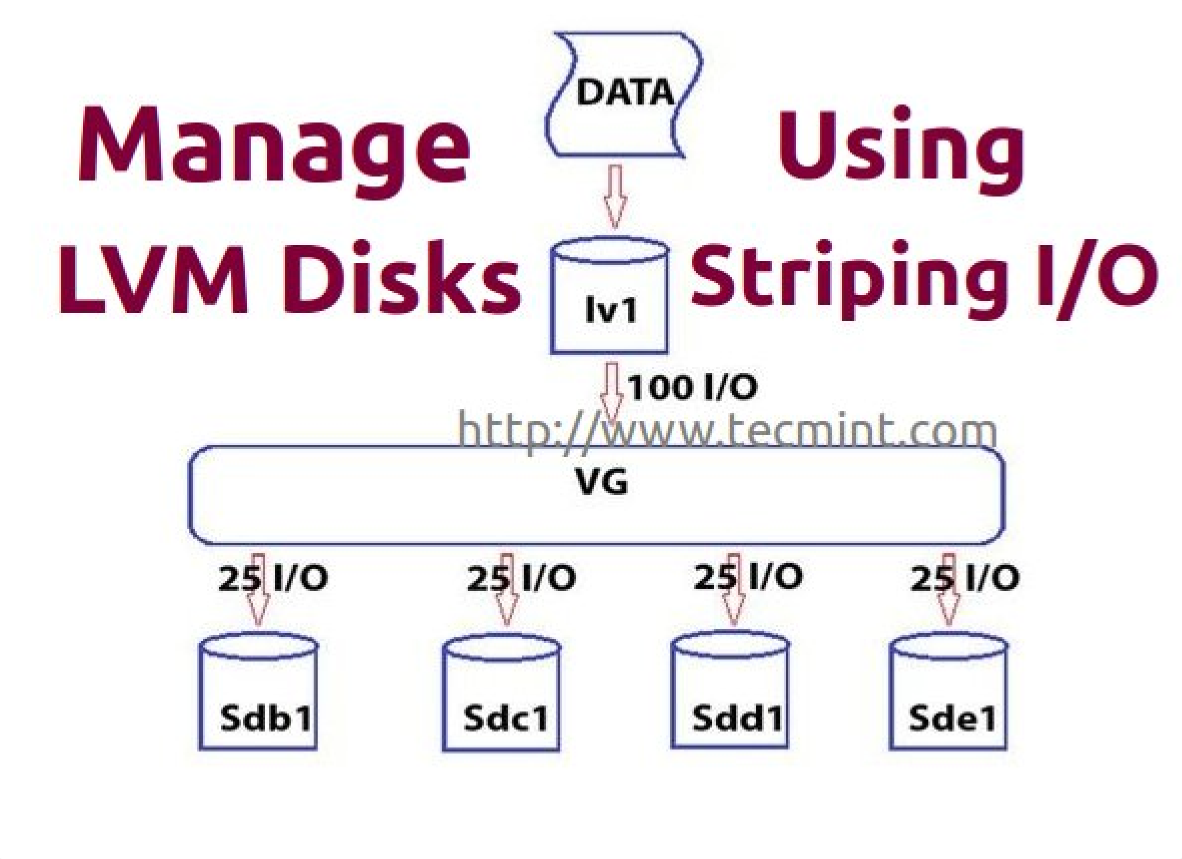
Manage Multiple Logical Volume Management Disks using Striping I/O

by Babin Lonston | Published: September 16, 2014 | Last Updated: January 3, 2015

In this article, we are going to see how the logical volumes writes the data to disk by striping I/O. Logical Volume management has one of the cool feature which can write data over multiple disk by striping the I/O.



Manage LVM Disks Using Striping I/O

What is LVM Striping?

LVM Striping is one of the feature which will writes the data over multiple disk, instead of constant write on a single Physical volume.

Features of Striping

- It will increase the performance of disk.
- Saves from hard write over and over to a single disk.
- Disk fill-up can be reduced using striping over multiple disk.

