in the following example.

```
SLES12SP1:~ # egrep iqn /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.1996-04.de.suse:01:2bb83dfd9278
```

2. If CHAP authentication is used to access the volume, edit the CHAP Setting section of the /etc/iscsi/iscsid.conf file.

```
SLES12SP1:~ # vi /etc/iscsi/iscsid.conf
```

3. If the initiator IP address is used, use the ehcmcli utility to get the **Adapter List** information.

```
SLES12SP1:~ # /usr/sbin/ehcmcli status
```

Perform the following steps from the Group Manager Web application.

4. If creating a **new volume**, the wizard requests access type information. Enter the CHAP authentication info, iSCSI initiator name, or initiator IP addresses to limit iSCSI access to the new volume. This example uses the iSCSI initiator name.
5. Select whether or not to allow simultaneous connections from initiators with different IQN names (see the following screen shot).

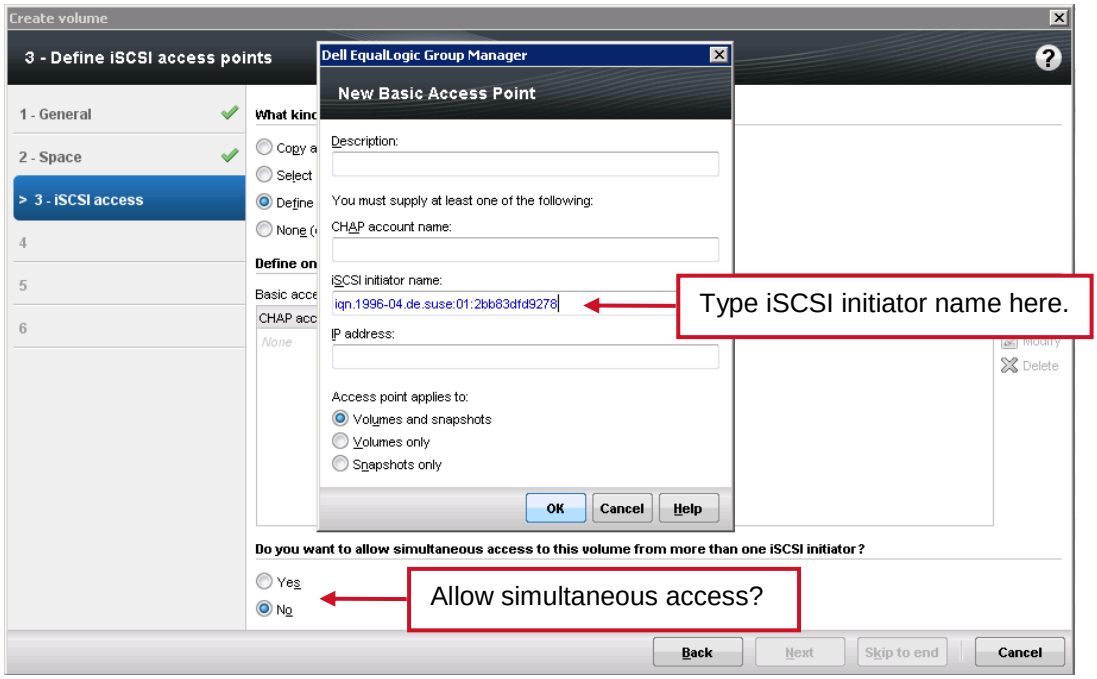

Figure 1 New volume options

6. If the volume was already created, click the volume **Access** tab and add a new **basic access point.**

Tip: Access points are used to add or change initiator names, CHAP, and IP address access for existing volumes.

2.6 Access volumes using Multipath I/O

Note: To access volumes using Multipath I/O (MPIO), you must discover targets (volumes) and then log into at least one iSCSI session for each volume. By default, the EqualLogic Host Connection Manager (ehcmd) service uses the software iSCSI initiator to connect to volumes.

To log into an MPIO volume, use the `ehcmcli` utility as such:

```
# ehcmcli login --target target_name --portal portal
```

- *`target_name`* indicates the full iSCSI-qualified name (IQN) or a volume name for the PS Series group target node
- *`portal`* indicates the iSCSI portal (group IP address).

If the entire target name is unknown, check the volume **Connections** tab in the EqualLogic Group Manager. The following EqualLogic Group Manager screenshot displays the iSCSI target used for this example.

