Contents

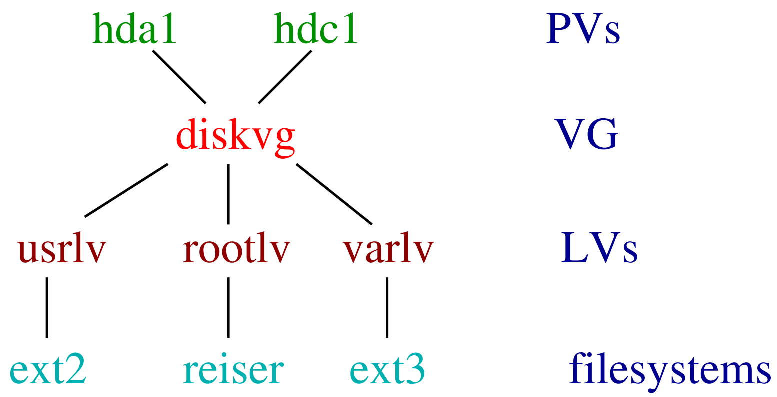
- LVM (Logical Volume Manager)
  - What is LVM (Logical Volume Manager) ?
  - Anatomy of LVM
  - Mapping modes (linear/striped)
  - Snapshots
  - Boot time scripts
  - LVM Files
  - LVM Commands
  - Examples
  - Kernel configuration

What is LVM (Logical Volume Manager) ?

- **LVM** provides a higher-level view of the disk storage on a computer system than the traditional view of disks and partitions.

- Storage volumes created under the control of the LVM can be resized and moved around almost at will.

Anatomy of LVM



VG (**V**olume **G**roup): gathers together LVs and PVs into one administrative unit.

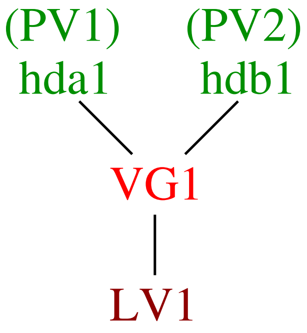
PV (**P**hysical **V**olume): a PV is typically a hard disk (partition), raid device, etc...

LV (**L**ogical **V**olume): the equivalent of a disk partition in a non-LVM system.

PE (**P**hysical **E**xtent): each PV is divided in chunks of data, known as PEs. These extents have the same size as the LEs.

LE (**L**ogical **E**xtent): each LV is split into chunks of data, known as LEs.

Mapping modes (linear/striped)   1/2


(PV1) (PV2)
hda1 hdb1
VG1
LV1

PV1 size = 99 extents, PV2 size = 248 extents

PE and LE = 4MB

VG

LVs

- We now can create a LV (LV1).
- LV1 can be any size between 1 and 347 (99+248) extents.
- When LV1 is created, a mapping is defined between LEs and PEs (eg. LE[i] could map onto PE[i] of PV1 and PE[i] of PV2).

Mapping modes (linear/striped) 2/2

Strategies for mapping LEs onto PEs:

1. **Linear** ($\simeq$ RAID Linear) mapping will assign a range of PEs to an area of an LV in order (eg., LE[1 - 99] map onto PV1 and LE[100 - 248] map onto PV2).

2. **Striped** ($\simeq$ RAID0):

   LE[1] $\rightarrow$ PV1[1],

   LE[2] $\rightarrow$ PV2[1],

   LE[3] $\rightarrow$ PV1[2],

   and so on.

Snapshots

- **Snapshots** allows the administrator to create a new block device which is an exact copy of a LV, frozen at some point in time.

- Used when for instance, we need to perform a backup on the LV, but you don't want to halt a live system that is changing the data.

