Linux Software RAID volumes with Dell PowerEdge Express Flash PCIe-SSD

A Dell Technical White Paper

Jose De la Rosa
Jordan Hargrave
Dell Linux Engineering

THIS WHITE PAPER IS FOR INFORMATIONAL PURPOSES ONLY, AND MAY CONTAIN TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES. THE CONTENT IS PROVIDED AS IS, WITHOUT EXPRESS OR IMPLIED WARRANTIES OF ANY KIND.

May 2012

Contents

1. Overview

This whitepaper describes how to create and manage Linux software RAID volumes with Dell PowerEdge Express Flash PCIe-SSDs. PCIe-SSD devices are a high-performance storage solution designed for environments that require low latencies and high disk I/O operations (for detailed information see here). Because PCIe-SSD devices do not support a hardware storage controller that can create and manage hardware RAID volumes (akin to Dell PERC controllers), Linux environments that require high availability capabilities can use Linux software RAID instead.

The whitepaper discusses management of software RAID volumes with the 'mdadm' tool and booting to a software RAID volume on PCIe-SSD devices. The whitepaper does not discuss which use cases are better suited for software RAID volumes; we will discuss such use cases in a follow-up whitepaper.

These instructions apply to Red Hat Enterprise Linux 6 SP1/SP2 and SuSE Linux Enterprise Server 11 SP2 and require the mtip32xx add-on Linux driver available at support.dell.com. It is assumed that the reader is fairly proficient in Linux system administration tasks.

Please note that when referring to RAID devices, the terms "array", "volume" and "virtual disk" are used interchangeably. For consistency purposes we use the term "volume". However, the operating system uses the term "array".

2. RAID levels

Each RAID level provides different types of high-availability and failover capabilities. Your selection will depend on the number of disks you have and the level of availability and serviceability you require.

Table 1 table below shows a brief summary of the most commonly used RAID levels:

RAID Level	Description	Redundancy	Minimum drives required
0	Striping without parity	No	1
1	Mirroring	Yes	2
5	Striping with distributed parity (parity across all disks)	Yes	3
6	Striping with dual parity (two levels of parity across all disks)	Yes	4
10	Combination of RAID 1 and RAID 0 (striping of mirrors)	Yes	4
50	Combination of RAID 5 and RAID 0 (striping of distributed parity arrays)	Yes	6
60	Combination of RAID 6 and RAID 0 (striping of dual parity arrays)	Yes	8

Table 1 – RAID levels

Click here for a more detailed description of the different types of RAID levels available.

3. Creating software RAID volumes

The command to create a software RAID volume is straight-forward. Here are some examples:

Example 1: Create a RAID 5 volume /dev/md0, using 3 PCIe-SSD drives (/dev/rssda, /dev/rssdb and /dev/rssdc):

```
# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3
/dev/rssd{a,b,c}
```

Example 2: Create a RAID 1 volume /dev/md0, using 2 PCIe-SSD drives (/dev/rssda and /dev/rssdb):

```
# mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/rssd{a,b}
```

Example 3: Create a RAID 10 volume /dev/md0, using 4 PCIe-SSD drives (/dev/rssda, /dev/rssdb, /dev/rssdc and /dev/rssdd):

```
# mdadm --create --verbose /dev/md0 --level=10 --raid-devices=4
/dev/rssd{a,b,c,d}
```

For all options available, refer to the 'mdadm' man page.

Continuing with Example 1 above, to view the status of the new RAID volume (also known as MD device), run:

```
# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Wed Mar 28 16:52:02 2012
     Raid Level : raid5
     Array Size : 341881856 (326.04 GiB 350.09 GB)
  Used Dev Size : 170940928 (163.02 GiB 175.04 GB)
   Raid Devices : 3
  Total Devices : 3
    Persistence : Superblock is persistent

    Update Time : Wed Mar 28 16:52:02 2012
          State : clean, degraded, recovering
 Active Devices : 2
Working Devices : 3
 Failed Devices : 0
  Spare Devices : 1

         Layout : left-symmetric
     Chunk Size : 512K

 Rebuild Status : 2% complete

           Name : 0
           UUID : 49ac3b7b:07680fbb:42f471e9:7ba94d07
         Events : 1

    Number   Major   Minor   RaidDevice State
       0      252        0        0      active sync   /dev/rssda
       1      252      256        1      active sync   /dev/rssdb
       3      252      512        2      spare rebuilding   /dev/rssdc
```

As you can see from the text highlighted in red, the RAID 5 volume is in process of being built (2% complete).