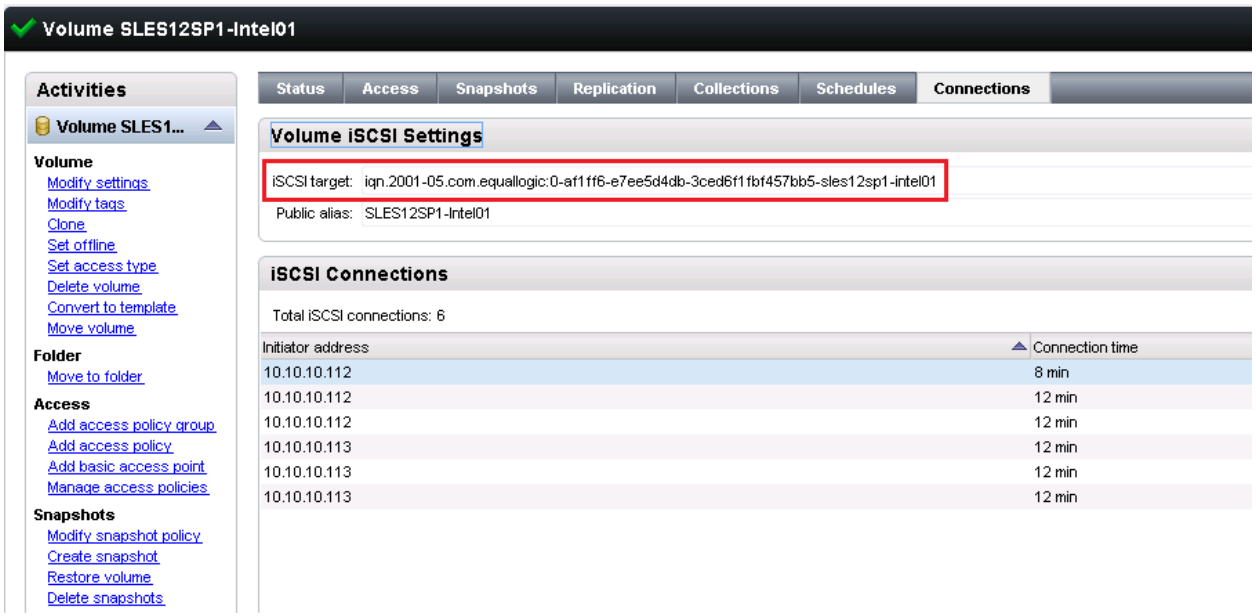

Figure 2 PS Volume iSCSI Settings

Using the verified target name, the following example shows the **Login succeeded** (highlighted in yellow) message and the **Device to mount** (highlighted in green) information.

```
SLES12SP1:~ # /usr/sbin/ehcmcli login --target iqn.2001-05.com.equallogic:0-
af1ff6-e7ee5d4db-3ced6f1fbf457bb5-sles12sp1-intel01 --portal 10.10.10.10
Login succeeded.  Device to mount:
/dev/eql/sles12sp1-intel01
```

Note: After a single session is created, the EqualLogic Host Connection Manager (`ehcmd`) will analyze the configuration and create additional iSCSI sessions as appropriate. When complete, an iSCSI session is initialized for the appropriate PS Series volume, `/dev/eql/`*`volume-name`*, and the volume is available for use. All further iSCSI management by `ehcmd` is transparent to any application using `/dev/eql/`*`volume-name`*.

2.7 Mount an MPIO volume

The following steps cover the process of mounting a PS volume after creating an XFS file-system on the presented device:

1. Confirm the name of the device to mount by running:

   ```
   SLES12SP1:~ # /usr/sbin/ehcmcli status
   ```

2. Look for the `Device to mount` line.

   ```
   Device to mount: /dev/eql/sles12sp1-intel01
   ```

3. Create a mount point.

   ```
   SLES12SP1:~ # mkdir -p /media/PS_Vol_01
   ```

4. Create an XFS filesystem on the presented device, mount the device and then add an entry within `/etc/fstab` for persistent access across reboots. Dell recommends to use the entire drive without partition for all non-boot drives.

