



逻辑卷管理



本章内容

- **通用LVM概念和术语**
- **建立逻辑卷**
- **管理逻辑卷**
- **LVM高级应用**

LVM概述

- **Logical Volume Manager , 逻辑卷管理**
 - 它是Linux环境下对磁盘分区进行管理的一种机制
 - 屏蔽了底层磁盘布局 , 便于动态调整磁盘容量
 - 需要注意 :
 - /boot分区用于存放引导文件 , 不能应用LVM机制

LVM概述

- LVM机制的基本概念

- **PV** (Physical Volume , 物理卷)

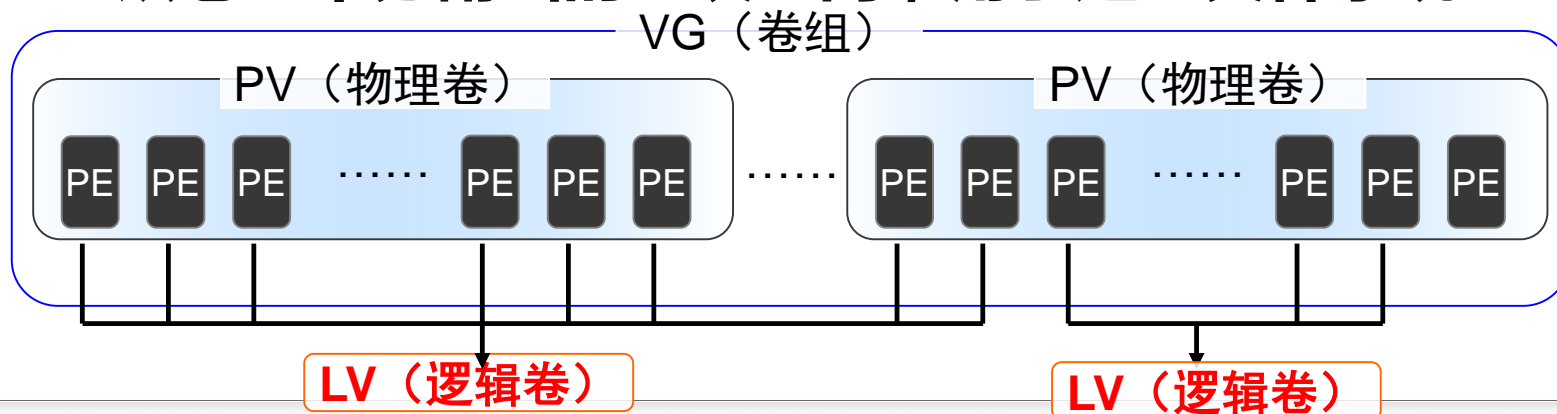
- 整个硬盘 , 或使用fdisk等工具建立的普通分区
- 包括许多默认4MB大小的**PE** (Physical Extent , 基本单元)

- **VG** (Volume Group , 卷组)

- 一个或多个物理卷组合而成的整体

- **LV** (Logical Volume , 逻辑卷)

- 从卷组中分割出的一块空间 , 用于建立文件系统



认识LVM的组成

Unused Space



4. Create logical volume (LV)



3. Create volume group (VG)



2. Create physical volume (PV)



1. Partition physical storage

安装逻辑卷命令行管理工具

❖ 安装 lvm2

```
[root@server0 Desktop]# yum install lvm2
已加载插件：langpacks
软件包 7: lvm2- 2.02.105- 14.el7.x86_64 已安装并且是最新版本
无须任何处理
[root@server0 Desktop]#
```

❖ 使用 lvm逻辑卷管理工具

```
[root@server0 Desktop]# lvm
```

```
lvm>
devtypes          lvreduce          pvresize          vgexport
dumpconfig        lvremove          pvs               vgextend
formats          lvrename          pvscan            vgimport
help              lvresize          segtypes          vgmerge
lvchange          lvs              tags              vgmknodes
lvconvert         lvscan            version           vgreduce
lvcreate          pvchange          vgcfgbackup       vgremove
lvdisplay         pvck              vgcfgrestore      vgrename
lvextend          pvcreate          vgchange           vgs
lvmchange         pvdata            vgck              vgscan
lvmddiskscan     pvdisplay         vgconvert          vgsplit
lvmsadc           pvmove            vgcreate
lvmsar            pvremove          vgdisplay
lvm>
```