In Logical volume management, if we need to create a logical volume the extended will get fully mapped to the volume group and physical volumes. In such situation if one of the **PV** (Physical Volume) gets filled we need to add more extends from other physical volume. Instead, adding more extends to PV, we can point our logical volume to use the particular Physical volumes writing I/O.

Assume we have **four disks** drives and pointed to four physical volumes, if each physical volume are capable of **100 I/O** totally our volume group will get **400 I/O**.

If we are not using the **stripe method**, the file system will writes across the underlying physical volume. For example, some data writes to physical volume 100 I/O will be write only to the first (**sdb1**) PV. If we create the logical volume with stripe option while writing, it will write to every four drives by splitting 100 I/O, that means every four drive will receive 25 I/O each.

This will be done in round robin process. If any one of the logical volume need to be extended, in this situation we can't add **1** or **2** PV. We have to add all **4** pvs to extend the logical volume size. This is one of the drawback in stripe feature, from this we can know that while creating logical volumes we need to assign the same stripe size over all logical volumes.

Logical Volume management has these features which we can stripe the data over multiple pvs at the same time. If you are familiar with logical volume you can go head to setup the logical volume stripe. If not then you must need to know about the logical volume managements basics, read below articles to know more about logical volume management.

Requirements

- Setup Flexible LVM Disk Storage in Linux – Part I

- How to Extend/Reduce LVM's in Linux – Part II

My Server Setup

Here I'm using **Centos6.5** for my workout. The same steps can be used in RHEL, Oracle Linux, and most of the distributions.

```
Operating System :       CentOS 6.5
IP Address :             192.168.0.222
Hostname :               tecmint.storage.com
```

Logical Volume management using Striping I/O

For demonstration purpose, I've used 4 Hard drives, each drive with 1 GB in Size. Let me show you four drives using '**fdisk**' command as shown below.

