Red Hat Enterprise Linux Configuration Guide for Dell Storage PS Series Arrays

Native RHEL Multipathing

Tom O'Loughlin, Storage Support Services Master Engineer
Randolph Nethers, Linux/UNIX Product Specialist
April 2016

A Dell Deployment and Configuration Guide

Revisions

Date	Revision	Description
April 2016	1.0	Release based RHEL 6 and 7 with native multipath

Table of contents

Executive summary

Red Hat® Enterprise Linux® (RHEL) version 7 was designed for mission-critical enterprise computing that brings together major improvements to make the operating system even more versatile and robust. This configuration guide provides information about integrating the RHEL 6 and 7 operating environments with the Dell™ Storage PS Series arrays using iSCSI technology. The guide includes guidance for working with volumes, using MPIO and some performance tips.

The server-side information included in this guide uses the command-line interface (CLI). The CLI is available to all UNIX and Linux operating systems and is often the most efficient way to accomplish the desired result.

For information about administering Dell PS Series arrays with Group Manager, see the *Dell EqualLogic Group Manager Administrator's Guide*.

Dell offers Host Integration Tools (HIT) for specific versions of Red Hat Enterprise Linux that integrate PS Series storage area networks (SANs) with hosts and applications. However, this configuration guide focuses on using tools that are native to Linux for iSCSI SAN administration and provide settings similar to HIT for Linux.

Note: Always follow proper change management and backup practices. Administrators should test changes to production in non-production environments whenever possible.

1 Operational necessities

This configuration guide assumes the reader is familiar with RHEL system administration, PS Series arrays and working with iSCSI SANs. Refer to the Appendix for relevant documentation to Linux system administration, PS Series arrays, and iSCSI-based SANs.

1.1 RHEL versions

Both RHEL 6.7 and 7.2 hosts were used to create this guide. The reader should be aware that procedures applicable to one version of RHEL may or may not work for another. For instance, the daemon startup and stop syntaxes are different between RHEL 6 and 7. When appropriate, this guide provides RHEL version-specific steps.

1.2 RHEL software requirements

Dell recommends keeping RHEL patches updated, especially security patches. Access to patches requires an active subscription with Red Hat. For details, refer to their website at http://www.redhat.com.

`iscsi-initiator-utils` is necessary for both RHEL 6 and RHEL 7 iSCSI connectivity and is installed using the yum package manager.

```
# yum -y install iscsi-initiator-utils
```

The `-y` switch is optional; it tells the yum program to answer all prompts with yes.

The device-mapper (multipath) packages are also needed for both RHEL 6 or RHEL 7. The packages are `device-mapper-multipath` and `device-mapper-multipath-libs`.

```
# yum -y install device-mapper-multipath device-mapper-multipath-libs
```

1.3 Starting necessary services

RHEL 6 and earlier versions use the SysV `init` scheduler. RHEL 7 uses the `systemd` scheduler. As a result, how an administrator starts and stops daemons is different between RHEL 6 and RHEL 7. Configuring boot time at startup also uses different syntaxes.

1.3.1 RHEL 6 services

For RHEL 6 to start the necessary services, use the `service` command. Use the `chkconfig` command to ensure the services start after a reboot.

RHEL 6 makes use of the legacy SysV `init` scheduler. Daemons are started and stopped using the `service` command to run `initd` scripts. For example, to start the multipath and iSCSI daemons on a RHEL 6 host, each must be started one at a time.

```
# service multipathd start
# service iscsi start
# service iscsid start
```

To make starting the daemons persistent between reboots, the `chkconfig` command updates runlevel information for the daemons:

```
# chkconfig multipathd on
# chkconfig iscsi on
# chkconfig iscsid on
```

1.3.2 RHEL 7 services

RHEL 7 introduced the use of the `systemd` system and service manager, which is far more robust and offers many options not available with SysV `init`. For more information on `systemd` see the article, "Overview of Systemd for RHEL7", and the chapter in the RHEL7 *System's Administrator Guide*, "Managing Services with systemd".
To start the necessary iSCSI and multipath daemons on a RHEL 7 system, run the following commands:

```
# systemctl start multipathd.service iscsi.service iscsid.service
```

Notice that `start` goes in front of the service rather than after. The `systemctl` command requires an action definition before the service name.
To ensure the daemons for iSCSI and multipath run at startup, replace `start` with `enable`.

```
# systemctl enable multipathd.service iscsi.service iscsid.service
```

1.4 Networking

The beauty of iSCSI protocol is that it can make use of existing Ethernet infrastructure, or use dedicated infrastructure that is familiar. The RHEL 6 and RHEL 7 distributions contain drivers for most common network manufacturers.

1.4.1 Network cards

Common networking cards found in Dell PowerEdge servers are Intel® or Broadcom® network interface cards (NICs). By default, both RHEL 6 and RHEL 7 installations include the 1G or 10G NIC drivers for either manufacturer. If updated drivers are needed, install them using yum, go to the Dell support website at support.dell.com, or refer to the vendor support website for the appropriate driver.

