Striped LVM Volumes

AUGUST 28, 2017 *by* **THE URBAN PENGUIN (HTTPS://WWW.THEURBANPENGUIN.COM/AUTHOR/THE-URBAN-PENGUIN/)**

In this module we look at Striped LVM Volumes and how we gain a performance hit by striping data in LVM2.

Gaining Performance with LVM Striped Volumes

Another consideration, when creating an LVM Logical Volume is to stripe the data. Normally, if an LV needs to span across multiply physical drives the data is written to the first PV and only crosses to the next drive when space is exhausted on the first PV. If we consider that each drive may be capable of 100 IOPS, then writing the data in a linear fashion like this to the disk does not take advantage of the 2 disks and dual channels that we have to read and write data. Potentially, we could gain 200 IOPS if both disk were written to concurrently. This is where striping comes into play. Instead of writing data in a linear fashion, data is written in stripes across each disk making up the LV. If we have two disks then we would stripe across those two disks. If we had 3 disks then we would be striping across 3 disks.

Follow the link below to download the 'Complete Guide to LVM2 in Linux', costing just £1.99! Once it is downloaded the eBook can be printed out if required.

Analysing a Standard Logical Volume

To begin with we will create a standard LV and take a look at where the LV is created. In this demonstration, we will be using the Volume Group, vg2; however, any VG with available space on multiple PVs will suffice for the example.

Striped and Linear LVs can coexist in the same VG.

```
# lvcreate -n lv2 -L 64m vg2
   Logical volume “lv2” created.
```

In this instance, we have created a very small Logical Volume that, probably, can exist on just one PV. There are two PVs with available space in the vg2 Volume Group. To identify where the Logical Volume has been created we can use one of 3 commands. Firstly, we will begin with the lvs command:

```
# lvs -o +devices vg2
   LV   VG   Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
   lv2  vg2  -wi-a----- 64.00m                                             /dev/loop0(0)
```

We have used the +devices option before but this time we drill down to just the one Volume Group by adding in the vg2 name. We see that the Logical Volume has been created only on the /dev/loop0 device. This would give us a single channel to read and write data, where, if the data was striped across two disks then, we would double the channel capacity for IOPS.

In reality, to gain additional performance the PVs would need to be located on separate disks. So not one disk with two partitions or, as in this case, two disk files on the same disk. This is one reason we suggest not using multiple partitions on disks for LVs, no advantage is gained with multiple partitions in most cases.

Another method, that we have used before is using the lsblk command, again we can drill down to just the loop back devices that make up the Volume Group:

```
# lsblk /dev/loop[0,1]
 NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
 loop0       7:0    0  100M  0 loop
 └─vg2-lv2 252:6    0   64M  0 lvm
 loop1       7:1    0  100M  0 loop
```

We, again, see that the new LV is 64M in size and only located on the loop0 device.

A new method that we can use to see the LV disk assignment is to make use of Devmapper commands. After all, LVM2 uses the Devmapper module:

```
# dmsetup deps /dev/vg2/lv2
1 dependencies : (7, 0)
```

The output is a little more cryptic when we view it this time. The disk devices are listed by the Major and Minor numbers, or 7,0 as we see in this example. 7 is the Major number or driver used for Loopback devices and being loop0, the Minor number is 0.

As we don’t need this LV now we will remove it before creating the Striped Logical Volume:

```
# lvremove /dev/vg2/lv2
Do you really want to remove and DISCARD active logical volume lv2? [y/n]: y
Logical volume “lv2” successfully remove
```

Creating Striped LVM Volumes

Striping the LV is a really simple matter when we look at creating the LV:

```
# lvcreate -n lv2 -L 64m -i2 vg2
   Using default stripesize 64.00 KiB.
   Logical volume “lv2” created.
```

Just using the option -i, we specify how many devices to stripe over. We have chosen 2 devices. The command lsblk now shows clearly that we have created the LV across the 2 devices:

```
# lsblk /dev/loop[0,1]
 NAME      MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
 loop0       7:0    0  100M  0 loop
 └─vg2-lv2 252:6    0   64M  0 lvm
 loop1       7:1    0  100M  0 loop
 └─vg2-lv2 252:6    0   64M  0 lvm
```

The LV is still only 64M but it is written in 64KiB stripes across both devices. We can change the stripe size using the -I option, i.e. Using -i2 -I128 will write across 2 devices in 128KiB stripes.

The command dmsetup also shows both devices:

```
# dmsetup deps /dev/vg2/lv2
 2 dependencies : (7, 1) (7, 0)
```

As does lvs:

```
# lvs vg2 -o +device
   LV   VG   Attr       LSize  Pool Origin Data%  Meta%  Move Log Cpy%Sync Convert Devices
   lv2  vg2  -wi-a----- 64.00m                             /dev/loop0(0),/dev/loop1(0)
```

Another way we can view a striped volume in action is using the dmsetup command with the table option. Here is the output of demsetup that includes another striped volume:

```
# dmsetup table
 vg1-tv1: 0 2097152 thin 252:3 1
 vg1-tpool-tpool: 0 409600 thin-pool 252:0 252:1 128 409600 0
 vg1-tpool_tdata: 0 409600 linear 8:34 223232
 vg1-tpool_tmeta: 0 8192 linear 8:34 632832
 vg1-tpool: 0 409600 linear 252:3 0
 vg1-lv1: 0 1015808 linear 8:33 2048
 vg1-lv1: 1015808 212992 linear 8:34 2048
 vg2-s1: 0 131072 striped 2 128 7:0 2048 7:1 2048
```

Take a look at the detail for vg2-s1 and compare it wil the linera volume vg1-lv1.

There has to be a Catch

Well if striping is so good what is the catch?

- Firstly, it is ideal that PVs of the same size are used in Volume Groups for striping. we don't want 1TB PV and a 2TB PV, leaving the potential for 1TB of unused space on the 2TB drive. Nor much of a drawback but it worth knowing.
- Perhaps a bigger issue is that of extending Striped LVs. If I have a striped LV across 2 PVs I have to have space on a further 2 PVs to be able to extend the LV. Often, this would mean that I have to add 2 PVs to the Volume Group. Having an LV striped across 4 devices would mean that I would need at add a further 4 devices to be able to extend the LV when space was exhausted in the VG.

(https://www.theurbanpenguin.com/product/complete-guide-lvm2-linux/)