Logical Volume Manager

Layne Churchill
CD/CMS
Fermilab

June 13, 2002

Abstract

This paper discusses the Logical Volume Manager (LVM) version 1.0.3 for Linux. The paper provides discussion on some of LVM's features, their implementation and their performance. This document does not discuss in detail all of the LVM commands and their options. It provides only a brief discussion on setting up LVM.

Contents

1 Introduction
- 1.1 What is LVM?
- 1.2 What are the advantages to using LVM?
- 1.3 Description of test system

2 General Set-up
- 2.1 Getting LVM
- 2.2 Setting up volume groups
- 2.3 Creating logical volumes
- 2.4 Creating the file system and mounting the logical volume

3 Features
- 3.1 Striping
- 3.2 Snapshots

4 More Information

5 Conclusions

1 Introduction

1.1 What is LVM?

LVM is a software tool that offers system administrators greater flexibility in organizing hard disk resources. Physical devices (partitions, disks, and RAID arrays) are arranged into "volume groups" (VG), which then are divided into "logical volumes" (LV). Logical volumes are similar to disk partitions in that they contain the file system, and they are mounted into the directory tree just as regular partitions are.

1.2 What are the advantages to using LVM?

As previously mentioned, LVM offers administrators great flexibility in organizing and maintaining hard disk resources. LVM includes tools that allow an administrator to resize logical volumes, implement striping across two or more physical devices, copy data from one volume to another, and make frozen, read-only images (called snapshots) of logical volumes for backup purposes. Physical devices can be installed and added to existing volume groups. All of this can be done on-the-fly, without interfering with users.

1.3 Description of test system

This document is based on tests that were run using the LVM system packaged with Red Hat Linux 7.3. The tests were done using two 60 GB Maxtor disk drives (Ultra ATA 100, 5400 RPM, 2MB, 8.9ms) connected to a Promise RAID controller. No array was defined however.

2 General Set-up

2.1 Getting LVM

Instructions for downloading and building LVM can be found at Sistina Software's LVM web site, http://www.sistina.com/products_lvm.htm .

2.2 Setting up volume groups

In order to create a volume group, the physical devices that are to be included in the group must first be initialized. This is done with the *pvcreate* command. This command creates a descriptor area at the start of the physical device. The descriptor area is where information regarding the volume group to which the device is associated is located.

To create a descriptor area on a hard disk, for instance /dev/hda, type:

```
# pvcreate /dev/hda
```

To create a descriptor area on a partition, for instance the first partition on /dev/hda, first change the partition type to 0x8e (Linux LVM) using fdisk, and then type:

```
# pvcreate /dev/hda1
```

After initializing the physical devices, use the *vgcreate* command to create the volume group.

```
# vgcreate newvg /dev/hda1 /dev/hdb1
```

This command creates a new volume group called *newvg*, and populates the descriptor areas on the two devices, /dev/hda1 and /dev/hdb1. A new directory, /dev/newvg, is created.

An important thing to note is that since the information regarding the volume group is stored on the disks themselves, the volume group itself is independent of physical addresses. If the computer was shut down, and the cables on these two drives were swapped, *newvg* would continue to function normally.

Once a volume group has been created, it must be activated. This requires the *vgchange* command. Type:

```
# vgchange -a y newvg
```

One can now create logical volumes in the volume group.

2.3 Creating logical volumes

Logical volumes are created using the *lvcreate* command. This command offers several useful options of which only a few will be discussed. The reader is encouraged to consult the command's man page for more information.

To create a logical volume in the volume group, type:

```
# lvcreate -L<size> -nlv1 newvg
```

This command creates a logical volume, *lv1*, in the volume group *newvg*, and a block special file, /dev/newvg/lv1. It is this file that will be later mounted into the directory tree.

The size is given in megabytes by default, but kilobytes or gigabytes can be specified by adding a 'k' or 'g' suffix to the value.