1.4.2 Jumbo frames

The standard Ethernet frame size is 1500 bytes. For most traffic this is perfectly acceptable, however, data may move more efficiently with a larger frame size. Jumbo frames are a good option if the I/O pattern may need a larger payload. The customary frame size for jumbo frames is 9000 bytes. Each device in the path between the array and the server must support and be configured to support jumbo frames.

Note: Not all switches support both jumbo frames and Flowcontrol simultaneously. Check with the switch vendor before enabling jumbo frames. Initially, run without jumbo frames to set a baseline and then look for improvements with jumbo frames enabled. It is important to note that higher payload requirements can hinder smaller I/O patterns.

For both RHEL 6 and RHEL 7, the MTU size is set in the `/etc/sysconfig/network-scripts/ifcfg-X` file, where X is the interface name. For example, if the interface name is em1, the configuration file would be `/etc/sysconfig/network-script/ifcfg-em1`. The variable that needs to be set is MTU. To affect the change, enter the following line in the appropriate configuration file:

```
MTU=9000
```

Note: For each additional NIC on the host connected to the SAN, repeat the steps above before restarting the network. To ensure the settings are correct, reboot the system and run the ifconfig or ip command below to verify the settings were made correctly.

To make the change effective, restart the network service, or reboot the system. For RHEL 6, the network can be restarted using the `service` command, and in RHEL 7 the `systemctl` command.

For RHEL 6:

```
# service network restart
```

For RHEL 7:

```
# systemctl restart network
```

Jumbo frames can be set at the command line until the next reboot. The method above results in persistent MTU values from one reboot to the next. For RHEL 6, the `ifconfig` command changes the interface MTU settings.

```
# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.6.156  netmask 255.255.255.0  broadcast 10.10.6.255
        inet6 fe80::ca1f:66ff:fed8:7bae  prefixlen 64  scopeid 0x20<link>
        ether c8:1f:66:d8:7b:ae  txqueuelen 1000  (Ethernet)
# ifconfig em1 mtu 9000
```

There is no output from the configuration command. The `ifconfig` command must be rerun to reflect the change.

```
# ifconfig em1
em1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 9000
        inet 10.10.6.156  netmask 255.255.255.0  broadcast 10.10.6.255
        inet6 fe80::ca1f:66ff:fed8:7bae  prefixlen 64  scopeid 0x20<link>
        ether c8:1f:66:d8:7b:ae  txqueuelen 1000  (Ethernet)
```

For RHEL 7, the iproute-3 package provides the IP command that can be used to manipulate the network interface.

```
# ip link set mtu 9000 dev em1
```

Output is not automatically displayed with configuration. Display the interface values using:

```
# ip a show em1 | grep mtu
2: em1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 9000 qdisc mq state UP qlen
1000
```

1.4.3 Flow control

Ethernet flow control is a means of pausing data transmission to help ensure zero data loss during network congestion. Flow control must be set to RX (receive) = on and TX (transmit) = on for every NIC connected to the SAN. Use the `ethtool` command and the em1 interface to test and set flow control settings.

```
# ethtool -a em1
Pause parameters for em1:
Autonegotiate:  on
RX:             on
TX:             on
```

In the event that RX or TX are not on, set the value with `ethtool` and the `-A` option.

```
# ethtool -A em1 rx on tx on
```

To ensure the change is persistent across reboots, add the following line to the `ifcfg-X` configuration file.

```
ETHTOOL_OPTS="rx on tx on"
```

1.4.4 Creating multiple iSCSI Interfaces

If there are multiple iSCSI interfaces available for multipath, use the `iscsiadm` command to make the iSCSI daemon aware of these interfaces. For example, two interfaces for iSCSI traffic (em1 and em2) are added as follows:

```
# iscsiadm -m iface -I em1 -o new
New interface em1 added
# iscsiadm -m iface -I em2 -o new
New interface em2 added
# iscsiadm -m iface -I em1 --op=update -n iface.net_ifacename -v em1
em1 updated.
# iscsiadm -m iface -I em2 --op=update -n iface.net_ifacename -v em2
em2 updated.
```

2 Working with volumes

This section explores discovering the volumes available to a host, logging in and out of those volumes, using Challenge-Handshake Authentication Protocol (CHAP), and checking the volume connectivity status.

For PS Series array management, volume creation and volume access control, see the Rapid EqualLogic Configuration Portal, and the *Dell EqualLogic Group Manager Administrator's Guide*. Limit volume access by initiator (client IP) address, initiator iSCSI qualified name (IQN) or by using CHAP.

