

Expanding Storage on Linux-based Servers

In response to growing user demands for storage, system administrators frequently must add additional storage space to existing file servers. This article describes how to add and configure local storage space on a file server running the Red Hat® Linux® operating system (OS). Three scenarios are presented: adding disks without a RAID controller, adding disks to a new logical drive on a RAID controller, and adding disks to an existing logical drive on a RAID controller.

BY MATT DOMSCH AND TEFAMARIAM MICHAEL

As enterprises expand, so do their storage needs. System administrators often must adjust existing infrastructures to meet these needs. This article describes how to expand local storage space—as opposed to remote storage such as network attached storage (NAS) or a storage area network (SAN)—on a file server running the Red Hat® Linux® operating system (OS). Although this article primarily focuses on systems that have a Dell™ PowerEdge™ Expandable RAID Controller (PERC), it also provides the necessary steps to accomplish this goal on systems without RAID controllers.

Understanding the process of storage expansion

Administrators must consider several conceptual layers when expanding Linux-based storage. Although these layers can vary depending on the system hardware, a system will

include a minimum of three layers: the physical disk drive, the partition, and the file system (see Figure 1). If the system includes hardware RAID, an additional layer exists—the logical drive (see Figure 2). PERC, as a hardware RAID controller, combines multiple physical disk drives and presents them as a single storage unit. This combined single storage unit is referred to as a logical drive.

Before trying to expand storage space, system administrators should perform a full system backup. Whether simply adding new storage or expanding existing logical disks, partitions, and file systems, administrators should start at the bottommost layer—adding physical disks, either to internal disk bays or to an existing or new Dell PowerVault™ disk enclosure. Internal disk backplanes are typically cabled to on-motherboard disk controllers;

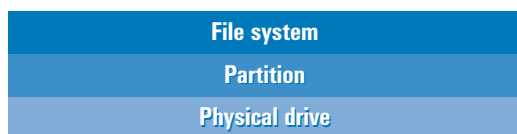


Figure 1. Layers on a system without PERC

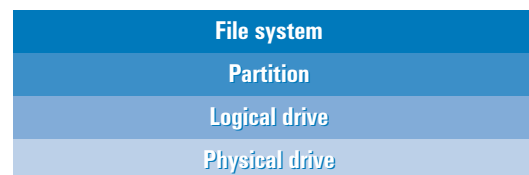


Figure 2. Layers on a system with PERC

Controller	Embedded RAID or add-in PCI controller	OEM	Linux tool	Package to download
PERC 4/Dual Channel Integrated (Di)	Embedded in PowerEdge 2600 server	LSI Logic	dellmgr	megaraid-util-rh7.3-i386.tar.gz
PERC 3/Quad Channel (QC)	Add-in PCI controller			
PERC 3/Dual Channel (DC)	Add-in PCI controller			
PERC 3/Dual Channel Light (DCL)	Add-in PCI controller			
PERC 2/DC	Add-in PCI controller			
PERC 2/Single Channel (SC)	Add-in PCI controller			
PERC 3/Di	Embedded in PowerEdge 1650, PowerEdge 2500, PowerEdge 2550, PowerEdge 2650, PowerEdge 4400, and PowerEdge 4600 servers	Adaptec	afacli	aacraid-util-rh7.3-i386.tar.gz
PERC 3/Single Channel Integrated (Si)	Embedded in PowerEdge 2450 server			
PERC 2/Si	Embedded in PowerEdge 2400 server			
PERC 2	Add-in PCI controller			

Figure 3. Storage management tools for Dell RAID controllers

external PowerVault enclosures are typically cabled to add-in Peripheral Component Interconnect (PCI) cards. Often, these on-motherboard disk controllers and PCI cards have unused channels to which more enclosures may be cabled. Once all existing channels have been used, additional PCI cards with additional controller channels may be added.

After adding the physical disks, administrators using hardware RAID must configure new or expanded logical drives to encompass the additional physical disks. The process for adding or expanding a logical disk differs based on the type of RAID controller (see the “Obtaining the appropriate RAID management software” section of this article).