Next, create a configuration file to make the volume name persistent across reboots:

```
# mdadm --examine --scan > /etc/mdadm.conf
```

The file /etc/mdadm.conf will look something like:

```
ARRAY /dev/md/0 metadata=1.2 UUID=49ac3b7b:07680fbb:42f471e9:7ba94d07 name=0
```

The UUID is a unique identifier for the MD device, so yours will be different.

Once the volume is finished building, it is ready to be used. You could optionally create a file system on top of it just like you would with any other storage device. For example:

```
# mkfs.ext4 /dev/md0
# mkdir /data
# mount /dev/md0 /data
```

4. Extending software RAID volumes

It is possible to extend the number of drives that are part of a software RAID volume, which can either serve as spares or be active parts of the volume.

In this example, we have a RAID 5 volume with 3 PCIe-SSD drives: /dev/rssda, /dev/rssdb and /dev/rssdc. We have inserted a 4th PCIe-SSD drive (i.e. /dev/rssdd) to our server, and would now like to add it to the volume.

Before we start, it is highly recommended to back up your existing data and stop all volume I/O. If you created a file system and is mounted, be sure to unmount it first. To add the new drive to an existing volume (i.e. /dev/md0), run:

```
# mdadm /dev/md0 --add /dev/rssdd
```

The new drive will be added as a spare:

```
# mdadm --detail /dev/md0 | tail -n 6
    Number   Major   Minor   RaidDevice State
       0      252        0        0      active sync   /dev/rssda
       1      252      256        1      active sync   /dev/rssdb
       3      252      512        2      active sync   /dev/rssdc

       4      252      768        -      spare   /dev/rssdd
```

To add the new drive to the volume and be used, run:

```
# mdadm --grow /dev/md0 --raid-devices=4
  mdadm: Need to backup 3072K of critical section..
```

Verify that the new PCIe-SSD device (i.e. /dev/rssdd) was added and that the volume is “reshaping”:

```
# mdadm --detail /dev/md0 | tail -n 5
    Number   Major   Minor   RaidDevice State
       0      252        0        0      active sync   /dev/rssda
       1      252      256        1      active sync   /dev/rssdb
       3      252      512        2      active sync   /dev/rssdc
       4     252      768        3      active sync   /dev/rssdd
```

Depending on the size of the software RAID volume, it will take anywhere from 5 to 30 minutes to reshape and resync the volume.

5. Replacing a drive from a software RAID volume

Except for RAID 0, all RAID configurations have redundancy which means they can continue to operate as normal in case any of the drives fail. When a drive fails, you may see something like this:

```
# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Thu Mar 29 16:07:31 2012
     Raid Level : raid5
     Array Size : 512822784 (489.07 GiB 525.13 GB)
  Used Dev Size : 170940928 (163.02 GiB 175.04 GB)
   Raid Devices : 4
  Total Devices : 4
    Persistence : Superblock is persistent

    Update Time : Fri Mar 30 12:05:01 2012
          State : clean, degraded
 Active Devices : 3
Working Devices : 3
 Failed Devices : 1
  Spare Devices : 0

         Layout : left-symmetric
     Chunk Size : 512K

           Name : 0
           UUID : 2e069751:d500a4c1:891ae8bc:a019c309
         Events : 2109

    Number   Major   Minor   RaidDevice State
       0       0        0        0      removed
       1     252      256        1      active sync   /dev/rssdb
       3     252      512        2      active sync   /dev/rssdc
       4     252      768        3      active sync   /dev/rssdd

       0     252        0        -      faulty spare   /dev/rssda
```

In this example, we have a software RAID 5 volume and drive /dev/rssda has had a hardware failure (it is no longer detected by the operating system). Because this is RAID 5 (striping with distributed parity) the volume continues to operate normally, though in degraded state.

Let's remove the faulty drive /dev/rssda from the volume:

```
# mdadm /dev/md0 --remove /dev/rssda
mdadm: hot removed /dev/rssda from /dev/md0
```

Because hot-removal and hot-insertion is supported with RHEL 6 SP1/SP2 and SLES 11 SP2, you should be able to replace the PCIe-SSD drive without rebooting your server. Please refer to the documentation at support.dell.com for instructions on how to prepare and remove a PCIe-SSD drive from your Dell PowerEdge server. Once you have replaced the drive, run the following to add it to the software RAID volume:

```
# mdadm /dev/md0 --add /dev/rssda
```

In this example, we assume that the new PCIe-SSD drive was enumerated as /dev/rssda (same as the original), which may not always be the case.

