Configuring and Managing Software RAID with

Red Hat Enterprise Linux 3

Thanks to dramatic advances in processing power, software RAID implementations are now a viable alternative to hardware-based RAID. By providing a mature software RAID layer and several management tools, the Red Hat® Enterprise Linux® 3 operating system can help system administrators build effective, cost-efficient software RAID implementations.

BY JOHN HULL AND STEVE BOLEY

Hardware-based RAID controllers have long been the preferred method for implementing RAID storage because they offload the management of RAID arrays onto a separate processor, freeing precious system CPU cycles for other tasks. However, the price/performance ratio of CPUs has decreased, making software-based RAID a viable alternative for system administrators in Linux®-based environments. The 2.4 kernel of the Linux operating system (OS) and its mature software RAID layer and management tools, particularly in Red Hat® Enterprise Linux 3, enable administrators to build an inexpensive RAID implementation.

Understanding software RAID in Linux environments

The Linux 2.4 kernel provides software-based RAID through the md device-driver layer, which sits on top of the storage controller device drivers. Because it is device-independent, the md device-driver layer, or *Linux RAID layer*, can work with all types of storage devices including SCSI and IDE. RAID-0 (striping), RAID-1 (mirroring), and RAID-5 (striping with parity) are supported in this kernel-level RAID implementation. Device nodes for md are denoted as /dev/md*x*, where *x* is a number from 0 to 15.

For software RAID, the `Linux raid autodetect` partition type is fd, which is an ID in the same manner that 83 is the type for ext3 partitions, 8e is the type for Logical Volume Manager (LVM) partitions, and 82 is the type for Linux swap partitions. When the Linux kernel boots, it automatically detects fd partitions as RAID partitions and starts the RAID devices. All kinds of partition types can be used with the Linux OS, including FAT16, FAT32, and even IBM® AIX® partitions, but fd is preferred because it does not require system administrators to start the RAID devices manually during boot as other partition types do.

The most useful RAID levels for enterprise storage usually are RAID-1 and RAID-5. Although this article focuses on creating and managing RAID-1 arrays in a Linux environment, much of the information also is applicable to RAID-5.

Creating RAID arrays during Linux OS installation

The easiest and most reliable method for configuring software RAID occurs during a new OS installation. For Red Hat Enterprise Linux 3, the Disk Druid tool provides a simple interface to define and create software RAID

configurations. When creating software RAID partitions and RAID-1 md devices, administrators must ensure that the system is configured correctly. Common best practices include:

- Create partitions that will correspond to the same sd*xy* device in the same order on each hard drive, where *x* changes with each drive but *y* remains the same for ease of administration in case of a drive failure. For example, sda1 and sdb1 (the first partitions on hard drives sda and sdb) both have a size of 100 MB. When administrators tag them with the fd partition type during installation, after creating a RAID device, these partitions become parts of device md0.
- Define precisely where each partition will reside, instead of letting Disk Druid determine the disk location. Disk Druid sometimes scatters partitions across disks, which can create problems for administrators if a disk must be replaced and the partitions rebuilt. To help determine where partitions reside, administrators can designate certain partitions as primary, which can help keep sda1 through sda3 and sdb1 through sdb3 aligned as md0, md1, and md2. This practice can help ease administration and recoverability.
- After creating matching partitions, create the md device for those partitions before defining the next set of partitions and md devices. This practice enables administrators to keep track more easily of which partitions match each other.
- Mirror all the partitions on the hard drives—including /boot, /swap, and so forth—to perform true RAID-1 mirroring. This practice helps ensure that all data on the system is backed up and can be restored if one drive fails.

Administrators then should complete the following steps in Disk Druid to create each software RAID device:

1. To create a software RAID partition, click the RAID button and then select "Create a software RAID partition." For the file system type, select "software RAID."
2. Ensure that only one drive is selected for the partition (for example, "sda" or "hda"), and make the desired configuration selections.
3. Repeat steps 1 and 2 but select the second hard drive (for example, "sdb" or "hdb") on which to create a RAID partition.
4. Click the RAID button again and select "Create a RAID device" when prompted. Choose the mount point, file system type, RAID device, and RAID level for this set of RAID partitions.

Prepping the system for drive failure

After configuring the RAID devices and installing the OS, administrators should prepare the system so that the RAID configuration can easily be restored to a failed drive. This process involves making a backup copy of the partitioning scheme on each drive and installing GRUB (GRand Unified Bootloader) on the Master Boot Record (MBR) of each drive.

The price/performance ratio of CPUs has decreased, making software-based RAID a viable alternative for system administrators in Linux-based environments.