2.1 Target discovery

Target discovery is required for access to volumes. The `iscsiadm` command, which is the open iSCSI administration utility, is used frequently throughout this paper. To view the database of discovered targets, use the `iscsiadm` command as follows:

```
# iscsiadm –m discoverydb
```

End the command with `discoverydb` to display all of the respective records. If no iSCSI SAN is in use, there will be no output. To discover a target and add it to the database, use `iscsiadm`. In the following example, the PS Series group address is 10.10.6.50.

```
# iscsiadm -m discovery -t st -p 10.10.6.50:3260
10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-
1f43a2e8b6356f3d-acct1
10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2
```

Two volumes in this example were discovered (**acct1** and **acct2**) and used in later examples.

2.2 iSCSI volume login

Volumes can be accessed from the target either simultaneously, as normal, or individually. To access all of the volumes at the same time, use the `iscsiadm` command in node mode with the `-l` argument.

```
# iscsiadm -m node -l
Logging in to [iface: default, target: iqn.2001-05.com.equallogic:4-
52aed6-d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260]
(multiple)
Logging in to [iface: default, target: iqn.2001-05.com.equallogic:4-
52aed6-fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10. 6.50,3260]
(multiple)
Login to [iface: default, target: iqn.2001-05.com.equallogic:4-52aed6-
d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260] successful.
Login to [iface: default, target: iqn.2001-05.com.equallogic:4-52aed6-
fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] successful.
```

To login to a specific volume, the entire target name is required (for example, iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-1f43a2e8b6356f3d-acct1 is the target name for acct1 and acct2 is iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-7073a2e8bbf56f3f-acct2).

To log into acct2, execute the following command:

```
# iscsiadm -m node –T iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2 –l –p 10.10.5.60:3260
Logging in to [iface: default, target: iqn.2001-05.com.equallogic:4-
52aed6-fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260]
(multiple)
Login to [iface: default, target: iqn.2001-05.com.equallogic:4-52aed6-
3260] successful.
```

2.3 Logout

If logging out of one volume or all the volumes is necessary, use the `iscsiadm` command.

To log out of a specific volume, the target name is used with the `-u` argument.

```
# iscsiadm -m node -T iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2 -u -p 10.10.6.50:3260
Logging out of session [sid: 2, target: iqn.2001-05.com.equallogic:4-
52aed6-fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260]
Logout of [sid: 2, target: iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] successful.
```

To log out of all the volumes, use `-u` without a target name.

```
iscsiadm -m node -u
Logging out of session [sid: 1, target: iqn.2001-05.com.equallogic:4-
52aed6-d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260]
Logging out of session [sid: 3, target: iqn.2001-05.com.equallogic:4-
52aed6-fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260]
Logout of [sid: 1, target: iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-
1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260] successful.
Logout of [sid: 3, target: iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] successful.
```

2.4 CHAP

CHAP is a network login protocol that uses a challenge-response mechanism. It is useful for restricting iSCSI access to volumes and snapshots to hosts that authenticate using the proper account name and password. CHAP provides a useful method of volume access control because it restricts access through account names and passwords, instead of IP addresses or iSCSI initiator names. For this discussion, it is the iSCSI initiator (client) that is authenticated by the iSCSI target (server). When the initiator attempts to login to its volumes, the target requires authentication by user name and password. CHAP accounts can be local or can be supplied by an external RADIUS server. Using RADIUS is beyond the scope of this document; refer to the Rapid EqualLogic Configuration Portal for a discussion of implementing RADIUS.

CHAP parameters can be stored in the `/etc/iscsi/iscsid.conf` file. The pertinent lines in the file are:

```
node.session.auth.authmethod = CHAP
node.session.auth.username = username
node.session.auth.password = password
discovery.sendtargets.auth.authmethod = CHAP
discovery.sendtargets.auth.username = username
discovery.sendtargets.auth.password = password
```

Be sure the lines are not proceeded with a hash symbol. A hash symbol constitutes the beginning of a comment that is not used for configuration. For a specific target, the `iscsiadm` command is the command to use to login with CHAP credentials.

```
# iscsiadm –m node –T <target name> -p <Group IP>:3260 –op=update –name
node.session.auth.authmethod –value=CHAP
# iscsiadm –m node –T <target name> -p <Group IP>:3260 –op=update –name
node.session.auth.username –value=<CHAP username>
# iscsiadm –m node –T <target name> -p <Group IP>:3260 –op=update –name
node.session.auth.password –value=<CHAP password>
# iscsiadm –m node –T <target name> -p –l
```

2.5 Status check

Information about the host connection to the target volumes is available using `iscsadm`. All active sessions and connections are displayed. There are various levels of information available. The –P3 option provides detail about each volume the host is logged into, where –P (for printlevel) is the level of detail provided (–P1 gives a little detail, and –P3 gives much more information). The below command provides dozens of lines of output for each device.

```
# iscsiadm -m session -P3
```

2.6 Disk timeout values

The PS Series arrays can deliver more network I/O than an initiator can handle, resulting in dropped packets and retransmissions. Other momentary interruptions in network connectivity can also cause problems, such as a mount point becoming read-only as a result of interruptions. To mitigate against unnecessary iSCSI resets during very brief network interruptions, change the value the kernel uses. The default setting for Linux is 30 seconds. This can be verified using the command:

```
# for i in $(find /sys/devices/platform –name timeout ) ; do cat $i ; done
30
30
```