   ```
   SLES12SP1:~ # mkfs.xfs /dev/eql/sles12sp1-intel01
   meta-data=/dev/eql/sles12sp1-intel01  isize=256    agcount=4,
   agsize=1311360 blks
            =                       sectsz=512   attr=2, projid32bit=1
            =                       crc=0        finobt=0
   data     =                       bsize=4096   blocks=5245440, imaxpct=25
            =                       sunit=0      swidth=0 blks
   naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
   log      =internal log           bsize=4096   blocks=2561, version=2
            =                       sectsz=512   sunit=0 blks, lazy-count=1
   realtime =none                   extsz=4096   blocks=0, rtextents=0

   SLES12SP1:~ # mount /dev/eql/sles12sp1-intel01 /media/PS_Vol_01/
   ```

```
SLES12SP1:~ # df -hT | awk 'NR==1 || /PS_Vol_01/'
Filesystem            Type      Size  Used Avail Use% Mounted on
/dev/eql/sles12sp1-intel01 xfs        20G   33M   20G   1%
/media/PS_Vol_01

SLES12SP1:~ # xfs_admin -u /dev/eql/sles12sp1-intel01
UUID = 15087ae8-b9e7-4ed5-8c64-1dace97be9c5
```

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

   ```
   UUID=15087ae8-b9e7-4ed5-8c64-1dace97be9c5 /media/PS_Vol_01 xfs
   defaults,_netdev 1 2
   ```

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

Note: The `dump` option of `1` and `pass` option of `2` in the prior `/etc/fstab` example, can be changed to reflect the storage policies of data integrity within the applicable SAN environment.

3 Host configuration with native iSCSI and multipath

This section focuses on configuring a single SLES 12 SP1 host for access to a Dell PS Series array. A pair of 10GB Ethernet interfaces are configured to communicate with the PS Series array over iSCSI while a single 1GB interface is used for data, non-SAN traffic.

While the recommended Dell best practice is to utilize the HIT/Linux kit for configuration, performance tuning, and optimization, it is possible to use native Linux iSCSI and multipath for connectivity to the PS Series array. The following subsection highlights those settings within the Linux host that must be manually configured when not using the HIT/Linux kit for optimal SAN performance.

3.1 Software requirements

Three software packages are required for SLES 12 SP1 iSCSI connectivity to the PS Series array. The following example shows the installation instructions for these three software packages.

```
SLES12SP1:~ # zypper install iscsiuio multipath-tools device-mapper*
```

- `iscsiuio` – Linux Broadcom NetXtremem II iscsi server
- `multipath-tools` – Tools to manage multipathed devices with the device-mapper
- `device-mapper` – Device mapper tools (the `mapper*` allows both 64 bit and 32 bit device-mapper packages to be installed)

The appropriate iSCSI unit files on the SLES 12 SP1 server need to be started after installation. Persistent availability requires enabling the unit files through the `enable` systemctl flag.

```
SLES12SP1:~ # systemctl start {iscsi.service,iscsid.socket}
SLES12SP1:~ # systemctl enable {iscsi.service,iscsid.socket}
```

Now confirm that both the `iscsi.service` and `iscsid.socket` services are active (highlighted in yellow) and set as `enabled` (highlighted in green) for persistent availability across reboots.

```
SLES12SP1:~ # systemctl status {iscsi.service,iscsid.socket} | egrep -B1
"active"
   Loaded: loaded (/usr/lib/systemd/system/iscsi.service; enabled)
   Active: active (exited) since Tue 2016-09-13 09:29:25 CDT; 4h 27min ago
--
   Loaded: loaded (/usr/lib/systemd/system/iscsid.socket; enabled)
   Active: active (running) since Tue 2016-09-13 09:29:20 CDT; 4h 27min ago
```

Enabling and starting the other services will be addressed later in this section.

3.2 MTU and Jumbo frames

The industry standard Ethernet frame size is 1500 bytes, however, to move more data more efficiently through your SAN, a larger frame size of 9000 bytes is recommended. This 9000-byte Ethernet frame size is referred to as Jumbo frames. This MTU sizing must be configured on and must match between the PS Series array eth ports, each port in the Ethernet switch configuration, and any NIC/CNA port at the host.

Frame size (MTU) needs to be set to Jumbo frames (9000) for each 10GB interface on the host that will be connected to the PS Series array. Edit the **/etc/sysconfig/network/ifcfg-nic** file for each respective port of the 10GB interface. For the purposes of this paper, the ports of the 10GB interface are p3p1 and p3p2. The following command reference will quote out any existing MTU size in the p3p1 & p3p2 file(s) and replace with the proper Jumbo frames value.

```
SLES12SP1:~ # sed -i -e 's/^MTU=/#&/' -e '$aMTU='"\'"'9000'"\'"'' 
/etc/sysconfig/network/ifcfg-p3p*
```

Note: The MTU (Jumbo frame) size must be configured with the same value on each port in the Ethernet switch configuration, any additional NIC on the host that will be connected to the SAN as well as the Dell Storage PS Series array. If the MTU size in all three components (storage, switch, and host) do not match, anomalous behavior can occur within the SAN.