Next, administrators must create the partition table, which the OS uses to slice disks into logical segments. Traditional tools such as `fdisk`, as well as newer tools such as GNU Parted, can be used to create and modify the partition table according to system needs.

After creating the partitions, administrators must create the file system in each partition. Administrators can create the Linux ext3 file system, which is the default Red Hat Linux file system; likewise, on each partition they can create any other file system, including the ext2 file system, ReiserFS, or IBM® journaling file system (JFS).

Finally, administrators must mount the new file system at a directory in the file system hierarchy so that it is visible to users.

Obtaining the appropriate RAID management software

LSI Logic and Adaptec are the original equipment manufacturers (OEMs) for the Dell-supported RAID controllers—the PERC line of products. When using any of the PERC products, administrators can create new logical drives and modify existing logical drives. The tools used to accomplish these tasks are OEM-specific. Figure 3 lists these tools by PERC type, PowerEdge server type, and OEM.

Depending on the PERC, administrators should download either `megaraid-util-rh7.3-i386.tar.gz` or `aacraid-util-rh7.3-i386.tar.gz` from <http://support.dell.com>. After downloading these packages, they should extract them using the following commands: `tar xvzf megaraid-util-rh7.3-i386.tar.gz` (for an LSI Logic™-supplied PERC) or `tar xvzf aacraid-util-rh7.3-i386.tar.gz` (for an Adaptec®-supplied PERC). Administrators should then install either `dellmgr`, `afaapps`, or both RPM™ (Red Hat Package Manager) packages by running the commands `rpm -ivh dellmgr-5.22.0-i386.rpm` (LSI Logic) or `rpm -ivh afaapps-2.6.1-i386.rpm` (Adaptec). If the OS was factory installed by Dell, these RPM packages should already be installed on the PowerEdge server.

Expanding storage step by step

Figure 4 shows the steps to expand storage for three scenarios:

- **Scenario A:** Adding SCSI drives to a system in which no PERC is involved and the added drives are connected to a SCSI adapter, such as the Adaptec SCSI Card 39160, or to an on-board SCSI controller
- **Scenario B:** Adding a new logical drive to a PERC
- **Scenario C:** Expanding an existing logical drive on a PERC

Step 1: Add physical drives (Scenarios A, B, and C)

Add physical drives to the storage enclosure, which may be the system disk backplane, a media bay, or an external storage enclosure such as the PowerVault 220S. If additional disk slots are required, connect more PowerVault enclosures to the server using existing SCSI channels or additional SCSI or PERC add-in PCI cards.

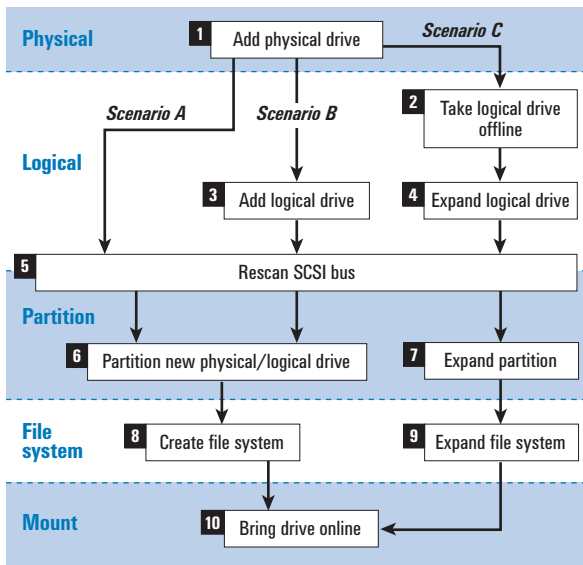


Figure 4. Storage expansion process for three scenarios

Step 2: Take the logical drive offline (Scenario C only)

Because file system size changes cannot be made while the file system is in use, the logical drive should be taken offline. Assuming that the drive is mounted on /data, take it offline using the command `umount /data`.