To increase the time it takes before an iSCSI connection is reset to 60 seconds, use the command:

```
# for i in $(find /sys/devices/platform –name timeout ); do echo "60" >
$i; done
```

To verify the changes, re-run the first command.

```
# for i in $(find /sys/devices/platform –name timeout ); do cat $i; done
60
60
```

When the system is rebooted, the timeout value will revert to 30 seconds, unless the appropriate udev rules file is created. Create a file named `/lib/udev/rules.d/99-eqlsd.rules` and add the following content:

```
ACTION!="remove", SUBSYSTEM=="block", ENV{ID_VENDOR}=="EQLOGIC", RUN+="/bin/sh –
c 'echo 60 > /sys/%p/device/timeout'"
```

To test the efficacy of the new udev rule, reboot the system. Test that the reboot occurred, and then run the "cat $i" command above.

```
# uptime
12:31:22 up 1 min, 1 user, load average: 0.78, 0.29, 0.10
# for i in $(find /sys/devices/platform –name timeout ) ; do cat $i ; done
60
60
```

2.7 FastAbort setting

After an initiator has sent a task management function (like an ABORT TASK or LOGICAL UNIT RESET), PS Series arrays prefer to continue responding to R2Ts (ready to transfer). Therefore, set FastAbort to **No**.

In the `/etc/iscsi/iscsid.conf` file add the line:

```
node.session.iscsi.FastAbort = No
```

For this change to take effect, restart the iscsi and iscsid processes; instructions are available in section 1.3.

3 MPIO and device mapper

Linux Device Mapper (DM)-Multipath permits multiple I/O paths between the server node and host. The multipath daemon ensures that connectivity to storage is maintained while checking for failed paths, reconfigures paths as needed, and it aggregates I/O paths to maintain performance and redundancy.

The device mapper works with multipath to create a device for every unique SCSI ID and path combination from the storage array. It organizes the I/O paths logically and creates a single multipath device on top of underlying devices. The overlaid devices are persistent between reboots, which is important in cases where a persistent device filename is required by an application. A thorough treatment of DM Multipath is available as part of the online RHEL 7 product documentation titled *DM Multipath*.

3.1 Multipath configuration

The system refers to the `/etc/multipath.conf` file for settings, but the configuration file is often not created at the time of installation. The information below could be added to the `multipath.conf` file to configure multipath behavior, and to direct the device mapper to blacklist certain devices that do not need treatment by the device mapper. Only devices connected to the host through the SAN should be manipulated by the device mapper. Local disks should be ignored as their connectivity does not change.

3.1.1 Device default values

A number of defaults are recommended by Dell PS Series support. The defaults statement below should be placed in the `/etc/multipath.conf` file.

```
defaults {
        polling_interval        10
        path_selector           "round-robin 0"
        path_grouping_policy    multibus
        path_checker            tur
        rr_min_io_rq            10
        max_fds                 8192
        rr_weight               priorities
        failback                immediate
        features                0
}
```

The following list provides a brief explanation of each of these values.

- The *polling_interval* is the amount of time between path checks in seconds.
- The *path_selector* determines the algorithm to use for I/O. The "round-robin 0" option sends data through every path in a round-robin pattern, sending the same amount of I/O down each path.
- The *path_grouping_policy* sets how paths are grouped. With multibus, all paths are placed in the same group priority.
- The path_checker value tells multipath how to determine path states. TUR stands for the TEST UNIT READY command, which is used to test path state.

- The value *rr_min_io_rq* sets the number of I/O to route to a path before switching to the next in the same path group. This parameter is a tuning option. Values between 10 and 20 tend to work best in SQL environments, where larger values (from 100 to 512) work best for sequential loads. Values larger than 200 require increases to the node.session.cmds_max and node.session.queue_depth values in the /etc/iscsi/iscsid.conf file. (See section 4.4, "iSCSI daemon settings".)
- The *max_fds* value specifies the maximum number of file descriptors that can be opened by multipath and multipathd. The *max_fds* value is equivalent to `ulimit -n` value.
- *Failback* tells `multipathd` how to manage a restored path. The value *immediate* causes `multipathd` to immediately failback to the highest priority path grouping that contains active paths.
- The value *features* indicated whether any device-mapper features are to be used. A value of '0' indicates no features are to be used.

3.1.2 Blacklisting

The following entry in the /etc/multipath.conf file will prevent most local drives from being managed by multipath. Information about blacklisting devices can be found in the Red Hat knowledgebase document at redhat.com.

```
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
}
```

3.1.3 Aliases

An alias can be assigned to a specific device that is easier to read and often easier to manage. For instance, if a volume was presented to the host specifically for accounting, the volume might be named ACCT1. Add an entry for each volume to the `multipath.conf` file using the WWID of the volume, which will correlate the WWID with the alias. A good reference about `multipath.conf` is found in the *Red Hat Enterprise Linux 6* manual in chapter three, "Setting Up DM-Multipath".

