

一、LVM的概念和基本原理

LVM是Linux为了动态管理磁盘空间，而实现的一个物理磁盘和文件系统之间的中间层实现。

涉及的相关概念：

物理存储介质 (*The physical media*)

这里指系统的存储设备：硬盘，如：/dev/hda1、/dev/sda等等，是存储系统最低层的存储单元。

物理卷 (*physical volume*)

物理卷就是指硬盘分区或从逻辑上与磁盘分区具有同样功能的设备(如RAID)，是LVM的基本存储逻辑块，但和基本的物理存储介质（如分区、磁盘等）比较，却包含有与LVM相关的管理参数。

卷组 (*Volume Group*)

LVM卷组类似于非LVM系统中的物理硬盘，其由物理卷组成。可以在卷组上创建一个或多个“LVM分区”（逻辑卷），LVM卷组由一个或多个物理卷组成。

逻辑卷 (*logical volume*)

LVM的逻辑卷类似于非LVM系统中的硬盘分区，在逻辑卷之上可以建立文件系统(比如/home或者/usr等)。

PE (*physical extent*)

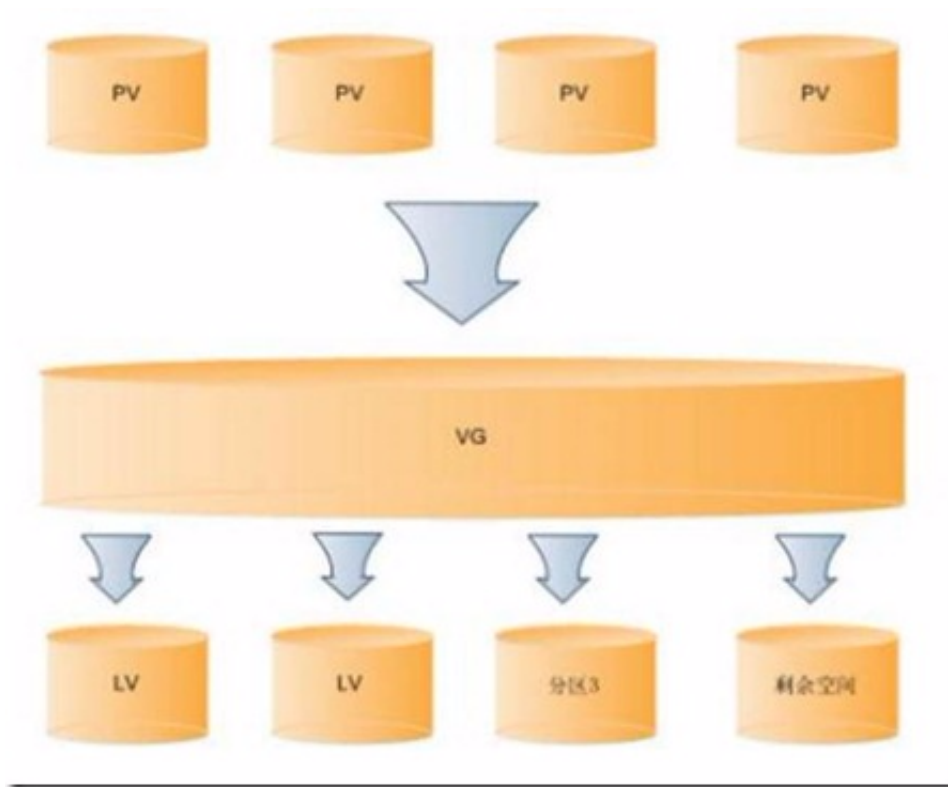
每一个物理卷被划分为称为PE(Physical Extents)的基本单元，具有唯一编号的PE是可以被LVM寻址的最小单元。PE的大小是可配置的，默认为4MB。

LE (*logical extent*)

逻辑卷也被划分为被称为LE(Logical Extents)的可被寻址的基本单位。在同一个卷组中，LE的大小和PE是相同的，并且一一对应。

简单来说，LVM将每个物理设备格式化为一个PV，这个PV由若干PE组成。。

一个或者多PV可以整合成一个VG，VG相当正常认知下的物理硬盘，可以在VG上进行分区，得到若干LV，用于挂载文件系统。



二、LVM的基本磁盘管理操作

```
1 #查看物理盘信息
2 fdisk -l
3
4 #将物理盘格式化为PV（物理卷）
5 pvcreate /dev/xxx
6
7 #查看PV信息
8 pvdisplay #所有PV信息
9 pvs #当前PV信息
10
11 #创建卷组(VG)，并将PV加入到卷组中
12 vgcreate vname /dev/xxx
13
14 #查看vg信息
15 vgdisplay #所有PV信息
16 vgs #当前PV信息
17
18 #基于卷组(VG)创建逻辑卷(LV)
19 lvcreate -n lvname -L 2G vname
20
```

```

21 #查看卷信息
22 lvdisplay    #详细卷信息
23 lvs #对应的卷信息
24
25 #格式化分区、并挂载使用
26 mkfs.ext4 /dev/vgname/lvname
27 mount /dev/vgname/lvname <挂载点>
28 mount    #当前的挂载状态
29