创建逻辑卷

❖ 创建步骤

- (1) 新建空分区
- (2) 初始化分区 (创建PV)
- (3) 创建卷组
- (4) 创建逻辑卷

创建逻辑卷

❖ 创建步骤-新建空分区

```
[root@server0 Desktop]# fdisk -c /dev/vdb  
WARNING: fdisk GPT support is currently new, and therefore in an experimental phase. Use at your own discretion.  
欢迎使用 fdisk (util-linux 2.23.2)。
```

更改将停留在内存中，直到您决定将更改写入磁盘。
使用写入命令前请三思。

```
命令(输入 m 获取帮助) : n  
分区号 (2-128, 默认 2) :  
第一个扇区 (34-20971486, 默认 3395584) :  
Last sector, +sectors or +size{K,M,G,T,P} (3395584-20971486, 默认 20971486) : +512M  
已创建分区 2
```

I

```
命令(输入 m 获取帮助) : w
```


创建逻辑卷

❖ 创建步骤-查看分区

```
[root@server0 Desktop]# fdisk -l
```

```
/dev/vda1 * 2048 20970332 1048  
WARNING: fdisk GPT support is currently new,  
perimental phase. Use at your own discretion.
```

```
磁盘 /dev/vdb: 10.7 GB, 10737418240 字节, 209  
Units = 扇区 of 1 * 512 = 512 bytes  
扇区大小(逻辑/物理) : 512 字节 / 512 字节  
I/O 大小(最小/最佳) : 512 字节 / 512 字节  
磁盘标签类型 : gpt
```

#	Start	End	Size	Type
1	2048	3394626	1.6G	Linux f
2	3395584	4444159	512M	Linux f
3	4444160	5492735	512M	Linux f

创建逻辑卷

❖ 创建步骤-创建物理卷

格式: **pvcreate** 设备名

将空分区/dev/vdb2及/dev/vdb3 创建为物理卷

```
[root@server0 Desktop]# lvm
lvm> pvcreate /dev/vdb2
Physical volume "/dev/vdb2" successfully created
lvm> pvcreate /dev/vdb3
Physical volume "/dev/vdb3" successfully created
lvm> █
```

创建逻辑卷

❖ 创建步骤-创建卷组

格式：`vgcreate -s 块大小 卷组名 物理卷设备名`

■ 选项 `-s`

用途：指定扩展块的大小

```
lvm> vgcreate -s 8M wgroup /dev/vdb2
Volume group "wgroup" successfully created
lvm> █
```

创建逻辑卷

```
lvm> vgdisplay
--- Volume group ---
VG Name                wgroup
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   2
VG Access              read/write
VG Status              resizable
MAX LV                 0
Cur LV                1
Open LV               0
Max PV                 0
Cur PV                1
Act PV                 1
VG Size                504.00 MiB
PE Size                8.00 MiB
Total PE               63
```

创建逻辑卷

❖ 创建步骤-创建逻辑卷

格式: `lvcreate -n 【逻辑卷名】 -L 【逻辑卷大小】 【已存在卷组名】`

选项 '-n'

用途: 指定逻辑卷卷名

选项 '-L'

用途: 指定逻辑卷的大小

```
lvm> lvcreate -n lv1 -L 100m wgroup
Rounding up size to full physical extent 104.00 MiB
Logical volume "lv1" created
```


创建逻辑卷

```
lvm> lvdisplay
--- Logical volume ---
LV Path                /dev/wgroup/lv1
LV Name                 lv1
VG Name                 wgroup
LV UUID                 Y6JLe1- LujY- NgdM- 9EUe- Xdb8- eID
LV Write Access         read/write
LV Creation host, time server0.example.com, 2014-09-0
0800
LV Status                available
# open                  0
LV Size                  104.00 MiB
Current LE               13
Segments                 1
Allocation               inherit
Read ahead sectors      auto
- currently set to      8192
Block device             252:0
```

创建逻辑卷

❖ 创建步骤-格式化逻辑卷

格式：`mkfs -t 文件系统类型 逻辑卷设备名`

选项 `-t`

用途：指定文件系统类型，`type`的首字母

找出逻辑卷的位置

```
[root@server0 Desktop]# fdisk -l
```

```
磁盘 /dev/mapper/wgroup-lv1: 109 MB, 109051904 字节, 21299
Units = 扇区 of 1 * 512 = 512 bytes
扇区大小(逻辑/物理) : 512 字节 / 512 字节
I/O 大小(最小/最佳) : 512 字节 / 512 字节
```