The example server has two volumes that can be viewed using the `multipath -ll` command. For either RHEL6 or RHEL7, the attached devices can be viewed using the `multipath -ll` command, which contains the WWID for each volume.

```
# multipath -ll
mpathb (364ed2a35b7a080d23d6f35b6e8a2431f) dm-7 EQLOGIC ,100E-00
size=200G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   - 13:0:0:0 sdc 8:32 active ready running
mpatha (364ed2a35b7a000fd3f6ff5bbe8a27370) dm-9 EQLOGIC ,100E-00
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   - 14:0:0:0 sdg 8:64 active ready running
```

The thirty-three digit hexadecimal number between the parentheses is the WWID for the particular volume. Using the WWIDs, the entries in the `multipath.conf` file is:

```
multipaths {
        multipath {
                # Accounting LUN 1 - PS Series LUN ID 16 (100 GB) mpatha
                wwid    364ed2a35b7a000fd3f6ff5bbe8a27370
                alias   ACCT1
        }
        multipath {
                # Accounting LUN 2 - PS Series LUN ID 17 (100 GB) mpathb
                wwid    364ed2a35b7a080d23d6f35b6e8a2431f
                alias   ACCT2
        }
}
```

After editing the `multipath.conf file`, the multipath daemon needs to be restarted. For RHEL 6, the service program is used, and for RHEL 7, the `systemctl` program is used.

For RHEL 6:

```
# service multipathd restart
```

For RHEL 7:

```
# systemctl restart multipathd.service
```

For either RHEL6 or RHEL7, the attached devices can be viewed using the `multipath -ll` command. Now, rather than displaying the 'mpath' names, the aliases placed in the configuration file are listed.

```
# multipath -ll
ACCT2 (364ed2a35b7a080d23d6f35b6e8a2431f) dm-7 EQLOGIC ,100E-00
size=200G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   - 13:0:0:0 sdc 8:32 active ready running
ACCT1 (364ed2a35b7a000fd3f6ff5bbe8a27370) dm-9 EQLOGIC ,100E-00
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   - 14:0:0:0 sdg 8:64 active ready running
```

3.2 Adding a filesystem to a device

Aside from the raw device name, there are other ways to persistently reference a file system. One common practice is to use the universally unique identifier (UUID) of a volume. The UUID is a 128-bit standard identifier that is used in software construction. Typically, the UUID is expressed in a hyphenated hexadecimal format making it more readable. Because it is unique, the UUID can be used in distributed systems. There is a one in 2^{128} (3.4×10^{38}) chance of a hash collision, where two identifiers have the same exact value.

The UUID is created with the file system. From the example above, the second larger volume (ACCT2) was used to create an EXT4 file system.

```
# mkfs.ext4 /dev/mapper/ACCT2
mke2fs 1.42.9 (28-Dec-2013)
Discarding device blocks: done
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
13115392 inodes, 52431360 blocks
2621568 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=2202009600
1601 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
2654208,
        4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information: done

# blkid /dev/mapper/ACCT2
/dev/mapper/ACCT2: LABEL="" UUID="0b12159a-f55b-4706-9acf-c2d723f0dba3"
TYPE="ext4"
```

The UUID is discovered using the `blkid` command. The file system can be mounted using its aliased device name or the UUID.

3.3 Mounting the filesystem

Using the aliased device name:

```
[root@rhel7x ~]# mount -t ext4 /dev/mapper/ACCT2 /acct2
[root@rhel7x ~]# df -h /acct2
Filesystem         Size  Used Avail Use% Mounted on
/dev/mapper/ACCT2  197G   61M  187G   1% /acct2
```

Using the UUID:

```
# mount -t ext4 -u 0b12159a-f55b-4706-9acf-c2d723f0dba3 /acct2
```

```
# df -h /acct2
Filesystem          Size  Used Avail Use% Mounted on
/dev/mapper/ACCT2   197G   61M  187G   1% /acct2
```

3.4 Mounting during boot

File systems entered into the file system table (/etc/fstab) will be mounted on each reboot. The example below shows the entry for ACCT2.

```
# grep acct2 /etc/fstab
UUID=0b12159a-f55b-4706-9acf-c2d723f0dba3 /acct2    ext4
   _netdev,noatime,nodiratime,discard,defaults,     0 0
```

Again, either the device mapper path, /dev/mapper/ACCT2 or the UUID will function properly. The path, /acct2, is the path to the space on the PS Series array.

The *discard* option for the file systems available in RHEL 6 and RHEL 7 takes advantage of UNMAP support in PS Series arrays. The SCSI UNMAP command reclaims space from the storage blocks that have been deleted by the host and makes them available for later use. This reduces the in-use space of a Thin Provisioned volume and allows more free pool space in the PS Series group.

The *_netdev* option tells the operating system to wait to mount the volume until the network is available. Obviously, mounting an iSCSI volume is not possible until the operating system has network access to the array.