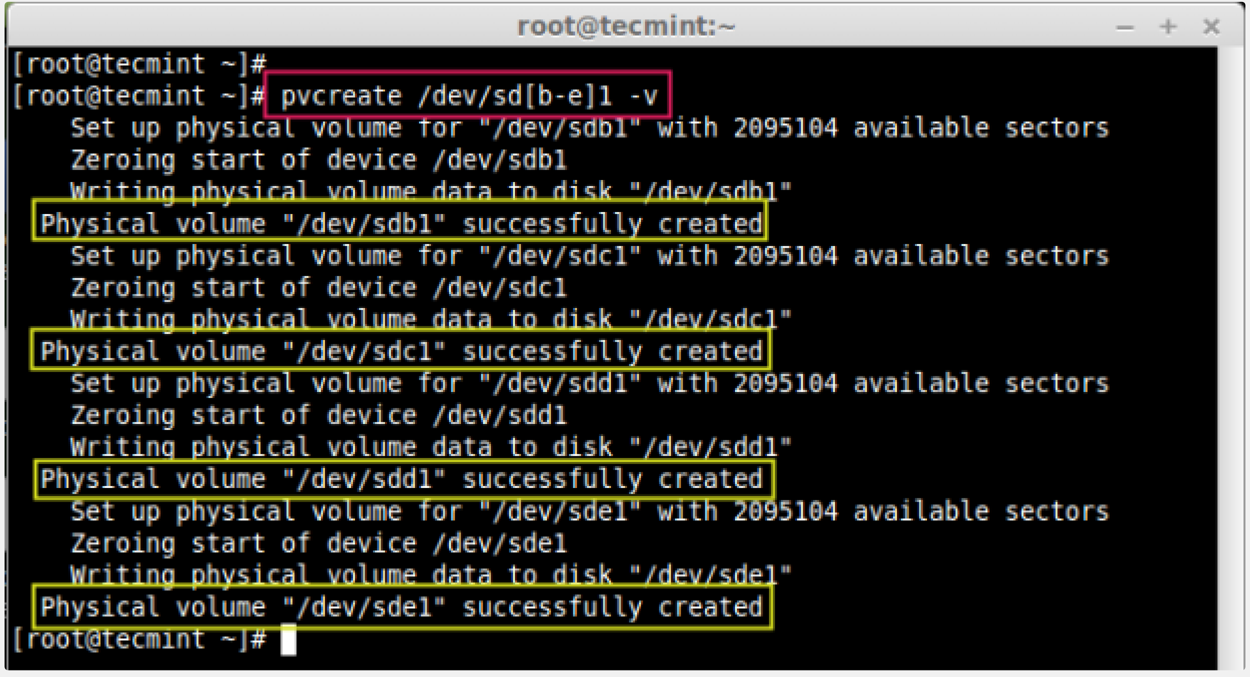
```
# fdisk -l | grep sd
```

List Hard Drives

Now we've to create partitions for these 4 hard drives **sdb**, **sdc**, **sdd** and **sde** using '**fdisk**' command. To create partitions, please follow the **step #4** instructions, given in the **Part 1** of this article (link give above) and make sure you change the type to **LVM** (**8e**), while creating partitions.

After you've created partitions successfully, now move forward to create Physical volumes using all these 4 drives. For creating PV's, use the following '**pvcreate**' command as shown.

```
# pvcreate /dev/sd[b-e]1 -v
```


```
[root@tecmint ~]#
[root@tecmint ~]# pvcreate /dev/sd[b-e]1 -v
    Set up physical volume for "/dev/sdb1" with 2095104 available sectors
    Zeroing start of device /dev/sdb1
    Writing physical volume data to disk "/dev/sdb1"
  Physical volume "/dev/sdb1" successfully created
    Set up physical volume for "/dev/sdc1" with 2095104 available sectors
    Zeroing start of device /dev/sdc1
    Writing physical volume data to disk "/dev/sdc1"
  Physical volume "/dev/sdc1" successfully created
    Set up physical volume for "/dev/sdd1" with 2095104 available sectors
    Zeroing start of device /dev/sdd1
    Writing physical volume data to disk "/dev/sdd1"
  Physical volume "/dev/sdd1" successfully created
    Set up physical volume for "/dev/sde1" with 2095104 available sectors
    Zeroing start of device /dev/sde1
    Writing physical volume data to disk "/dev/sde1"
  Physical volume "/dev/sde1" successfully created
[root@tecmint ~]#
```

Create Physical Volumes in LVM

Once PV's created, you can list them using '**pvs**' command.

```
# pvs
```

Verify Physical Volumes

Now we need to define volume group using those 4 physical volumes. Here I'm defining my volume group with **16MB** of Physical extended size (PE) with volume group named as **vg_strip**.

```
# vgcreate -s 16M vg_strip /dev/sd[b-e]1 -v
```

The description of above options used in the command.

- **[b-e]1** – Define your hard drive names such as sdb1, sdc1, sdd1, sde1.
- **-s** – Define your physical extent size.
- **-v** – verbose.

Next, verify the newly created volume group using.

```
# vgs vg_strip
```

Verify Volume Group

To get more detailed information about VG, use switch '**-v**' with **vgdisplay** command, it will give us a every physical volumes which all used in **vg_strip** volume group.

```
# vgdisplay vg_strip -v
```

Volume Group Information

Back to our topic, now while creating Logical volume, we need to define the stripe value, how data need to write in our logical volumes using stripe method.

Here I'm creating a logical volume in the name of **lv_tecmint_strp1** with **900MB** size, and it needs to be in **vg_strip** volume group, and I'm defining as 4 stripe, it means the data writes to my logical volume, needs to be stripe over 4 PV's.