Boot time scripts 1/3

- The **startup** of LVM requires just the following two commands:

```
# vgscan
vgscan – reading all physical volumes (this may take a while...)
vgscan – "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan – WARNING: This program does not do a VGDA backup of your volume group
```

```
# vgchange -ay
```

- And the **shutdown** only one:

```
# vgchange -an
```

Follow the next instructions depending on the Linux distribution you are running

Boot time scripts - Debian (lvm tool package) 2/3

Create a startup script in `/etc/init.d/lvm` containing the following:

```
#!/bin/sh
case "$1" in
  start)  /sbin/vgscan
          /sbin/vgchange -ay ;;
  stop) /sbin/vgchange -an ;;
  restart|force-reload)
          /sbin/vgchange -an && /sbin/vgscan \\
          && /sbin/vgchange -ay ;;
esac
exit 0
```

Boot time scripts - Redhat 3/3

1. Modify `/etc/rc.d/rc.sysinit` to load the LVM:

```
# LVM initialization
if [ -e /proc/lvm -a -x /sbin/vgchange -a -f /etc/lvmtab ]
then
     action $"Setting up Logical Volume Management:" \
     /sbin/vgscan && /sbin/vgchange -ay
fi
```

2. Edit the file `/etc/rc.d/init.d/halt` to deactivate the volume groups at shutdown.

   Insert the `vgchange -an` command near the end of this.

LVM Files

- `/etc/lvmtab`

  Informs about the activated VGs

- `/proc/lvm/*`

  Inform about the VGs structure

- `/dev/my_VG[1-n]/*`

  LVM device files

LVM Commands 1/7

- **Initializing** disks or disk partitions
  - For entire disks: `# pvcreate /dev/hdb`
  - For partitions: set the partition type to 0x8e using fdisk. Next:

    ```
    # pvcreate /dev/hdb1
    ```

    pvcreate – physical volume "/dev/hdb1" successfully created

- **Creating** a **VG**:

  ```
  # vgcreate my_VG /dev/hdb1 [/dev/hdc1] ...
  ```

  vgcreate – INFO: using default **physical extent size 4 MB *(default)***

  vgcreate – INFO: maximum logical volume size is 255.99 Gigabyte

  vgcreate – doing automatic backup of volume group "my_VG"

  vgcreate – volume group "my_VG" successfully created and activated

  ```
  # more /etc/lvmtab
  my_VG
  ```

LVM Commands 2/7

- **Activating** a **VG**: `# vgchange -ay my_VG`

  Note: in the startup script

- **Removing** a **VG**:

  1. Deactivate the VG: `# vgchange -an my_VG`
  2. Remove the VG: `# vgremove my_VG`

- **Adding PV**s to a **VG**:

  ```
  # pvcreate /dev/hdb2
  ```

  ```
  # vgextend my_VG /dev/hdb2
  ```

- **Removing PV**s from a **VG**:

  ```
  # vgreduce my_VG /dev/hdb2
  ```

LVM Commands 3/7

- **Creating** an **LV** in `my_VG: # lvcreate options -nmy_LV`

  Main options:

  `-C` ***y/n*** Linear. Default is no Linear (Striped)

  `-i` ***PV_Number*** number of PVs to scatter the LV.

  `-I` ***S_Size*** **S**trip size (in KBytes). The strip is the transactions unit.

  `-l` ***LEs_Number*** number of LEs (LE size = PE size) for the new LV.

  `-L` ***LV_Size[kKmMgGtT]*** size for the new LV. K (kilobytes), M (megabytes)
  G (gigabytes) or T (terabytes). Default unit is megabytes.

  `-n` **LV_Name**

  -`p` **r/w** permission

LVM Commands 4/7

- To create a 1500MB linear LV named `my_LV` in `my_VG`:

```
# lvcreate -Cy -L1500 -nmy_LV my_VG
```

- To create a LV of size 100 LEs (or PEs), with 2 PVs and strip size of 4 KB:

```
# lvcreate -i2 -I4 -l100 -nmy_LV my_VG
```

```
# vgdisplay my_VG
```

```
— Volume group —
VG Name my_VG
VG Access read/write
VG Status available/resizable
VG # 0
MAX LV 256
- - - - - - - - - - - - - -
```

LVM Commands 5/7

```
VG Size 1.95 GB

PE Size 4 MB

Total PE 498

Alloc PE / Size 100 / 400 MB

Free PE / Size 398 / 1.55 GB
```

- If you want to create an LV that uses the entire VG, use `vgdisplay` to find the "Total PE" size, then use that when running `lvcreate`:

```
# vgdisplay my_VG | grep "Total PE"
```

```
Total PE 10230
```

```
# lvcreate -l10230 my_VG -nmy_LV
```

LVM Commands 6/7

- **Create** an **ext2** file system on the LV

```
# mke2fs /dev/my_VG/my_LV
```