By keeping backup copies of drive partition tables, administrators can quickly restore an original partition table on a replacement drive and avoid having to edit configuration files or re-create partitions manually with the fdisk utility. To copy a partition table, administrators should create a directory in which to store the partition information, and then use the `sfdisk` command to write partition information files for each disk into that directory:

```
mkdir /raidinfo
sfdisk -d /dev/sda > /raidinfo/partitions.sda
    (or hda for IDE drives)
sfdisk -d /dev/sdb > /raidinfo/partitions.sdb
    (or hdb for IDE drives)
```

During the RAID configuration and OS installation process, the installer mechanism places GRUB on the MBR of the primary hard drive only ("sda" or "hda"). However, if the primary disk drive fails, the system can be booted only by using a boot disk. To avoid this problem, administrators should install GRUB on the MBR of each drive.

To enter the GRUB shell, type `grub` at the command prompt. Next, at the `grub>` prompt, type `find /grub/stage1`. The subsequent output will specify where the GRUB setup files are located. For example:

```
(hd0,0)
(hd1,0)
```

The output lists the locations of root for GRUB, which is GRUB syntax for where the /boot partition is located. The Red Hat Linux OS specifically mounts the /boot partition as the root partition for GRUB. In the example output, sda is hd0 and sdb is hd1 (these refer to SCSI drives; for IDE drives, hda is hd0 and hdb is hd1). The second number specifies the partition number, where 0 is the first partition, 1 is the second partition, and so on. Thus, assuming SCSI disk drives, `(hd0,0)` signifies that the /boot partition resides on the first partition of sda (sda1); `(hd1,0)` refers to /boot residing on the first partition of sdb (sdb1).

Next, administrators should install GRUB on the MBR of the secondary RAID drive, so that if the primary drive fails, the next drive

has an MBR with GRUB ready to boot. When booting, the BIOS will scan the primary drive for an MBR and active partitions. If the BIOS finds them, it will boot to that drive; if not, it will go on to the secondary drive. Therefore, multiple drives in a system can have MBRs and active partitions, and the system will not have problems booting.

The 2.4 kernel of the Linux OS and its mature software RAID layer and management tools enable administrators to build an inexpensive RAID implementation.

To install GRUB on the MBR of the secondary drive, administrators must temporarily define the secondary drive as the primary disk. To do so, administrators identify sdb (or hdb) as hd0, and instruct GRUB to write the MBR to it by typing the following at the prompt `grub>`:

```
device (hd0) /dev/sdb (or /dev/hdb for IDE drives)
root (hd0,0)
setup (hd0)
```

GRUB will echo all the commands it runs in the background of the `setup` command to the screen, and then will return a message that the `setup` command succeeded. Both drives now have an MBR, and the system can boot off either drive.

Identifying important Linux software RAID administration tools

After the system with software RAID is ready for production, several useful software RAID files and management utilities can help administrators manage the RAID devices:

- **/etc/raidtab:** This file contains information about the system's software RAID configuration, including which block devices belong to which md device. It can help administrators determine which RAID configuration the kernel expects to find on the system.
- **/proc/mdstat:** This file shows the real-time status of the md devices on the system, including online and offline partitions for each device. When rebuilding RAID partitions, this file also shows the status of that process.

In addition, the Red Hat Enterprise Linux raidtools package provides several useful tools:

- `lsraid`**:** This command-line tool allows administrators to list and query md devices in multiple ways. It presents much of the same information as /etc/raidtab and /proc/mdstat. Administrators can view this tool's man page for more information.
- `raidhotadd`**:** This command-line utility allows administrators to add disk partitions to an md device and to rebuild the data on that partition.
- `raidhotremove`**:** This command-line utility allows administrators to remove disk partitions from an md device.

The next section demonstrates some key uses for these files and tools.

Restoring the RAID configuration after drive failure

When a drive in a RAID-1 array fails, administrators can restore the RAID array onto a new drive by following a three-step process: replace the failed drive, partition the replacement drive, and add the RAID partitions back into the md devices.

Replacing a disk drive

Once a hard disk drive fails, it must be replaced immediately to preserve the data redundancy that RAID-1 provides. The method by which the drive is replaced depends on the type of disk drives in the system. Because hot plugging of IDE drives is not supported in the Linux 2.4 kernel, administrators must replace an IDE drive by shutting down the system, swapping the drive, and then rebooting. However, the Linux device drivers for the Adaptec® and LSI Logic® SCSI controllers that ship on Dell™ PowerEdge™ servers do support hot plugging of drives, so administrators can replace SCSI drives while the system is still running.