格式化

创建逻辑卷

```
[root@server0 Desktop]# mkfs -t ext4 /dev/mapper/wgroup-lv1
mke2fs 1.42.9 (28-Dec-2013)
文件系统标签=
OS type: Linux
块大小=1024 (log=0)
分块大小=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
26624 inodes, 106496 blocks
5324 blocks (5.00%) reserved for the super user
第一个数据块=1
Maximum filesystem blocks=33685504
13 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: 完成
正在写入inode表: 完成
Creating journal (4096 blocks): 完成
```

管理逻辑卷

- ❖ 扩大逻辑卷及文件系统
- ❖ 缩小逻辑卷及文件系统

功能	物理卷管理	卷组管理	逻辑卷管理
Scan 扫描	pvscan	vgscan	lvscan
Create 建立	pvcreate	vgcreate	lvcreate
Display 显示	pvdisplay	vgdisplay	lvdisplay
Remove 删除	pvremove	vgremove	lvremove
Reduce 缩减		vgreduce	lvreduce
Extend 扩展		vgextend	lvextend

主要命令的用法

pvcreate 设备名

vgcreate -s 8M 卷组名 物理卷名1 物理卷名2

lvcreate -L 大小 -n 逻辑卷名 卷组名

lvextend -L +大小 /dev/卷组名/逻辑卷名

lvreduce -L +大小 /dev/卷组名/逻辑卷名

❖ 扩大逻辑卷及文件系统

■ 确定卷组大小是否满足逻辑卷扩展需求

```
lvm> vgdisplay
--- Volume group ---
VG Name                wgroup
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   2
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 1
Open LV                 0
Max PV                 0
Cur PV                 1
Act PV                 1
VG Size                 504.00 MiB
PE Size                 8.00 MiB
Total PE                63
Alloc PE / Size        13 / 104.00 MiB
Free PE / Size          50 / 400.00 MiB
VG UUID                 0leznC-NdDw-tpyL-oYwz-7BXN-0osF-2tUE96
```

管理逻辑卷

❖ 扩大逻辑卷及文件系统

- 扩展逻辑卷的容量大小
- 命令格式

Lvextend -L 【扩展大小】 【逻辑卷设备名】

```
lvm> lvextend -L 200M /dev/mapper/wgroup-lv1
Extending logical volume lv1 to 200.00 MiB
Logical volume lv1 successfully resized
lvm> lvdisplay
--- Logical volume ---
LV Path                /dev/wgroup/lv1
LV Name                 lv1
VG Name                wgroup
LV UUID                Y6JLeI- LujY- NgdM- 9EUe- Xdb8- eIDn- Yoyhhj
LV Write Access         read/write
LV Creation host, time server0.example.com, 2014- 09- 01  22: 10: 23
LV Status               available
# open                  0
LV Size                 200.00 MiB
Current LE              25
Segments                1
Allocation              inherit
Read ahead sectors     auto
- currently set to     8192
Block device            252: 0
```

❖ 扩大逻辑卷及文件系统

- 扩展文件系统大小
- 格式

resize2fs -p 逻辑卷设备名

```
[root@server0 Desktop]# resize2fs -p /dev/mapper/wgroup-lv1
resize2fs 1.42.9 (28-Dec-2013)
Resizing the filesystem on /dev/mapper/wgroup-lv1 to 204800 (1k) blocks.
The filesystem on /dev/mapper/wgroup-lv1 is now 204800 blocks long.
```

管理逻辑卷

❖ 缩小逻辑卷及文件系统

- 缩小文件系统
- 注意缩小时必须将逻辑卷卸载并确定数据使用量

```
[root@server0 Desktop]# umount /dev/mapper/wgroup- lv1
umount: /dev/mapper/wgroup- lv1: 未挂载
[root@server0 Desktop]# fsck -f /dev/mapper/wgroup- lv1
fsck, 来自 util-linux 2.23.2
e2fsck 1.42.9 (28-Dec-2013)
第一步: 检查 inode, 块, 和大小
第二步: 检查目录结构
第3步: 检查目录连接性
Pass 4: Checking reference counts
第5步: 检查簇概要信息
/dev/mapper/wgroup- lv1: 11/51200 files (9.1% non- contiguous), 12109/2048
ocks
```