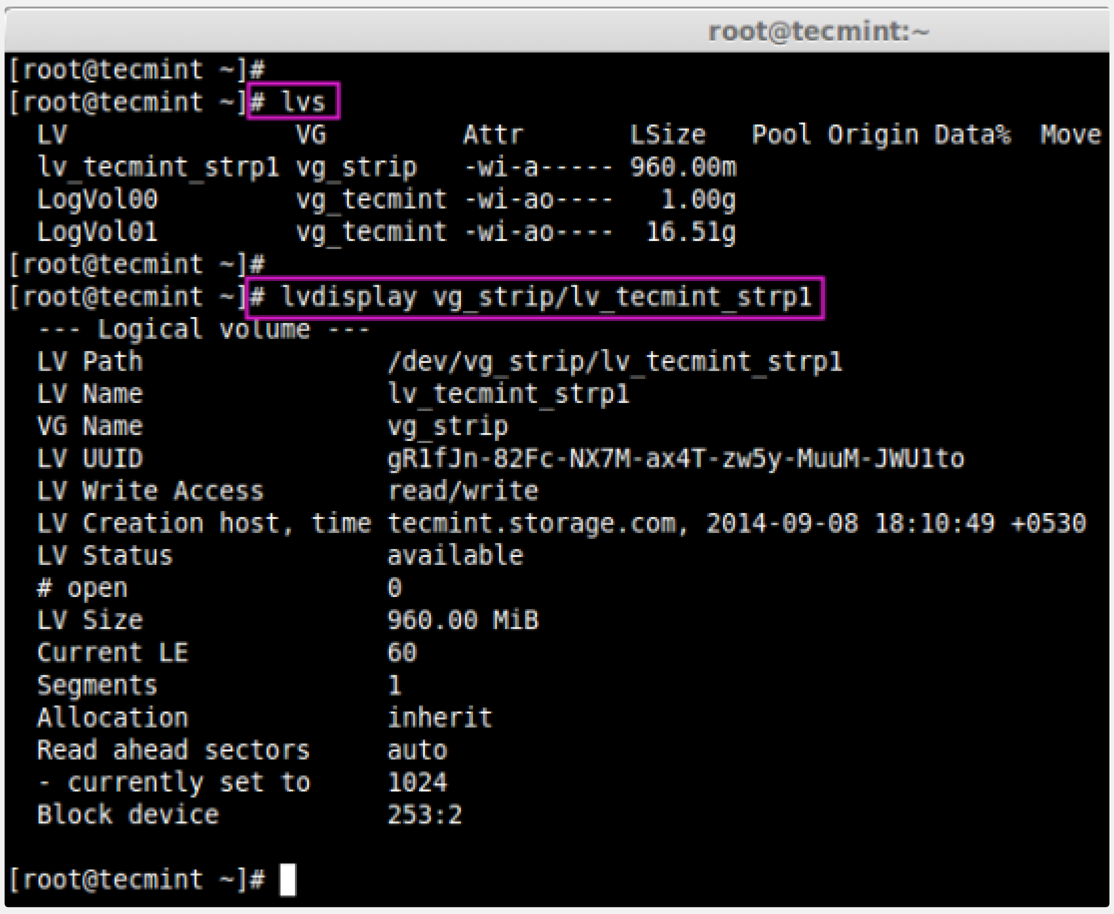
```
# lvcreate -L 900M -n lv_tecmint_strp1 -i4 vg_strip
```

- -L –logical volume size
- -n –logical volume name
- -i –stripes

Create Logical Volumes

In the above image, we can see that the default size of stripe-size was **64 KB**, if we need to define our own stripe value, we can use **-I** (Capital I). Just to confirm that the logical volume are created use the following command.

```
# lvdisplay vg_strip/lv_tecmint_strp1
```



```
root@tecmint:~
[root@tecmint ~]#
[root@tecmint ~]# lvs
  LV               VG         Attr       LSize   Pool Origin Data%  Move
  lv_tecmint_strp1 vg_strip   -wi-a----- 960.00m
  LogVol00         vg_tecmint -wi-ao----   1.00g
  LogVol01         vg_tecmint -wi-ao----  16.51g
[root@tecmint ~]#
[root@tecmint ~]# lvdisplay vg_strip/lv_tecmint_strp1
  --- Logical volume ---
  LV Path                /dev/vg_strip/lv_tecmint_strp1
  LV Name                lv_tecmint_strp1
  VG Name                vg_strip
  LV UUID                gR1fJn-82Fc-NX7M-ax4T-zw5y-MuuM-JWU1to
  LV Write Access        read/write
  LV Creation host, time tecmint.storage.com, 2014-09-08 18:10:49 +0530
  LV Status              available
  # open                 0
  LV Size                960.00 MiB
  Current LE             60
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     1024
  Block device           253:2

[root@tecmint ~]#
```