```

删除操作

```

1 ①首先将正在使用的逻辑卷卸载掉
2 umount /dev/vgname/lvname
3 ②将逻辑卷先删除
4 lvremove /dev/vgname/lvname
5 ③删除卷组
6 vgremove vgname
7 ④最后再来删除我们的物理卷
8 pvremove /dev/xxx

```

扩容、裁剪操作

```

1 #扩容VG
2 vgextend vgname /dev/hda6          #将PV /dev/hda6加入分区
3 #扩容LV操作
4 lvextend -L 1G /dev/VolGroup00/LogVol01    #扩展LV
5 resize2fs /dev/VolGroup00/LogVol01        #执行该重设大小，对于当前正在使用的
6 #缩小LV
7 umount /mnt                            #一定要先卸载
8 e2fsck -f /dev/vgname/lvname            #使用 e2fsck 命令来检查我们的文件
9 resize2fs /dev/vgname/lvname 10G        #将文件系统缩小为10G
10 lvreduce -L -4G /dev/vgname/lvname      # 缩小4G大小的逻辑卷
11 #缩小VG
12 umount /mnt                            #卸载所有挂载
13 vgreduce vgname pvname                #将pv从对应的VG中移除

```

```
14 #删除PV
15 pvremove /dev/sdd
```

参考链接：

<http://www.cnblogs.com/xiaoluo501395377/archive/2013/05/22/3093405.html>

<http://www.cnblogs.com/xiaoluo501395377/archive/2013/05/24/3096087.html>

<http://www.cnblogs.com/xiaoluo501395377/archive/2013/05/24/3097785.html>

三、磁盘分区相关知识

主磁盘分区、扩展磁盘分区、逻辑分区

一个硬盘主分区至少有1个，最多4个，扩展分区（EBR）可以没有，最多1个。且主分区+扩展分区总共不能超过4个。逻辑分区可以有若干个。

主分区的编号一般为1-4（hda1-4或者sda1-4），逻辑分区从5开始，逻辑分区从扩展分区中划出。

MBR

MBR是整个硬盘最重要的区域，有512个字节，且可以分为两个部分：

(1)第一部分有446个字节，用于存放引导代码，即bootloader。（存放了部分代码，启动时会启动可启动盘，如/boot分区或/分区）

(2)第二部分有64个字节，用于存放磁盘分区表。其中，每个分区的信息需要用16个字节来记录，因此，一个硬盘最多可以有4个分区。这4个分区称之为主分区和扩展分区（extended）。

四、fdisk -ls命令解释

fdisk -ls命令实际上输出了磁盘信息，VG信息，以及LV信息

下面的关于/dev/vda磁盘信息的解释如下：

- ①磁盘的名称，以及一些物理信息，如大小，扇区
- ②这块磁盘上有两个主分区（编号为1,2.逻辑分区应当为5以后），/dev/vda1为引导分区
- ③该磁盘下的若干虚拟分区信息，此处涉及了Linux系统的Device mapper机制（参考：<http://www.ibm.com/developerworks/cn/linux/l-devmapper/>）

Device mapper机制（参考：<https://www.ibm.com/developerworks/cn/linux/l-devmapper/>）

```
[root@localhost boot]# fdisk -ls

Disk /dev/vda: 64.4 GB, 64424509440 bytes, 125829120 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000aae58

   Device Boot      Start         End      Blocks   Id  System
/dev/vda1 *          2048       2099199       1048576    83   Linux
/dev/vda2            2099200      125829119      61864960    8e   Linux LVM

Disk /dev/mapper/cl-root: 38.2 GB, 38235275264 bytes, 74678272 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/cl-swap: 6442 MB, 6442450944 bytes, 12582912 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/cl-home: 18.7 GB, 18668847104 bytes, 36462592 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk /dev/mapper/stack--volumes--lvmdriver--1-volume--e35aeeba--031c--44f7--8895--64773fb75c48: 1073 MB, 1073741824 bytes, 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/mapper/stack--volumes--lvmdriver--1-volume--bae7cba2--5cb3--43c8--9907--93ee489f68e1: 1073 MB, 1073741824 bytes, 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x00000000

                                Device Boot      Start         End      Blocks   Id  System
/dev/mapper/stack--volumes--lvmdriver--1-volume--bae7cba2--5cb3--43c8--9907--93ee489f68elp1 * 16065         80324        32130    83   Linux
```