After you add the new drive to the volume, it will start rebuilding. You can check the status with “`mdadm --detail /dev/md0`” or check the contents of */proc/mdstat*:

```
# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 rssda[5] rssdc[3] rssdb[1] rssdd[4]
      512822784 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/3] [_UUU]
      [===>.................]  recovery = 16.6% (28444132/170940928)
finish=13.0min speed=182260K/sec

unused devices: <none>
```

6. Removing software RAID volumes

In this example, we will be removing /dev/md0, which consists of 4 PCI-SSD drives (/dev/rssda through /dev/rssdd):

Stop the device:

```
# mdadm --stop /dev/md0
```

Clear out the MD metadata on all the PCIe-SSD drives:

```
# mdadm --zero-superblock /dev/rssda /dev/rssdb /dev/rssdc /dev/rssdd
```

7. Booting from software RAID volumes

Disclaimer: As of this writing (May 2012), booting from PCIe-SSD drives is not supported by Dell. However, if you’re the adventurous type, here is how you do it:

Booting from PCIe-SSD drives requires using UEFI Boot (United Extensible Firmware Interface). UEFI is a pre-boot environment that supports devices that may not be supported for the traditional BIOS boot method. UEFI requires using a GPT (GUID Partition Table) labeled disk instead of the traditional MBR (Master Boot Record). When installing the OS, you must use UEFI Boot from the DVD or Network so that the disk partition utility will create GPT partitions by default.

In this example, we will install Red Hat Enterprise Linux 6 on a RAID-bootable virtual disk across 4 PCIe-SSD drives:

1. During system post, press F11 and select UEFI Boot (if not already set by default). On the UEFI boot menu, the SAS/SATA drives will be listed as boot devices, but the PCIe-SSD drives will not be listed until after GRUB has been installed. After selecting the DVD or Network device to install from, the system will automatically boot the installation kernel. With RHEL 6 SP1 and SP2 you will require a

driver disk for the PCIe-SSD drives, so be sure to add 'dd' to the install boot command line and provide the appropriate driver disk.

2. Partition the disks and create file systems. You will have to change the default partition configuration suggested by the OS installer. See table below for a partitioning guideline. The partition configuration below is for each disk:

Partition	Recommended Size	File System	File System Type	Supported RAID Level
/dev/rssdX1	50 MB (minimum)	/boot/efi	EFI (vfat)	File system can't be on RAID virtual disk
/dev/rssdX2	100 MB	/boot	ext4	RAID 1
/dev/rssdX3	Depends on RAM	swap	swap	Any
/dev/rssdX4	Rest of disk	/	ext4	Any

Table 2 – Partitioning guideline

Note: The PCIe-SSD disks are /dev/rssda, /dev/rssdb, /dev/rssdc and /dev/rssdd. So replace X above with 'a', 'b', 'c' and 'd'.

3. /boot should be installed on a separate partition because the only RAID level that /boot can be created on is RAID 1 (this limitation is in GRUB 0.97 and is addressed in GRUB 2). Additionally, it is standard practice to create a separate partition for /boot.

4. /boot/efi contains an EFI boot binary and is required to boot in UEFI mode. Since all four disks will be bootable disks (not required, but recommended practice), there needs to be an EFI partition on each disk. However, because the OS installer will only let us create one EFI partition (which we create in /dev/rssda), we create vfat partitions (which have the same format as EFI) on the other 3 disks.

5. After installation of the OS is complete but before rebooting we have to copy the EFI partition from /dev/rssda to the other 3 disks. Switch to a virtual console (Ctrl+Alt+F2) and run the following for /dev/rssdb, /dev/rssdc and /dev/rssdd (replace X below accordingly):

```
# parted -s /dev/rssdX toggle 1 boot
# dd if=/dev/rssda1 of=/dev/rssdX1
```

6. Reboot the system. During system post, press F11 and select UEFI Boot (if not already set by default). All of the PCIe-SSD disks should now show up as UEFI bootable devices. You can select any of the four disks to boot from. The names in the UEFI Boot menu will be something like:

*** PCIe Solid State Drive in Slot X in Bay Y: Red Hat Enterprise Linux**

7. If you need to modify the GRUB configuration file, on UEFI systems it is located in `/boot/efi/EFI/redhat/grub.conf.`

8. Conclusion

As you can see, managing software RAID volumes with Express Flash PCIe-SSD devices is no different than with other types of disks. Because of its high-performance characteristics and targeted use case scenarios, software RAID volumes may not always be the best solution. Nonetheless, Linux software RAID is a viable solution for deployments seeking high availability capabilities.

Once again, please note that booting from PCIe-SSD devices is currently not supported by Dell. In addition, it is unclear under which use cases it would be advantageous to boot off these high-performance devices. The instructions in this document are for illustration purposes only and not necessarily supported by Dell.