管理逻辑卷

```
[root@server0 Desktop]# resize2fs -p /dev/mapper/wgroup-lv1  
resize2fs 1.42.9 (28-Dec-2013)  
The filesystem is already 204800 blocks long.  Nothing to do!
```



```
[root@server0 Desktop]# resize2fs -p /dev/mapper/wgroup-lv1 100M  
resize2fs 1.42.9 (28-Dec-2013)  
Resizing the filesystem on /dev/mapper/wgroup-lv1 to 102400 (1k) blocks.  
Begin pass 3 (max = 25)  
正在扫描inode表XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
The filesystem on /dev/mapper/wgroup-lv1 is now 102400 blocks long.
```


管理逻辑卷

❖ 缩小逻辑卷及文件系统

- 缩小逻辑卷容量
- 命令格式

lvreduce -L 100M /dev/mapper/wgroup/lv1

```
[ root@server0 Desktop]# lvreduce -L 100M /dev/wgroup/lv1
Rounding size to boundary between physical extents: 104.00 MiB
WARNING: Reducing active logical volume to 104.00 MiB
THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lv1? [y/n]: y
Reducing logical volume lv1 to 104.00 MiB
Logical volume lv1 successfully resized
[ root@server0 Desktop]#
```

实验：创建并使用LVM逻辑卷

●需求描述

- 创建一个VG和LV,PE大小8M，LV中PE个数100，开机时自动挂载到/mnt/wshare 格式化为ext4文件系统，其中Vg名称为wgroup,lv名称为wshare
- 新建一个LV，名称为home，大小为100M，文件系统为ext4，永久挂载到/test，将home逻辑卷从100M缩小为50M

❖ 修改逻辑卷文件系统

1、查看当前文件系统

```
df -hT
```

2、卸载文件系统

```
umount /mnt/lv
```

3、重新格式化

```
mkfs -t vfat /dev/group/lv
```

4、修改fstab文件

```
vim /etc/fstab
```

5、重新挂载

```
mount -a
```

LVM应用示例

- 推荐步骤：

- PV → VG → LV → 格式化，挂载使用文件系统

/dev/sdb



/dev/sdc

/dev/sdb1
/dev/sdc1

转换物理卷

mail_store

创建卷组

/dev/mail_store/mail

创建逻辑卷

格式化为ext4文件系统，
并挂载到 /mail 目录

实验：创建并使用逻辑卷

●需求描述

- 新建一个大小为500M的VG（VG的名字为group_test；PE大小为16M），在该卷组上新建一个大小为20个PE的LV（LV的名字为test11，格式化为ext4的文件系统，永久挂载到/mnt/test目录）
- 将逻辑卷test11的大小扩大到800M
- 将逻辑卷test11重新格式化为vfat文件系统，并实现永久挂载

LVM应用示例

- **需求描述：**

- **公司的邮件服务器由于用户数量众多，邮件存储需要大量的空间，考虑到动态扩容的需要，计划增加两块SCSI硬盘并构建LVM逻辑卷，挂载到“/mail”目录专门用于存放邮件数据**

创建快照来促进数据备份

- **快照是临时保留所更改的逻辑卷的原始数据的逻辑卷**
- **快照提供原始卷的静态视图，从而能够以一致状态备份其数据**
- **确定快照的大小**
 1. **预期更改率**
 2. **所需快照时间**

创建快照来促进数据备份

- **创建和使用快照进行备份**

- 1.创建lvname名为snaplvname,大小为20M的新快照卷

```
lvcreate -s -n snaplv -L 20M /dev/vgname/lvname
```

- 2.挂载快照，并使备份程序指向新挂载点

```
mkdir /snapmount
```

```
mount -ro /dev/vgname/snaplv /snapmount
```

- 3.验证快照逻辑卷的状态

```
lvscan /dev/vgname/lvname
```

- 4.不再使用快照时，解除挂载并删除

```
umount /snapmount
```

```
lvremove /dev/vgname/lvname
```

● 需求描述

- Allison需要存储业务数据。她的客户数据库目前大小为256M。在普通工作日，数据库中的数据每小时大约更改10MB。备份软件需要10分钟才能完成完全备份。
- 创建名为allison的新卷组，使其空间足够容纳512 MB的卷和为备份软件创建的该卷快照。
- 在创建卷组之后，为Allison的客户数据库创建大小为512MB、名为custdb的逻辑卷。还为Allison的备份软件创建名为custdbsnap的客户数据库快照卷。