Using *noatime* and *nodiratime* disables writing the last time a file or directory was read. This can impact backups that use last access times. However, disabling write access times removes a write that would otherwise follow every read, which also reduces snapshot and replica sizes.

3.5 MPIO failover

RHEL provides native multipath through the device-mapper-multipath package. One purpose of multipath is to provide a robust storage solution through path redundancy. For path redundancy to work, there needs to be two or more paths to the PS Series array. The examples above assumed that one interface (em1) was available for iSCSI connectivity. If the em2 interface is also available, it must be added.

Invoke the `iscsiadm` command with iface mode to add em2.

```
# iscsiadm –m iface -I em2 –o new
# iscsiadm –m iface -I em2 --op=update –n iface.net_ifacename –v em2
```

These commands add a file in `/var/lib/iscsi/ifaces` called em2 with the following lines of information:

```
iface.iscsi_ifacename = em2
iface.net_ifacename = em2
iface_hwaddress = default
```

```
iface.transport_name = tcp
```

Volumes discovered on the first interface need also to be discovered on the new interface.

```
iscsiadm -m discovery -t st -p 10.10.6.50:3260
10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-
1f43a2e8b6356f3d-acct1
10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-
1f43a2e8b6356f3d-acct1
10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2
10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2
```

Verify that there is an entry for each volume with every path. Both em1 and em2 are dedicated to provide access to the PS Series array; the output reflects this fact. After the new path has been discovered, login to the volumes again.

```
# iscsiadm -m node –l
Logging in to [iface: em1, target: iqn.2001-05.com.equallogic:4-52aed6-
d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260] (multiple)
Logging in to [iface: em2, target: iqn.2001-05.com.equallogic:4-52aed6-
d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260] (multiple)
Logging in to [iface: em1, target: iqn.2001-05.com.equallogic:4-52aed6-
fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] (multiple)
Logging in to [iface: em2, target: iqn.2001-05.com.equallogic:4-52aed6-
fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] (multiple)
Login to [iface: em1, target: iqn.2001-05.com.equallogic:4-52aed6-
d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260] successful.
Login to [iface: em2, target: iqn.2001-05.com.equallogic:4-52aed6-
d280a0b73-1f43a2e8b6356f3d-acct1, portal: 10.10.6.50,3260] successful.
Login to [iface: em1, target: iqn.2001-05.com.equallogic:4-52aed6-
fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] successful.
Login to [iface: em2, target: iqn.2001-05.com.equallogic:4-52aed6-
fd00a0b73-7073a2e8bbf56f3f-acct2, portal: 10.10.6.50,3260] successful.
```

The sessions mode of `iscsiadm` shows that both adapters are connected to the PS Series array.

```
# iscsiadm -m session
tcp: [10] 10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-
1f43a2e8b6356f3d-acct1 (non-flash)
tcp: [11] 10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-d280a0b73-
1f43a2e8b6356f3d-acct1 (non-flash)
tcp: [12] 10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2 (non-flash)
tcp: [13] 10.10.6.50:3260,1 iqn.2001-05.com.equallogic:4-52aed6-fd00a0b73-
7073a2e8bbf56f3f-acct2 (non-flash)
```

If there are multiple paths to a volume, the multipath command output reveals it.

```
# multipath -ll
ACCT2 (364ed2a35b7a080d23d6f35b6e8a2431f) dm-7 EQLOGIC ,100E-00
size=200G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   - 13:0:0:0 sdc 8:32 active ready running
   - 13:0:3:0 sdd 8:34 active ready running
ACCT1 (364ed2a35b7a000fd3f6ff5bbe8a27370) dm-9 EQLOGIC ,100E-00
size=100G features='0' hwhandler='0' wp=rw
`-+- policy='round-robin 0' prio=1 status=active
   - 14:0:0:0 sdg 8:64 active ready running
   - 14:0:3:0 sdh 8:66 active ready running
```

The new paths are underlined. Output for both ACCT1 and ACCT2 reflect the added connectivity.

4 Performance

This section presents some general knowledge and direction regarding some tuning options and variables available in Linux. These are a starting point from which to begin deciding how to achieve optimal performance for the particular environment. It helps to know the applications using the Dell PS Series and the demands those applications will make individually and in conjunction is very important to beginning the process. Often the tuning process is iterative. A few methods are available for evaluating the effectiveness of tuning attempts. Use perception, and metrics provided by the application using the storage are very useful.

4.1 Kernel settings

The `sysctl` command is an interface for examining and dynamically changing kernel parameters in Linux. The interface mechanism is exported to /proc/sys. The kernel has default settings that can be changed to optimize the system as needed. The user-defined kernel settings are stored in the /etc/sysctl.conf file exclusively for RHEL 6 and can be separated into the /etc/sysctl.conf files in the /etc/sysctl.d directory for RHEL 7.