Confirm Logical Volumes

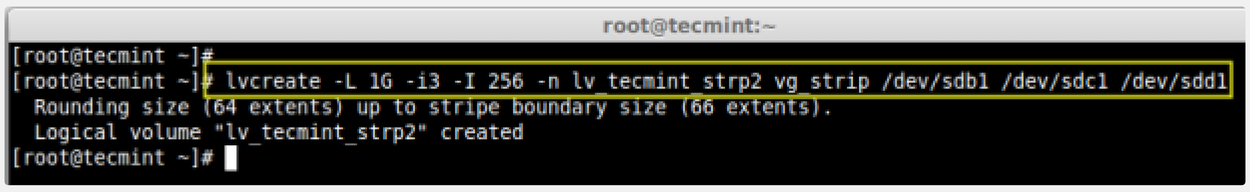
Now next question will be, How do we know that stripes are writing to 4 drives?. Here we can use '**lvdisplay**' and **-m** (display the mapping of logical volumes) command to verify.

```
# lvdisplay vg_strip/lv_tecmint_strp1 -m
```

Check Logical Volumes

To create our defined stripe size, we need to create one logical volume with **1GB** size using my own defined Stripe size of **256KB**. Now I'm going to stripe over only 3 PV's, here we can define which pvs we want to be striped.

```
# lvcreate -L 1G -i3 -I 256 -n lv_tecmint_strp2 vg_strip /dev/sdb1 /dev/sdc1 /dev/sdd1
```



```
[root@tecmint ~]#
[root@tecmint ~]# lvcreate -L 1G -i3 -I 256 -n lv_tecmint_strp2 vg_strip /dev/sdb1 /dev/sdc1 /dev/sdd1
  Rounding size (64 extents) up to stripe boundary size (66 extents).
  Logical volume "lv_tecmint_strp2" created
[root@tecmint ~]# 
```

Define Stripe Size

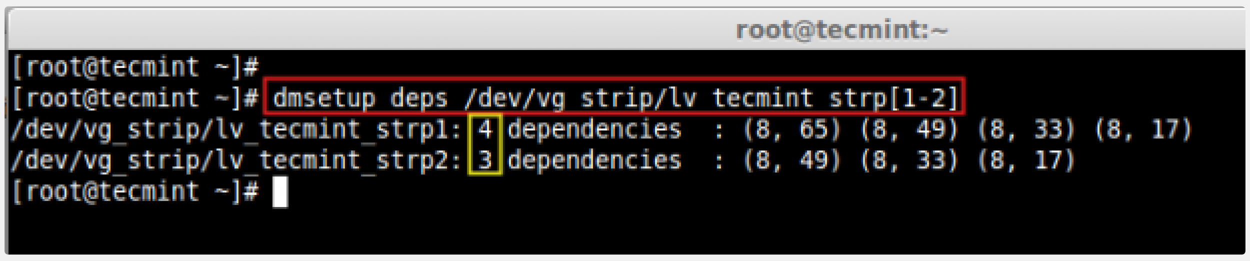
Next, check the stripe size and which volume does it stripes.

```
# lvdisplay vg_strip/lv_tecmint_strp2 -m
```

Check Stripe Size

It's time to use a device mapper, for this we use command '**dmsetup**'. It is a low level logical volume management tool which manages logical devices, that use the device-mapper driver. We can see the lvm information using dmsetup command to know the which stripe depends on which drives.

```
# dmsetup deps /dev/vg_strip/lv_tecmint_strp[1-2]
```



```
[root@tecmint ~]#
[root@tecmint ~]# dmsetup deps /dev/vg_strip/lv_tecmint_strp[1-2]
/dev/vg_strip/lv_tecmint_strp1: 4 dependencies  : (8, 65) (8, 49) (8, 33) (8, 17)
/dev/vg_strip/lv_tecmint_strp2: 3 dependencies  : (8, 49) (8, 33) (8, 17)
[root@tecmint ~]#
```

Device Mapper

Here we can see that strp1 depend on 4 drives, and strp2 depend on 3 devices.

Hope you have learnt, that how we can stripe through logical volumes to write the data. For this setup one must know about the basic of logical volume management. In my next article, I will show you how we can migrate in logical volume management, till then stay tuned for updates and don't forget to give valuable comments about the article.