Flow control must be set to RX=On and TX=On for each port of the 10GB interface that will be connected to the PS Series array. Use the command **`ethtool -a`** to check these values. Use the command **`ethtool -A`** if these values need to be changed to the value of `on`.

```
SLES12SP1:~ # for x in {p3p1,p3p2}; do ethtool -a $x; done
Pause parameters for p3p1:
Autonegotiate:  off
RX:             on
TX:             on

Pause parameters for p3p2:
Autonegotiate:  off
RX:             on
TX:             on
```

Once all of the networking configuration parameters have been defined, the networking service must be restarted for the new values to take effect. With SLES 12, a new network management tool for Linux was introduced named Wicked. Use the **`wicked`** command to dynamically change the configuration of these interfaces as well as check the status of each interface confirming the new configuration values.

```
SLES12SP1:~ # systemctl restart wicked
SLES12SP1:~ # systemctl status wicked
wicked.service - wicked managed network interfaces
   Loaded: loaded (/usr/lib/systemd/system/wicked.service; enabled)
   Active: active (exited) since Mon 2016-09-19 10:14:46 CDT; 6s ago
  Process: 26129 ExecStop=/usr/sbin/wicked --systemd ifdown all (code=exited,
status=0/SUCCESS)
```

```
  Process: 26653 ExecStart=/usr/sbin/wicked --systemd ifup all (code=exited,
status=0/SUCCESS)
 Main PID: 26653 (code=exited, status=0/SUCCESS)

Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: lo              up
Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: em1             up
Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: p3p1            up
Sep 19 10:14:46 SLES12SP1-Intel wicked[26653]: p3p2            up
Sep 19 10:14:46 SLES12SP1-Intel systemd[1]: Started wicked managed network
interfaces.
SLES12SP1:~ # wicked ifstatus {p3p1,p3p2}
p3p1            up
      link:     #4, state up, mtu 9000
      type:     ethernet, hwaddr a0:36:9f:03:a4:98
      config:   compat:suse:/etc/sysconfig/network/ifcfg-p3p1
      leases:   ipv4 static granted
      addr:     ipv4 10.10.10.112/24 [static]

p3p2            up
      link:     #7, state up, mtu 9000
      type:     ethernet, hwaddr a0:36:9f:03:a4:9a
      config:   compat:suse:/etc/sysconfig/network/ifcfg-p3p2
      leases:   ipv4 static granted
      addr:     ipv4 10.10.10.113/24 [static]
```

3.3 Kernel configuration parameters

The following Linux kernel parameters need to be configured to provide a starting point from which to achieve optimal performance for the SAN environment deployed within. While these settings are automatically added when using the **eqltune** utility included with the HIT/Linux kit, the following values can be manually added to the Linux host through editing the ***/etc/sysctl.conf*** file and then reading those values into the running kernel through the **sysctl** command.

1. If using multiple iSCSI connections from the host to the PS Series array, the default Linux ARP behavior for these iSCSI connectors must be changed to prevent ARP resets (RST) from the iSCSI initiators to the target. Changing this ARP reset behavior allows more than a single iSCSI interface to receive/serve traffic. Add the following configuration variables to the ***/etc/sysctl.conf*** file for each iSCSI interface accessing the PS Series array (for this example, interface p3p1 and p3p2 are used).

   ```
   net/ipv4/conf/p3p1/arp_ignore = 1
   net/ipv4/conf/p3p1/arp_announce = 2
   net/piv4/conf/p3p1/rp_filter = 2
   net/ipv4/conf/p3p2/arp_ignore = 1
   net/ipv4/conf/p3p2/arp_announce = 2
   net/ipv4/conf/p3p2/rp_filter = 2
   ```

2. The network kernel settings need to be configured to allow for a larger receive/send socket buffer as well as increase the Linux autotuning TCP buffer limit to 16MB. Add these values to a new file titled **/etc/sysctl.d/dell_ps.conf**.

```
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 8192 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.wmem_default = 262144
net.core.rmem_default = 262144
```