1. When there are multiple iSCSI connections to a PS Series array, alter the Linux ARP behavior to prevent ARP resets (RST) from the initiator to the target, which would not allow more than one interface to serve traffic. Add the following information to the `/etc/sysctl.conf` file.

   ```
   net.ipv4.conf.em1.arp_ignore = 1
   net.ipv4.conf.em1.arp_announce = 2
   net.ipv4.conf.em1.rp_filter = 2
   net.ipv4.conf.em2.arp_ignore = 1
   net.ipv4.conf.em2.arp_announce = 2
   net.ipv4.conf.em2.rp_filter = 2
   ```

2. In a RHEL 6 system, add the following to the /etc/sysctl.conf file as well. In a RHEL 7 system, create a file called /etc/sysctl.d/equallogic.conf and add the same content.

   ```
   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 saved, read in the file to sysctl, with sysctl -p:

   a. For RHEL 6:

   ```
   # sysctl -p /etc/sysctl.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
```

b. For RHEL 7:

```
# sysctl -p /etc/sysctl.d/equallogic.conf
net.ipv4.conf.em1.arp_ignore = 1
net.ipv4.conf.em1.arp_announce = 2
net.ipv4.conf.em1.rp_filter = 2
net.ipv4.conf.em2.arp_ignore = 1
net.ipv4.conf.em2.arp_announce = 2
net.ipv4.conf.em2.rp_filter = 2
```

4.2 Readahead kernel

Readahead is a Linux kernel system call that loads the contents of a file into the kernel page cache. This way the file contents are read from main memory rather than storage. The goal is lowered file access latencies. However, changing the readahead value is only valuable with sequential I/O-type applications. Changing the value can cause performance problems with high random I/O.

Different distributions and kernel versions set their own values. RHEL 6 sets the default value to 256 sectors, which means 256 sectors are requested when doing a read. For RHEL 7, the value was increased to 8192 sectors.

The `blockdev` command can be used to change the readahead setting on an sd/dm device, telling the SCSI layer to read a specific number of sectors ahead. The `--getra` switch to the `blockdev` command will provide the readahead value.

```
# /sbin/blockdev --getra /dev/mapper/ACCT2
```

The `--setra` switch to the `blockdev` command sets the value.

```
# /sbin/blockdev --setra 4096 /dev/mapper/ACCT2
```

To make the value persistent across reboots, add the `--setra` command line to the `/etc/rc.local` file, for RHEL 6.

Note: 4096 sectors is just an example. With 4096 sectors, the readahead_kb value would be 2048k. Some experimentation might be necessary to determine the optimal value for the particular set of applications using the PS Series arrays.

Red Hat has modified the default number of readahead sectors to 8192 (4096k). In the Dell HIT for Linux v1.4, the readahead is decreased to the recommended 2048 value. However, testing may be necessary to determine the best value for the particular environment using native multipathing.

Setting persistence across reboots for RHEL 7 requires creating a file in `/lib/udev/rules.d` called `38-equallogic.rules` with the below information. (Notice that the readahead size is in KB rather than 512-byte sectors.)

```
# This skips the rules for any non-Dell PS Series devices
ENV{EQL_DEVICE}!= "1", GOTO="not_eql_device"
# Set the readahead value to 2048 KB for all Dell PS Series devices:
ATTR{queue/read_ahead_kb}="2048"
```

Reboot the RHEL 7 system and test the readahead value using the `--getra` command as shown earlier in this section.

4.3 Linux I/O scheduler

The purpose of I/O schedules is to reduce the number of seek operations, prioritize requests and ensure I/O request completion before a specified deadline. There are three types of I/O schedules (also called I/O elevators).

1. CFQ (Completely Fair Queuing) promotes I/O from real-time processes and uses historical data to anticipate an application that may issue more I/O requests in the near future.
2. Deadline attempts to provide a guaranteed latency for requests and dispatches I/O in batches. By default, reads have priority over writes since applications are more likely to block on read I/O.
3. Noop implements a simple FIFO (first in, first out) scheduling algorithm, helping CPU-bound systems with fast storage.

For example, to implement the cfq schedule at reboot, use the command below for either RHEL 6 or RHEL 7.

```
# grubby --update-kernel=ALL --args="elevator:cfq"
```

To remove the implementation, the below command reverses the change.

```
# grubby --update-kernel=ALL --remove-args="elevator:cfq"
```

More information regarding this is available in the "I/O Scheduler" discussion in the RHEL 7 *Performance Tuning Guide*.

4.4 iSCSI daemon settings

The iscsid.conf file dictates behavior of the iSCSI daemon. The settings in the configuration file can be set using the iscsiadm command, in node mode, with the `--op` command.

The uncommented entries for the iSCSI daemon configuration file are echoed to screen with the command below. There are comments in the configuration file that explain each entry in detail.