Step 3: Add a logical drive (Scenario B only)

The process for adding a logical drive to a PERC is OEM-specific. Based on the controller type, follow these instructions:

If the controller is an LSI Logic-supplied PERC. Change to text mode by pressing CTRL + ALT + F1 and log in as root. From the command line, start `dellmgr`, which should bring up a text user interface (TUI). This TUI screen is similar to the one found in the controller’s BIOS configuration utility (accessible by pressing CTRL + M) during system boot. From this TUI, select “Select Adapter” and choose the adapter to which the logical drive should be added (if the system has only one adapter, this option is not available). Select “Configure” and then “View/Add Configuration.” Choose the added hard drives and create the logical drive.

If the controller is an Adaptec-supplied PERC. From a command line, start `afacli`. At the `afacli` command-line interface (CLI), use the commands shown in Figure 5.

Step 4: Expand the logical drive (Scenario C only)

Similar to step 3, the process for expanding an existing logical drive on a PERC is OEM-specific. Based on the controller type, follow these instructions:

If the controller is an LSI Logic-supplied PERC. Change to text mode by pressing CTRL + ALT + F1 and log in as root. From the command line, start `dellmgr`, which should bring up a TUI. From

this TUI: If the system has more than one of these controllers, select “Select Adapter” and choose the adapter to modify. Select “Reconstruct” and choose the logical drive that will be expanded. Drives available to be added should be displayed. Determine which drives will be added by selecting the drive name and then pressing the spacebar. Next, press the enter key, select the RAID level, and select “Reconstruct the logical drive” to start the reconstruction process.

If the controller is an Adaptec-supplied PERC. From a command line, start `afacli`. At the `afacli` CLI, use the commands shown in Figure 6.

Please note that expanding a logical drive may take several hours because the controller must redistribute the data across the new disks. However, this redistribution occurs as a background operation so the system can be used during reconstruction.

For additional information about the tools used in steps 3 and 4 (`dellmgr` and `afacli`), consult the user guide documents found at <http://support.dell.com>.

Step 5: Rescan the SCSI bus (Scenarios A, B, and C)

After changing the physical and logical drive layers, notify the OS of the changes. Rebooting the system is the easiest way to do so—the new disks will be detected at system startup—but Linux-based systems can notify the OS without rebooting.

If an existing logical drive was expanded, tell the OS to remove the drive and then add it back to the OS. The `/proc/scsi/scsi` file

Command	Description
<code>open afa0</code>	Opens connection with the PERC (afa0 represents the first PERC)
<code>controller rescan</code>	Detects the added hard drives
<code>disk list</code>	Displays the bus, ID, and logical unit number (LUN) of the drives; also shows the initialization status of drives (those that are not initialized at this point should be initialized)
<code>disk initialize (bus, id, lun)</code>	Initializes the drive in the given bus, ID, and LUN
<code>container list</code>	Displays all the logical drives in the controller
<code>disk show space</code>	Shows the disk space
<code>container create raid5 ((bus, id1, lun), disk_size) (bus, id2, lun) (bus, id3, lun)</code>	Creates RAID-5 using the specified drives with <i>disk_size</i> space from each drive. For example, <code>container create raid5 ((1,0,0), 33000M) (1,1,0) (1,2,0) (1,3,0)</code> will create a new logical drive across the first 33 GB of disks 1 through 4. To learn how to create other RAID levels and options, issue <code>container create help</code>
<code>container rescan</code>	Adds the newly created RAID to <code>/proc/scsi/scsi</code>
<code>close afa0</code>	Closes the connection to the PERC
<code>exit</code>	Quits out of <code>afacli</code>