3. Once steps 1 and 2 have been completed, read these new values into the running kernel using the **`sysctl -p`** command.

```
SLES12SP1:~ # sysctl -p {/etc/sysctl.conf,/etc/sysctl.d/dell_ps.conf}
net.ipv4.conf.p3p1.arp_ignore = 1
net.ipv4.conf.p3p1.arp_announce = 2
net.ipv4.conf.p3p2.arp_ignore = 1
net.ipv4.conf.p3p2.arp_announce = 2
net.ipv4.conf.p3p1.rp_filter = 2
net.ipv4.conf.p3p2.rp_filter = 2
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_rmem = 8192 87380 16777216
net.ipv4.tcp_wmem = 4096 65536 16777216
net.core.wmem_default = 262144
net.core.rmem_default = 262144
```

3.4 iSCSI daemon parameters

The following iSCSI daemon parameters need to be configured to provide a starting point from which to achieve optimal performance for the SAN environment deployed within. While these settings are automatically changed from their default values when using the HIT/Linux kit, the following values can be manually added to the Linux host through editing the **/etc/iscsi/iscsid.conf** file and then restarting the iSCSI daemon.

```
node.session.cmds_max = 1024
node.session.queue_depth = 128
node.session.iscsi.FastAbort = No
```

Next, restart the iSCSI services with the following command.

```
SLES12SP1:~ # systemctl restart {iscsid.service,iscsid.socket}
```

Note: If there are active iSCSI connections on the host, prior to restarting the iSCSI services for the new configuration values to take effect, those iSCSI targets must be logged out of with the **`iscsiadm -n node -u`** command.

3.5 Add multiple iSCSI interfaces

Each iSCSI interface that accesses the PS Series array must be added to the iscsiadm interface database and it must have a defined alias name for use when binding with a software iSCSI session so the iSCSI daemon is aware of each interface and can be available for multipath use. In the following example, two iSCSI interfaces (p3p1 and p3p2) are added to the iscsiadm interface database and their network interface names are updated.

1. Check the status of the iscsiadm interface database to confirm these two interfaces (p3p1 and p3p2) are not configured therein:

```
SLES12SP1:~ # iscsiadm -m iface -o show
default tcp,<empty>,<empty>,<empty>,<empty>
iser iser,<empty>,<empty>,<empty>,<empty>
```

2. Add both interfaces (p3p1 and p3p2) to the iscsiadm interface database.

```
SLES12SP1:~ # iscsiadm -m iface -I p3p1 --op=new
New interface p3p1 added
SLES12SP1:~ # iscsiadm -m iface -I p3p2 --op=new
New interface p3p2 added
```

3. Update both interfaces to have a defined alias name.

```
SLES12SP1:~ # iscsiadm -m iface -I p3p1 --op=update -n iface.net_ifacename
-v p3p1
p3p1 updated.
SLES12SP1:~ # iscsiadm -m iface -I p3p2 --op=update -n iface.net_ifacename
-v p3p2
p3p2 updated.
```

Now, these two iSCSI interfaces are validated as present within the iscsiadm interface database (highlighted in yellow), and their network interface alias names (highlighted in green) are verified as updated through the following command, which displays the list of configured iSCSI interfaces from the **/etc/iscsi/ifaces** directory.

```
SLES12SP1:~ # iscsiadm -m iface -o show | egrep "p3p1|p3p2"
p3p1 tcp,<empty>,<empty>,p3p1,<empty>
p3p2 tcp,<empty>,<empty>,p3p2,<empty>
```

3.6 Discover iSCSI PS Series targets

The iSCSI daemon can now discover PS Series volumes, referred to as targets when using the **iscsiadm** command, using the two configured iSCSI interfaces. In the following example, the **iscsiadm** command queries the PS Series group located at 10.10.10.10 using both iSCSI interfaces (p3p1 and p3p2), reporting back the available volumes for the host.

```
SLES12SP1:~ # iscsiadm -m discoverydb -t sendtargets -p 10.10.10.10 -o new -D
10.10.10.10:3260,1 iqn.2001-05.com.equallogic:0-af1ff6-04ae5d4db-
838d6f20f2c57bf4-sles12sp1-intel02
10.10.10.10:3260,1 iqn.2001-05.com.equallogic:0-af1ff6-04ae5d4db-
838d6f20f2c57bf4-sles12sp1-intel02
```

One volume was discovered in this example (sles12sp1-intel01) and each iSCSI interface reported back seeing each volume for a total of two targets displayed.