- **Mount** and **Test** the File System

```
# mount -t auto /dev/my_VG/my_LV /mnt/LV1

# df
```

```
   Filesystem       1k-blocks     Used     Available   Use%   Mounted on
   /dev/hda1        35886784    13360620   20703192    40%        /
/dev/my_VG/my_LV     396672        13       376179      1%      /mnt/LV1
```

- Obtaining **LV** information

`# lvdisplay /dev/my_VG/my_LV` (strip size 4KB)

- **Removing** a **LV**

```
# umount /dev/my_VG/my_LV && lvremove /dev/my_VG/my_LV
```

LVM Commands 7/7

- **Extending** (+4MBs) a **LV** (`/dev/my_VG/my_LV`) mounted on `/mnt/LV1`:

```
# lvextend -L+4M /dev/my_VG/my_LV
# umount /dev/my_VG/my_LV
# e2fsck -f /dev/my_VG/my_LV
# resize2fs /dev/my_VG/my_LV
# mount /dev/my_VG/my_LV /mnt/LV1
```

- **Reducing** (-4MBs) a **LV** (`/dev/my_VG/my_LV`) mounted on `/mnt/LV1`:

```
# lvreduce -L-4M /dev/my_VG/my_LV
# umount /mnt/LV1
# e2fsck -f /dev/my_VG/my_LV
# resize2fs /dev/my_VG/my_LV
# mount /dev/my_VG/my_LV /mnt/LV1
```

Examples

- Setting up LVM on three SCSI disks (/dev/sda, /dev/sdb, and /dev/sdc) with striping
- Taking a Backup Using Snapshots
- Removing an Old Disk (/dev/hdb)
- Distributing Old Extents to a New Replacement Disk

Setting up LVM on three SCSI disks (/dev/sda, /dev/sdb, and /dev/sdc) with striping 1/2

- Preparing the disk partitions

```
# pvcreate /dev/sda

# pvcreate /dev/sdb

# pvcreate /dev/sdc
```

- Setup a VG

1. Create a VG

   ```
   # vgcreate my_VG /dev/sda /dev/sdb /dev/sdc
   ```

2. Run `vgdisplay` to verify volume group

Setting up LVM on three SCSI disks (/dev/sda, /dev/sdb, and /dev/sdc) with striping 2/2

- **Creating** the **LV** (1GB) with 3 PVs and strip size of 4 KB on the VG:

```
# lvcreate -i3 -I4 -L1G -nmy_LV my_VG
```

- **Create** an **ext2** file system on the LV

```
# mke2fs /dev/my_VG/my_LV
```

- **Mount** and **Test** the File System

```
# mount /dev/my_VG/my_LV /mnt

# df
```

Taking a Backup Using Snapshots

1. **Create** the snapshot volume of 500MB (free space in **`my_VG`**):

   ```
   # lvcreate -L500M -s -nmy_LV_backup /dev/my_VG/my_LV
   ```

2. **Mount** the snapshot volume:

   ```
   # mkdir /mnt/my_LV_backup
   # mount /dev/my_VG/my_LV_backup /mnt/my_LV_backup
   ```

3. Remove the snapshot:

   ```
   # umount /mnt/my_LV_backup
   # lvremove /dev/my_VG/my_LV_backup
   ```

Removing an Old Disk (/dev/hdb)