```
# cat /etc/iscsi/iscsid.conf | grep -v ^#
node.startup = automatic
```

```
node.session.timeo.replacement_timeout = 120
node.conn[0].timeo.login_timeout = 15
node.conn[0].timeo.logout_timeout = 15
node.conn[0].timeo.noop_out_interval = 5
node.conn[0].timeo.noop_out_timeout = 5
node.session.err_timeo.abort_timeout = 15
node.session.err_timeo.lu_reset_timeout = 20
node.session.cmds_max = 1024   # default is 128
node.session.queue_depth = 128 # default 32
node.session.iscsi.InitialR2T = No
node.session.iscsi.ImmediateData = Yes
node.session.iscsi.FirstBurstLength = 262144
node.session.iscsi.MaxBurstLength = 16776192
node.conn[0].iscsi.MaxRecvDataSegmentLength = 131072
discovery.sendtargets.iscsi.MaxRecvDataSegmentLength = 32768
node.session.iscsi.FastAbort = No # default is 'Yes'
```

Increase the node.session.cmds_max and node.session.queue_depth parameters from the default if the rr_min_io_rq value in the /etc/multipath.conf file is going to be increased higher than 200.

During target discovery, the iscsiadm tool creates node records in /var/lib/iscsi/nodes, when logging into a target. Discovery records are stored in /var/lib/iscsi in a directory that matches the discovery type (such as `sendtarget` from the `-t st` command).

Once the iscsid.conf file is changed, reconfigure the discovery record. This is done by running `iscsiadm` for discovery, as shown in section 2.1.

4.5 Testing the changes with DSTAT

The `dstat` package permits a system administrator to view all the system resources instantly. Network bandwidth and disk throughput can be directly compared during the same interval in numerous ways that make the data easy to use. If the `dstat` package is not installed, use the yum package manager to install it.

```
# yum -y install dstat
```

Test these changes by writing zeroes from the operating system to the PS Series volume. To accomplish the test, open two command line windows. In one window run the `dd` command.

```
# dd if=/dev/zero of=/acct1/mpio.test bs=1M count=100000
```

The `dd` command converts or copies files. This example is writing zeroes generated by the kernel from the /dev/zero device file (if = input file) to a file on the ACCT1 volume called mpio.test (of = output file). One hundred thousand (count=100000) or one-megabyte sized blocks (bs=1M) of zeros, are being written to the mpio.test file. This data is 100 GB and provides enough time to deliver a good view of performance.

While the `dd` command is running, use `dstat` in another command line window to observe network states for specific Ethernet interfaces. Our example system has em1 and em2 interfaces assigned to iSCSI traffic.

The `-t` argument causes `dstat` to output time and other data, `-n` directs `dstat` to display network statistics, and `-N` is a comma-delimited list of network interfaces, and the output is displayed every (1) second.

```
# dstat -t -n -N em1,em2 1
----system----  --net/em1-   --net/em2-
  date/time     recv  send:  recv  send
29-03 13:01:46|   0     0 :    0     0
29-03 13:01:47|1399B  801B: 8909B  662B
29-03 13:01:48| 520k   57M:  890k  113M
29-03 13:01:49| 521k   72M: 1001k  155M
29-03 13:01:50| 501k   78M: 1090k  139M
29-03 13:01:51| 532k   81M: 1044k  132M
29-03 13:01:52| 587k   82M: 1024k  145M
29-03 13:01:53| 436k   79M: 1190k  162M
29-03 13:01:54| 467k   72M: 1112k  154M
29-03 13:01:55| 533k   78M: 1023k  155M
```

The output ends after the `dd` command completes or sooner if the command is cancelled by pressing [**Ctrl**] and [**C**].

```
29-03 13:03:12| 499k   88M: 1002k  162M
29-03 13:03:13| 320k   37M:  522k   70M
29-03 13:03:14|1522B  334B: 1498B  179B
29-03 13:03:15|1243B  150B:  971B  180B
29-03 13:03:16| 761B  201B:  836B    0
29-03 13:03:17| 437B  101B:  832B    0
```

After the test, be sure to remove the test file:

```
# rm -f /acct1/mpio.test
```

A 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	US Phone support
PS Series (EqualLogic)	http://eqlsupport.dell.com	eqlx-customer-service@dell.com	800-945-3355
SC Series and Compellent	https://customer.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
XC Series	http://www.dell.com/support	Specific to service tag	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.1 Related documentation

Item	Resource
Ret Hat KB Article	"Overview of Systemd for RHEL7"
RHEL 6 MPIO Setup	"Setting Up DM-Multipath"
RHEL7 System Admin's Guide	"Managing Services with systemd"
RHEL 7 Overview on Systemd	"Overview of Systemd"
RHEL 7 DM Multipath	"DM Multipath"
RHEL 7 Security	"Using Firewalls"
RHEL 7 I/O Scheduler/Performance Tuning	"I/O Scheduler"
Dell Resource	
Host Integration (HIT) Kit Guide for Linux	
Dell EqualLogic Group Manager Administrator's Guide (Login required)	
Best practices for sharing an iSCSI SAN Infrastructure with Dell PS Series and Dell SC Series Storage using Linux Hosts	