3.7 Access iSCSI PS Series volumes

To access the PS Series volumes, the host must log into previously discovered targets with the **iscsiadm** command. All of the discovered targets within the iscsiadm discovery database can be logged into at the same time with the **iscsiadm -m node -l** command. In the following example, the host logs into all targets within the iscsiadm discovery database from each configured iSCSI interface.

```
SLES12SP1:~ # iscsiadm -m node -l
Logging in to [iface: p3p1, target: iqn.2001-05.com.equallogic:0-af1ff6-
04ae5d4db-838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260]
(multiple)
Logging in to [iface: p3p2, target: iqn.2001-05.com.equallogic:0-af1ff6-
04ae5d4db-838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260]
(multiple)
Login to [iface: p3p1, target: iqn.2001-05.com.equallogic:0-af1ff6-04ae5d4db-
838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260] successful.
Login to [iface: p3p2, target: iqn.2001-05.com.equallogic:0-af1ff6-04ae5d4db-
838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260] successful.
```

When necessary, the host must log out of each discovered target from the PS Series array. The logout process is executed with the **iscsiadm -m node -u** command which logs out of every target defined within its discovered target database.

```
SLES12SP1:~ # iscsiadm -m node -u
Logging out of session [sid: 59, target: iqn.2001-05.com.equallogic:0-af1ff6-
04ae5d4db-838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260]
Logging out of session [sid: 60, target: iqn.2001-05.com.equallogic:0-af1ff6-
04ae5d4db-838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260]
Logout of [sid: 59, target: iqn.2001-05.com.equallogic:0-af1ff6-04ae5d4db-
838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260] successful.
Logout of [sid: 60, target: iqn.2001-05.com.equallogic:0-af1ff6-04ae5d4db-
838d6f20f2c57bf4-sles12sp1-intel02, portal: 10.10.10.10,3260] successful.
```

3.8 Multipath configuration

When using the HIT/Linux kit, the installed EqualLogic Host Connection Manager (ehcmd) service uses the host software iSCSI initiator to connect to the discovered volumes. Without the HIT/Linux kit, the multipathd service is needed to allow for multiple I/O paths (referred to as MPIO) between the host and storage array.

After the Linux host has established iSCSI connections to the PS Series array (section 3.7), the next steps are to obtain the Linux device name of the newly presented PS Series volume as well as its corresponding WWID for the targets being presented from the queried storage array. These values are used to build out the multipath configuration file for local MPIO access.

The following one-line command, using the **`lsscsi`** and **`scsi_id`** commands, can be used to get both the Linux device name and associated WWID for easier reference:

```
SLES12SP1:~ # for psVol in $(lsscsi |egrep -i eqlogic|awk -F' ' '{print $6}');
do echo -n "$psVol:"; scsi_id -g -u "$psVol"; done | sort -t":" -k2
/dev/sdf:360fff1ba4d5d8e08f47be5f2206f0da1
/dev/sdg:360fff1ba4d5d8e08f47be5f2206f0da1
```

Now that the WWIDs of the PS Series volumes are identified, the local `/etc/multipath.conf` can be built for local MPIO access to the volumes. Edit /etc/multipath.conf to include the previously found WWIDs from the PS Series array. The resulting configuration should be identical to the following sample for this single volume.

```
defaults {
        user_friendly_names     yes
        polling_interval        5
        path_selector   "round-robin 0"
        path_grouping_policy    multibus
        path_checker    tur
        no_path_retry   queue
}

blacklist_exceptions {
        #PS Array
        wwid " 360fff1ba4d5d8e08f47be5f2206f0da1"

}

multipaths {
        multipath {
                wwid    360fff1ba4d5d8e08f47be5f2206f0da1
                alias   PS_Vol_02
        }
}
```

Once the `/etc/multipath.conf` is configured with the correct WWID of the presented PS Series volume, local MPIO can then be enabled. The following steps compete this action.

1. Flush any existing multipath bindings.

   ```
   SLES12SP1:~ # multipath -F
   ```

2. Create new multipath bindings.

   ```
   SLES12SP1:~ # multipath –v2 -l
   ```

3. Start and enable the multipath service.

   ```
   SLES12SP1:~ # systemctl start multipathd.service
   SLES12SP1:~ # systemctl enable multipathd.service
   ```

4. Display the current multipath configuration showing the MPIO enabled volume with friendly name (highlighted in green), WWID (highlighted in yellow), and local device name (highlighted in blue).

   ```
   SLES12SP1:~ # multipath –ll
   PS_Vol_02 (360fff1ba4d5d8e08f47be5f2206f0da1) dm-1 EQLOGIC,100E-00
   size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
   `-+- policy='round-robin 0' prio=1 status=active
     |- 11:0:0:0 sdf 8:80  active ready running
     `- 12:0:0:0 sdg 8:96  active ready running
   ```

3.9 Mounting an MPIO volume

Now that multipath is configured and the volume has been enabled to use multiple paths for access from the Linux host, the volume can now be mounted and a filesystem applied to it for further use.