1. Distributing Old Extents to Existing Disks in VG

```
# pvmove /dev/hdb
```

2. Remove the unused disk

```
# vgreduce my_VG /dev/hdb
```

3. The drive can now be removed when the machine is powered down

Distributing Old Extents to a New Replacement Disk

1. Prepare the new disk

   ```
   # pvcreate /dev/sdf
   ```

2. Add it to the VG `my_VG`

   ```
   # vgextend my_VG /dev/sdf
   ```

3. Move the data

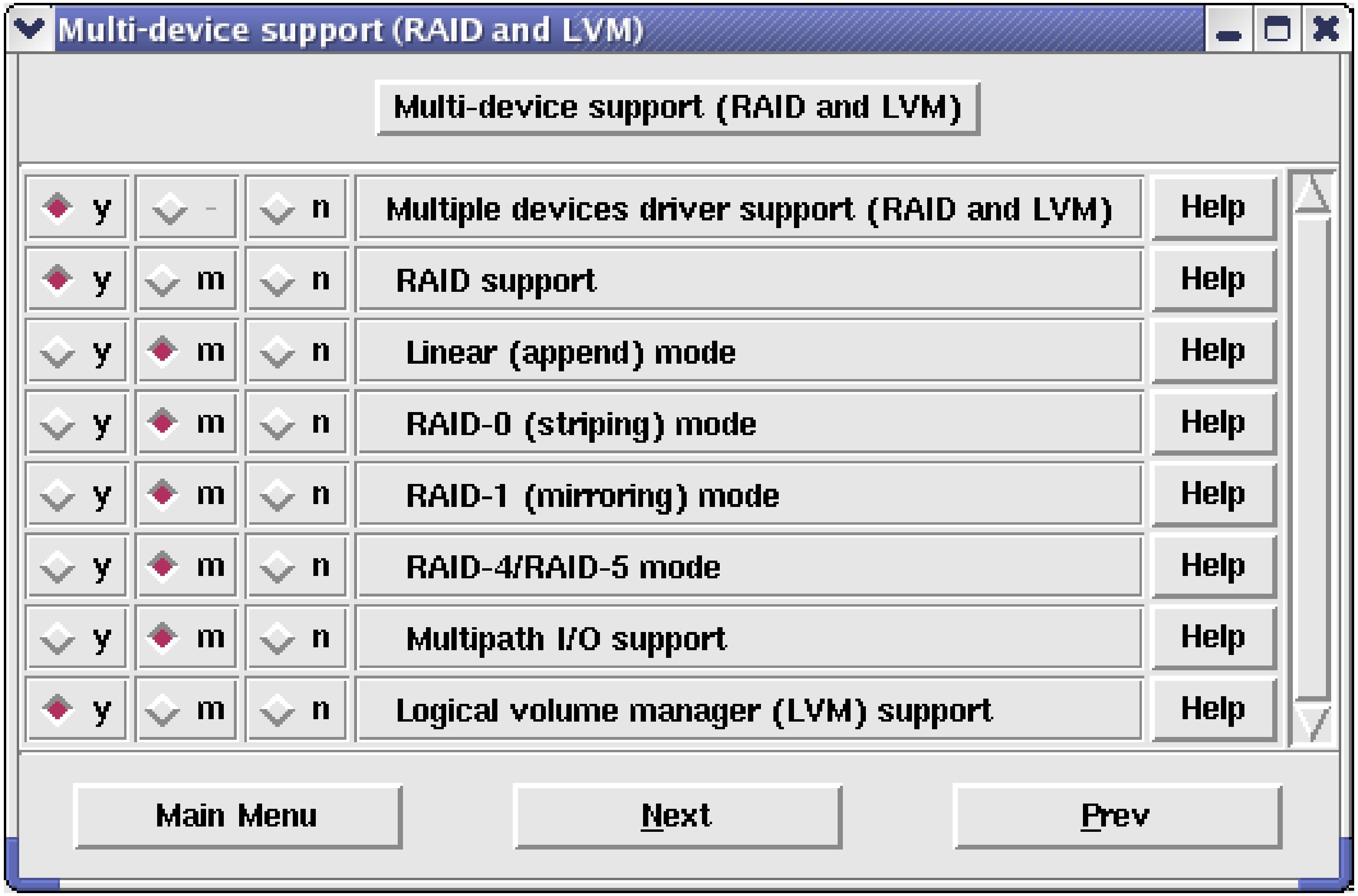
   ```
   # pvmove /dev/hdb /dev/sdf
   ```

4. Remove the unused disk

   ```
   # vgreduce my_VG /dev/hdb
   ```

5. The drive can now be removed when the machine is powered down

Kernel configuration



Multi-device support (RAID and LVM)
Multi-device support (RAID and LVM)
Help
Main Menu
Next
Prev

● y	○ -	○ n	Multiple devices driver support (RAID and LVM)
● y	○ m	○ n	RAID support
○ y	● m	○ n	Linear (append) mode
○ y	● m	○ n	RAID-0 (striping) mode
○ y	● m	○ n	RAID-1 (mirroring) mode
○ y	● m	○ n	RAID-4/RAID-5 mode
○ y	● m	○ n	Multipath I/O support
● y	○ m	○ n	Logical volume manager (LVM) support