Figure 5. Adding a logical drive to an Adaptec-supplied PERC

Command	Description
open afa0	Opens connection with the PERC (afa0 represents the first PERC)
controller rescan	Detects the added hard drives
disk list	Displays the bus, ID, and LUN of the drives; also shows the initialization status of drives (those that are not initialized at this point should be initialized)
disk initialize (bus, id, lun)	Initializes the drive in the given bus, ID, and LUN
container list	Displays all the logical drives in the controller
container reconfigure /partition_size=drive_size logical_drive_id (bus, id, lun)	Adds the given drive to the specified logical drive. For example, container reconfigure /partition_size=33000M 1 (1,4,0) adds 33 GB of disk 4 to logical drive 1
container rescan	Adds the newly created RAID to /proc/scsi/scsi
close afa0	Closes the connection to the PERC
exit	Quits out of afacli

Figure 6. Expanding a logical drive in an Adaptec-supplied PERC

lists all SCSI devices currently known to the OS, and when written to, this file informs the OS of changes. The following command adds a drive to the OS:

```
echo "scsi add-single-device Host Bus Id Lun" >
    /proc/scsi/scsi
```

(where *Host*, *Bus*, *Id*, and *Lun* correspond to the new drive parameters)

Similarly, the following command removes a drive from the OS:

```
echo "scsi remove-single-device Host Bus Id Lun" >
    /proc/scsi/scsi
```

For example, the command `cat /proc/scsi/scsi` generates the output shown in Figure 7.

This output in Figure 7 shows two logical drives. The first drive is on an Adaptec RAID controller at SCSI host 0, channel or bus 0, ID 0, LUN 0. The second drive is on an LSI Logic RAID controller

```
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: DELL      Model: PERCRAID      Rev: 2.7
  Type:   Direct-Access      ANSI SCSI revision: 03
Host: scsi1 Channel: 00 Id: 00 Lun: 00
  Vendor: MegaRAID  Model: LD 0 RAID5 42G Rev: 1.75
  Type:   Direct-Access      ANSI SCSI revision: 03
```

Figure 7. Example data within the /proc/scsi/scsi file

at SCSI host 1, channel or bus 0, ID 0, LUN 0. If the size of the logical drive on the LSI Logic RAID controller changed, administrators would issue the following commands to remove the drive from the OS and add it back with the new size:

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

If a new physical or logical drive was added, tell the OS where to find it. For the example system used in Figure 7, administrators would issue the following command after adding a new logical drive to the LSI Logic RAID controller:

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

Note that the value in the ID position was incremented to match the new logical drive number. Similarly, administrators can inform the OS of newly added physical drives not attached to RAID controllers by determining the parameters of a new drive. To do so, note the parameters of existing disks on the same controller and increment the ID value accordingly.

When adding new devices, watch the messages returned by the kernel in `/var/log/messages`. These messages display the SCSI device names (sdb, sdc, and so forth) to use for further operations.

Step 6: Create partitions (Scenarios A and B)

Partition the new physical drive (Scenario A) or the new logical drive (Scenario B). Like all other operating systems, Linux uses disk partitions to logically divide large areas of storage (physical or logical drives) into smaller pieces. Each partition contains its own file system. For the purposes of this article, the logical drive has only a single partition, although multiple smaller partitions are possible.

The `fdisk` utility is the traditional partitioning tool with which most system administrators are familiar. A newer partitioning tool, GNU Parted, includes the functionality of `fdisk`, plus the ability to

Command	Description
<code>parted /dev/sdb print</code>	Displays the size of disk <code>/dev/sdb</code> and the partition table (if any) on disk <code>/dev/sdb</code>
<code>parted /dev/sdb mklabel msdos</code>	Creates a new empty MS-DOS-style partition table on the disk
<code>parted /dev/sdb mkpart p ext2 0 end_size</code>	Creates a new primary partition ready for an ext2 or ext3 file system, starting at 0 MB and expanding to <code>end_size</code> MB

Figure 8. Sample GNU Parted commands for displaying and creating disk partitions

expand file systems when resizing the underlying partition—all in one step. Use either tool to create the desired partitions on the new logical or physical drive. Figure 8 shows sample GNU Parted commands for displaying and creating partitions.

Step 7: Expand the partition (Scenario C only)

Use GNU Parted to expand the partition. Figure 9 shows sample GNU Parted commands to expand partitions.