1. Create a mount point:

   ```
   SLES12SP1:~ # mkdir -p /media/PS_Vol_02
   ```

2. Create an XFS filesystem on the presented device, mount the device, and add an entry within **/etc/fstab** for persistent access across reboots. Dell recommends using the entire drive without partition for all non-boot drives.

   ```
   SLES12SP1-Intel:~ # mkfs.xfs /dev/mapper/PS_Vol_02
   meta-data=/dev/mapper/PS_Vol_02  isize=256    agcount=4, agsize=1311360
   blks
            =                       sectsz=512   attr=2, projid32bit=1
            =                       crc=0        finobt=0
   data     =                       bsize=4096   blocks=5245440, imaxpct=25
            =                       sunit=0      swidth=0 blks
   naming   =version 2              bsize=4096   ascii-ci=0 ftype=0
   log      =internal log           bsize=4096   blocks=2561, version=2
            =                       sectsz=512   sunit=0 blks, lazy-count=1
   realtime =none                   extsz=4096   blocks=0, rtextents=0
   ```

```
SLES12SP1:~ # mount /dev/mapper/PS_Vol_02 /media/PS_Vol_02/


SLES12SP1-Intel:~ # df -hT | awk 'NR==1 || /PS_Vol_02/'
Filesystem            Type      Size  Used Avail Use% Mounted on
/dev/mapper/PS_Vol_02 xfs        20G   33M   20G   1% /media/PS_Vol_02


SLES12SP1:~ # xfs_admin -u /dev/mapper/PS_Vol_02
UUID = 864a7926-0d53-4fc3-882f-245329669c10
```

3. Verify that the entry in the **/etc/fstab** file is similar to:

   ```
   UUID= 864a7926-0d53-4fc3-882f-245329669c10/media/PS_Vol_02 xfs
   defaults,_netdev 1 2
   ```

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

Note: The `dump` option of `1` and `pass` option of `2` in the prior `/etc/fstab` example can be changed to reflect the storage policies of data integrity within the applicable SAN environment.

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 PS 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 PS Series device definition by default. Little effort is required to configure simple multipath environments.

After the host SCSI subsystem detects a PS 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 five important default Device Mapper volume settings:

Setting	Description
polling_interval 5	Defines the time in seconds between the end of one path checking cycle and the beginning of the next patch checking cycle.
path_selector “round-robin 0”	Defines the load-balancing algorithm used to balance traffic across all active paths in a priority group.
path_grouping_policy multibus	Places all available paths to storage target into one priority group.
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 following example 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 following simple multipath configuration file assigns an easy-to-read alias to the device instead of the long SCSI ID string.

```
multipaths {
    multipath {
        wwid 360fff1ba4d5d4e0af47b05f3206f8d12
        alias PS_Vol_01
    }
}
```

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

```
SLES12SP1:~ # systemctl reload multipathd.service
```

Before changing the /etc/multipath.conf file, the output from **`multipath -ll`** shows the WWID for the mapped volume (highlighted in yellow).

```
SLES12SP1:~ # multipath -ll PS_Vol_01
360fff1ba4d5d4e0af47b05f3206f8d12 dm-3 EQLOGIC,100E-00
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:0:0 sdh 8:112 active ready running
  `- 14:0:0:0 sdi 8:128 active ready running
```

After the change to the multipath.conf file and reloading the multipath daemon, the output from **`multipath -ll`** shows the friendly name alias (highlighted in green) and WWID of the mapped volume.

```
SLES12SP1:~ # multipath -ll PS_Vol_01
PS_Vol_01 (360fff1ba4d5d4e0af47b05f3206f8d12) dm-3 EQLOGIC,100E-00
size=20G features='1 queue_if_no_path' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
  |- 13:0:0:0 sdh 8:112 active ready running
  `- 14:0:0:0 sdi 8:128 active ready running
```

4.2.1 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, PS 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 PS Series arrays.

4.2.2 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.

4.2.3 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.

4.2.4 XFS

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 as referenced in section 2.7.

4.2.5 Ext3 and 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.

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 PS Series arrays and their individual and combined demands is key to beginning the process. This tuning process is often iterative and uses 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: 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 iSCSI configuration changes individually and incrementally. Evaluate them against multiple workload types to understand the effects on overall performance. iSCSI tuning is often more time consuming due to 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, prioritize requests, and ensure I/O request completion before a specified deadline. Choosing the best I/O elevator depends on the workload and hardware. For a complete discussion, see section 12 in the *System Analysis and Tuning Guide* for SLES 12.