To hot plug a SCSI disk drive, administrators first must disable the drive in the kernel, then physically replace the drive, and finally enable the new drive in the kernel. To disable a SCSI drive, administrators echo the device out of the real-time /proc file system within Linux, and the system instructs the corresponding drive to "spin down" and stop operating (for example, a 10,000 rpm drive would go from a speed of 10,000 rpm to 0 rpm). Conversely, to enable a SCSI drive, administrators echo the device into the real-time /proc file system, and the system instructs the drive to "spin up" to operating speed (using the previous example, the drive would go from 0 rpm to 10,000 rpm).

To obtain the syntax to pass to the /proc file system, administrators can type `cat /proc/scsi/scsi` at the command prompt. This command will provide a list of all SCSI devices detected by the kernel at the moment the command was received. For example, suppose a system has two SCSI disk drives, and the drive with SCSI ID 1 fails and must be replaced. The output of the `/proc/scsi/scsi` command would be:

```
Host:  scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: Seagate Model: . . .
Host:  scsi0 Channel: 00 Id: 01 Lun: 00
  Vendor: Seagate Model: . . .
```

To disable the drive in the kernel, the administrator would type the following command:

```
echo "scsi remove-single-device" 0 0 1 0 >
   /proc/scsi/scsi
```

The administrator then would receive a message stating that the system is spinning down the drive. Another message is sent when this process is complete, after which the administrator can remove the failed drive and replace it with a new one. To enable the new drive in the kernel, the administrator must spin it back up:

```
echo "scsi add-single-device" 0 0 1 0 >
   /proc/scsi/scsi
```

After this process completes, the kernel is ready to use the drive.

Partitioning the replacement drive

Once the failed disk drive has been replaced, administrators must restore the partitions that were saved earlier in the /raidinfo directory. For example, if replacing drive sdb, the administrator would issue the following command to restore the original partition scheme for sdb to the new drive:

```
sfdisk /dev/sdb < /raidinfo/partitions.sdb
```

Adding the RAID partitions back into the md device

Next, the administrator adds the partitions back into each RAID device. The /proc/mdstat file displays the status of each RAID device. For example, a system that is missing a partition from the md0 device would show the following:

```
md0 : active raid1 sda1[0]
      40064 blocks [2/1] [U_]
```

This output indicates that md0 is active as a RAID-1 device and that partition sda1 is currently active in that RAID device. However, it also shows that the second partition is not available to the device, as denoted by the following information: the first line does not list a second partition; the output `[2/1]` denotes that two partitions should be available to the device (the first value), but only one is currently available (the second value); and the output `[U_]` shows that the second partition is offline.

To add partition sdb1 back into the md0 device and to rebuild the data on that partition, administrators use the following command:

```
raidhotadd /dev/md0 /dev/sdb1
```

While the partition is rebuilding, administrators can track the status by periodically viewing /proc/mdstat, which displays the percentage of rebuilding that is complete. Once the rebuilding is finished, /proc/mdstat would show the following output for the example device:

```
md0 : active raid1] sda1[0] sdb1[1]
      40064 blocks [2/2] [UU]
```

Administrators must complete the `raidhotadd` command to add each partition back into its respective RAID device. Once the failed drive has been replaced, administrators simply run the GRUB commands discussed in "Prepping the system for drive failure" to install GRUB on the MBR of the new disk. After this step, the RAID configuration will be fully restored. These functions can easily be placed into a script. Then, from a single executable point, administrators can complete all rebuild functions—making software RAID more palatable by easing drive administration.

Building cost-efficient RAID in Linux

The increasing cost-effectiveness of software RAID offers Linux system administrators an alternative to more expensive hardware-based RAID implementations, thanks to the performance and cost advantages of the Linux OS and rapid advancements in processor power. Using the management tools available in Red Hat Enterprise Linux 3, administrators can create RAID implementations that best suit their data center requirements.

John Hull (john_hull@dell.com) is a software engineer at Dell and is currently the lead Linux engineer for Dell Precision™ workstations.

Steve Boley (steve_boley@dell.com) is a Gold Server Support senior network engineer at Dell. He provides hardware and software support for U.S.-based customers, with an emphasis on Linux. Steve is a Microsoft® Certified Systems Engineer (MCSE) and a Red Hat Certified Engineer.

FOR MORE INFORMATION

Dell and Linux:
http://www.dell.com/linux

Nadon, Robert and Thomas Luo. "Implementing Software RAID on Dell PowerEdge Servers." *Dell Power Solutions*, August 2003.
http://www1.us.dell.com/content/topics/global.aspx/power/en/ps3q03_nadon?c=us&cs=555&l=en&s=biz