Step 8: Create file systems on the new partitions (Scenarios A and B)

After partitioning, create or expand the partition's file system. Linux supports multiple file system types. Use the `mke2fs` tool to create the most commonly used file systems on Linux: ext2 and ext3. The command `mke2fs -j /dev/sdb1` creates a new ext3 file system on partition `sdb1`; `mke2fs /dev/sdb1` creates a new ext2 file system on partition `sdb1`.

Step 9: Expand the file system (Scenario C only)

Use GNU Parted to expand the file system. Or, use `resize2fs` to expand the file system once the partition has been expanded. Figure 9 shows sample GNU Parted and `resize2fs` commands to expand file systems.

Step 10: Bring the drive online (Scenarios A, B, and C)

The system is now ready to use the added or modified storage. Bring the storage online by mounting it. For example, the command `mount /dev/sdb1 /data` attaches the file system `/dev/sdb1` to the directory `/data`. Use tools such as `df` to verify that the storage has expanded. For example, `df /data` displays disk space statistics for `/data`.

Considering other steps for storage management


Ongoing work in the Linux community will enhance the storage management process. Administrators can use an additional abstraction layer—the Logical Volume Manager (LVM)—to group and redistribute disparate physical or logical drives into a single expandable logical volume. Two open source LVM projects are in development: the Sistina® Logical Volume Manager 2 (LVM2) and the IBM Enterprise Volume Management System (EVMS). Commercial products from VERITAS also may be used in this capacity.

Command	Description
<code>parted /dev/sdb print</code>	Displays the current partition on disk <code>/dev/sdb</code>
<code>parted /dev/sdb resize partition_number start end</code>	Changes the size of the partition with the number <code>partition_number</code> on disk <code>/dev/sdb</code> ; the new partition starts <code>start</code> MB from the beginning of the disk to <code>end</code> MB from the beginning of the disk (the <code>start</code> and <code>end</code> values can be obtained from the print switch mentioned above)
<code>resize2fs /dev/sdb1</code>	Expands the file system to the end of the <code>/dev/sdb1</code> partition; <code>resize2fs</code> is needed only if using <code>fdisk</code> (GNU Parted will perform this task automatically)

Figure 9. Sample GNU Parted and `resize2fs` commands for expanding partitions and file systems

One drawback to the approach presented in this article is that the file system being modified must be taken offline before being changed, which means visible downtime for end users. Developers are working on the ext3 file system so that it can be expanded online without first being unmounted, thus eliminating this visible downtime.

Expanding storage, expanding opportunities

The three storage expansion scenarios presented in this article, featuring both RAID and non-RAID configurations, present system administrators with several options for adding storage capacity to Red Hat Linux-based file servers. By adding space at the physical disk and logical disk layers, changing partition sizes, and growing file systems, organizations can accommodate the growing appetite for storage. 

Matt Domsch (matt_domsch@dell.com) is a lead and senior engineer in the Linux Development Team of the Dell Product Group, which tests Linux on all Dell PowerEdge servers. Matt has an M.S. in Computer Science from Vanderbilt University and a B.S. in Computer Science and Engineering from the Massachusetts Institute of Technology. His primary areas of interest include networking and operating systems.

Tesfamariam Michael (tesfamariam_michael@dell.com) is a software engineer in the Linux Development Team of the Dell Product Group, which tests Linux on all Dell PowerEdge servers. Tesfamariam has an M.S. in Computer Science from Clark Atlanta University, a B.S. in Electrical Engineering from the Georgia Institute of Technology, and a B.S. in Mathematics from Clark Atlanta University. His areas of interest include operating systems and I/O devices.

FOR MORE INFORMATION

Documentation for `dellmgr` and `afacli`: <http://support.dell.com>

GNU Parted: <http://www.gnu.org/software/parted>

Sistina LVM2: http://www.sistina.com/products_lvm.htm

IBM EVMS: <http://evms.sourceforge.net/>