If striping is desired, type:

```
# lvcreate -L<size> -i2 -I4 -nlvstr newvg
```

This command would create a logical volume, *lvstr*, with two stripes and a stripe size of 4 KB. Stripe size can be given in megabytes or gigabytes by adding an 'm' or 'g' suffix to the value.

2.4 Creating the file system and mounting the logical volume

File systems are created on logical volumes just as they would be on normal disk partitions. Once the file system is created, the logical volume can be mounted into the directory tree in the same fashion as a normal disk partition. Both operations are done with the block special file created with the *lvcreate* command.

3 Features

3.1 Striping

Striping allows LVM to balance the disk I/O load over two or more physical devices. The concept is similar to that of RAID devices. Striping can result in a dramatic increase in I/O performance.

To test the increase in I/O performance, the benchmark tool bonnie++ was used. First, the two drives to be included in the volume group were examined. Both drives had an ext3 file system.

	Sequential Write		Sequential Read		Random Seeks	
	K/sec	%CPU	K/sec	%CPU	#/sec	%CPU
/dev/hde	33602	12	40930	5	91.2	0
/dev/hdg	34539	12	42042	5	91.4	0

Table 1: I/O Performance of Disk Drives

The drives were then initialized and arranged in a volume group. A logical volume spanning the entire volume group was created. The ext3 and ReiserFS file systems were each tested with and without striping. The results are shown in Tables 2 and 3 below.

	Sequential Write		Sequential Read		Random Seeks	
	K/sec	%CPU	K/sec	%CPU	#/sec	%CPU
ext3	33186	12	40568	6	90.8	0
ReiserFS	35506	21	38006	7	95.8	0

Table 2: I/O Performance of LVM w/o Striping

	Sequential Write		Sequential Read		Random Seeks	
	K/sec	%CPU	K/sec	%CPU	#/sec	%CPU
ext3	61741	25	81280	12	96.6	0
ReiserFS	60475	39	79521	15	103.2	0

Table 3: I/O Performance of LVM w/ Striping

It is clear from these tests that the striping feature of LVM offers a significant boost in disk I/O performance.

3.2 Snapshots

The snapshot feature in LVM is incredibly useful for backing up data stored on a logical volume. A snapshot is a virtual read-only copy of a logical volume which contains all of the data that was in the volume at the time the snapshot was created. The snapshot can be used to create a backup of the data without disrupting use of the logical volume, and without the worry that data will be changed during the backup process.

While the snapshot *appears* to be a complete read-only copy of the logical volume, this is not actually the case. When a snapshot of a logical volume is created, the logical volume itself is made read-only, and all changes to the volume are recorded in the space allocated for the snapshot. This is all completely transparent to the user, who can continue to read from and write to the directory where the logical volume is mounted, and to the administrator, who sees a complete read-only copy of the logical volume on the snapshot.

To create a snapshot of the logical volume *lv1* in the volume group *newvg*, type:

```
# lvcreate -L<size> -s -nlvbackup /dev/newvg/lv1
```

The snapshot must be mounted after it is created. Type:

```
# mkdir /mnt/lvbackup
# mount /dev/newvg/lvbackup /mnt/lvbackup
```

Now the backup can be made. For instance, one could type:

```
# tar -cf /dev/rmt0 /mnt/lvbackup
```

After the backup is complete, the snapshot must be unmounted and removed from the volume group using the *lvremove* command in order to return to normal operation.

It is of critical importance that the snapshot be made large enough to contain any changes that may be made while the snapshot exists. If the snapshot becomes full, it will cease to function, and the backup will fail. There must be enough space allocated to the snapshot to contain all new data written to the drive plus the accumulated amounts of overhead required for recording file creation, deletion, and editing.

4 More Information

More information on Logical Volume Management can be found at Sistina Software's LVM web site, http://www.sistina.com/products_lvm.htm . The "how to" documents contain a wealth of information about the LVM system and its commands and features.

5 Conclusions

The various features of LVM and the added flexibility of working with volume groups and logical volumes rather than disks and partitions more than compensates for the small investment in time and effort it takes to implement. The commands are intuitive and easy to use, and the system appears to be very reliable.