

Linux 系统下调整 VG 大小

一【实验目标】

- 学习并掌握 Linux 系统下创建 PV

二【实验环境】

- 实验机环境：Centos 6.6
- 目标机环境：Centos 6.6
- 实验拓扑：如图 1 所示。



图 1 实验拓扑

三【实验原理】

在 Linux 系统下，我们往往会遇到扩充磁盘的情况。普通情况下需要新加一块盘，重分区、格式化、数据复制、卸载就分区、挂载新分区等繁琐的步骤。其实，我们可以在安装系统时使用 LVM 来管理我们的文件系统，这样就可以弹性调整文件系统的容量。

四【实验步骤】

1、新加磁盘启动系统后，查看现有磁盘使用情况

查看现有 lv_test 容量

命令：df -h

```
[uroot@localhost ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/mapper/VolGroup-lv_root
                 50G   3.3G   44G    7% /
tmpfs           497M    84K  497M    1% /dev/shm
/dev/sda1       477M   29M  424M    7% /boot
/dev/mapper/VolGroup-lv_home
                 12G    30M   11G    1% /home
```

图 2

2、在磁盘 sdb 上创建新分区

命令：fdisk /dev/sdb

输入 p 打印现有分区情况(还没分区)

输入 n 新建分区

输入 p 建立主分区

输入 2 建立第二个主分区

分区起始位置和最后位置可以直接回车，采用默认值。

输入 p 打印分区情况，发现已建立一个分区/dev/sdb2，但是此分区为 linux 格式

```
[root@localhost ~]# fdisk /dev/sdb

WARNING: GPT (GUID Partition Table) detected on '/dev/sdb'! The util fdisk doesn't support GPT. Use GNU Parted.

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to switch off the mode (command 'c') and change display units to sectors (command 'u').

Command (m for help): p

Disk /dev/sdb: 1000.2 GB, 1000170586112 bytes
255 heads, 63 sectors/track, 121597 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks    Id System
/dev/sdb1             1           650       5221093+   8e  Linux LVM

Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 2
First cylinder (651-121597, default 651):
Using default value 651
Last cylinder, +cylinders or +size{K,M,G} (651-121597, default 121597):
Using default value 121597

Command (m for help): p

Disk /dev/sdb: 1000.2 GB, 1000170586112 bytes
255 heads, 63 sectors/track, 121597 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks    Id System
/dev/sdb1             1           650       5221093+   8e  Linux LVM
/dev/sdb2          651       121597      971506777+   83   Linux
```

由于分区/dev/sdb2 为 Linux 格式，我们需要改变系统标识符为 Linux LVM 格式：

输入 t 改变分区的属性

输入 2 表示改变第二个分区的属性

输入 8e 改变分区 2 为 Linux LVM 格式

输入 p 打印分区情况，发现建立的分区/dev/sdb1 为 Linux LVM 格式输入 w 保存分区

```

Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): 8e
Changed system type of partition 2 to 8e (Linux LVM)

Command (m for help): p

Disk /dev/sdb: 1000.2 GB, 1000170586112 bytes
255 heads, 63 sectors/track, 121597 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1             1           650       5221093+   8e  Linux LVM
/dev/sdb2          651       121597       971506777+   8e  Linux LVM

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource
busy.
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.

```

再次使用 `fdisk -l` 查看系统内存从盘情况发现 `/dev/sdb` 上已有一个 Linux LVM 格式的 `/dev/sdb2` 分区

```
[root@localhost uroot]# fdisk -l

Disk /dev/sda: 68.7 GB, 68719476736 bytes
255 heads, 63 sectors/track, 8354 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk identifier: 0x000c9a6b

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1 *          1           64       512000    83  Linux
Partition 1 does not end on cylinder boundary.
/dev/sda2            64        8355     66595840    8e  Linux LVM

Disk /dev/mapper/VolGroup-lv_root: 53.7 GB, 53687091200 bytes
255 heads, 63 sectors/track, 6527 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk identifier: 0x00000000

Disk /dev/mapper/VolGroup-lv_swap: 2113 MB, 2113929216 bytes
255 heads, 63 sectors/track, 257 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disk identifier: 0x00000000

WARNING: GPT (GUID Partition Table) detected on '/dev/sdb'! The util fdisk doesn't support GPT. Use GNU Parted.

Disk /dev/sdb: 1000.2 GB, 1000170586112 bytes
255 heads, 63 sectors/track, 121597 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk identifier: 0x00000000

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1            1           650     5221093+    8e  Linux LVM
/dev/sdb2           651       121597     971506777+    8e  Linux LVM
```

3、创建 PV:

命令: pvcreate /dev/sdb2

查看系统 pv: pvscan

```
[root@localhost uroot]# pvcreate /dev/sdb2
Physical volume "/dev/sdb2" successfully created
[root@localhost uroot]# pvscan
PV /dev/sdb1   VG vg_test    lvm2 [4.98 GiB / 0   free]
PV /dev/sda2   VG VolGroup    lvm2 [63.51 GiB / 0   free]
PV /dev/sdb2   VG              lvm2 [926.50 GiB]
Total: 3 [994.99 GiB] / in use: 2 [68.48 GiB] / in no VG: 1 [926.50 GiB]
```

4、增加 VG 容量

命令: vgextend vg_test /dev/sdb2

查看: vgdisplay

```
[root@localhost ~]# vgextend vg_test /dev/sdb2
Volume group "vg_test" successfully extended
[root@localhost ~]# vgs
--- Volume group ---
VG Name          vg_test
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 3
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV          1
Open LV          0
Max PV           0
Cur PV          2
Act PV           2
VG Size          931.48 GiB
PE Size          4.00 MiB
Total PE         238458
Alloc PE / Size  1274 / 4.98 GiB
Free PE / Size   237184 / 926.50 GiB
VG UUID          k2BBqd-vmy8-9cSX-YEZw-7dgz-aLjf-lnQ640
```

图 9

五【实验思考】

- 使用 fdisk 命令进行分区。
- 增加 VG 容量命令 vgextend。
- 查看 VG 容量命令 vgs。