5.2 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.

5.3 SCSI queue variables

SCSI device queue settings can be tuned to improve performance. Sections 5.4–5.6 describe some common tunable settings. For multipath devices, the tunable values are in **/sys/block/dm-X/queue**. Tunable settings for block devices 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.

5.4 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 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.

5.6 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 section 12 of the *System Analysis and Tuning Guide* for SLES 12.

6 Useful tools

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

6.1 lsscsi

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

The following example contains output from the **lsscsi** command including disk and volume information:

```
SLES12SP1:~ # lsscsi
[0:2:0:0]    disk    DELL     PERC H310        2.12  /dev/sda
[5:0:0:0]    cd/dvd  HL-DT-ST DVD-ROM DU30N    A300  /dev/sr0
[7:0:0:0]    disk    EQLOGIC  100E-00          9.0   /dev/sdb
[8:0:0:0]    disk    EQLOGIC  100E-00          9.0   /dev/sdc
[9:0:0:0]    disk    EQLOGIC  100E-00          9.0   /dev/sdd
[10:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdf
[11:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sde
[12:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdg
[13:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdh
[14:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdi
[15:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdj
[16:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdk
[17:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdl
[18:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdm
[19:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdn
[20:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdo
[21:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdp
[22:0:0:0]   disk    EQLOGIC  100E-00          9.0   /dev/sdq
```

This output shows eight volumes from the PS Series array. There are multiple options for **lsscsi** that provide even more detailed information.

Notice in the output 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 host*x* 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, which is the target, corresponds to the front-end ports of the PS Series array. Finally, the last number is the LUN number or address for the volume listed.

6.2 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 following example includes output from **scsi_id**:

```
SLES12SP1:~ # /lib/udev/scsi_id -u -g /dev/sdh
360fff1ba4d5d4e0af47b05f3206f8d12
```

6.3 /proc/scsi/scsi

As mentioned in section 6.1, 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.

```
SLES12SP1:~ # cat /proc/scsi/scsi
<output truncated>
Host: scsi7 Channel: 00 Id: 00 Lun: 00
  Vendor: EQLOGIC  Model: 100E-00          Rev: 9.0
  Type:   Direct-Access                    ANSI  SCSI revision: 05
<output truncated>
```

6.4 /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.

```
SLES12SP1:~ # cat /proc/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=8110112k,nr_inodes=2027528,mode=755 0 0
<output truncated>
/dev/eql/PS_Vol_01 /media/PS_Vol_01 xfs rw,relatime,attr2,inode64,noquota 0 0
```

6.5 /sys/block/sdX/queue

Each Linux disk device, distinguished by sd*x*, has a set of separate parameters and setting which are found in the /sys pseudo-filesystem in the **/sys/block/sdx/queue**.

```
SLES12SP1:~ # ls /sys/block/sdb/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
```

6.6 dmesg

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

The following shows sample output from the **dmesg** command showing a LUN with a total of about 20 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.

```
SLES12SP1:~ # dmesg|egrep sdb
[354853.534398] sd 7:0:0:0: [sdb] 41963520 512-byte logical blocks: (21.4
GB/20.0 GiB)
[354853.534969] sd 7:0:0:0: [sdb] Write Protect is off
[354853.534972] sd 7:0:0:0: [sdb] Mode Sense: 81 00 00 00
[354853.535058] sd 7:0:0:0: [sdb] Write cache: disabled, read cache: enabled,
doesn't support DPO or FUA
[354853.542344]  sdb: unknown partition table
[354853.543637] sd 7:0:0:0: [sdb] Attached SCSI disk
```

A Additional resources

A.1 Technical support and resources

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

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 SP1 documentation from the vendor, see the following:

- SLES 12 SP1 *Administration Guide*
- *SLES 12 SP1 Deployment Guide*
- SLES 12 SP1 *Storage Administration*
- SLES 12 SP1 *System Analysis and Tuning*

For additional Dell Storage information, see the following:

- *Best practices for sharing an iSCSI SAN Infrastructure with Dell PS Series and Dell SC Series Storage using Linux Hosts*
- *Using HIT/Linux and ASM/LE with Dell PS Series Storage*
- *Red Hat Enterprise Linux Configuration Guide for Dell Storage PS Series Arrays*