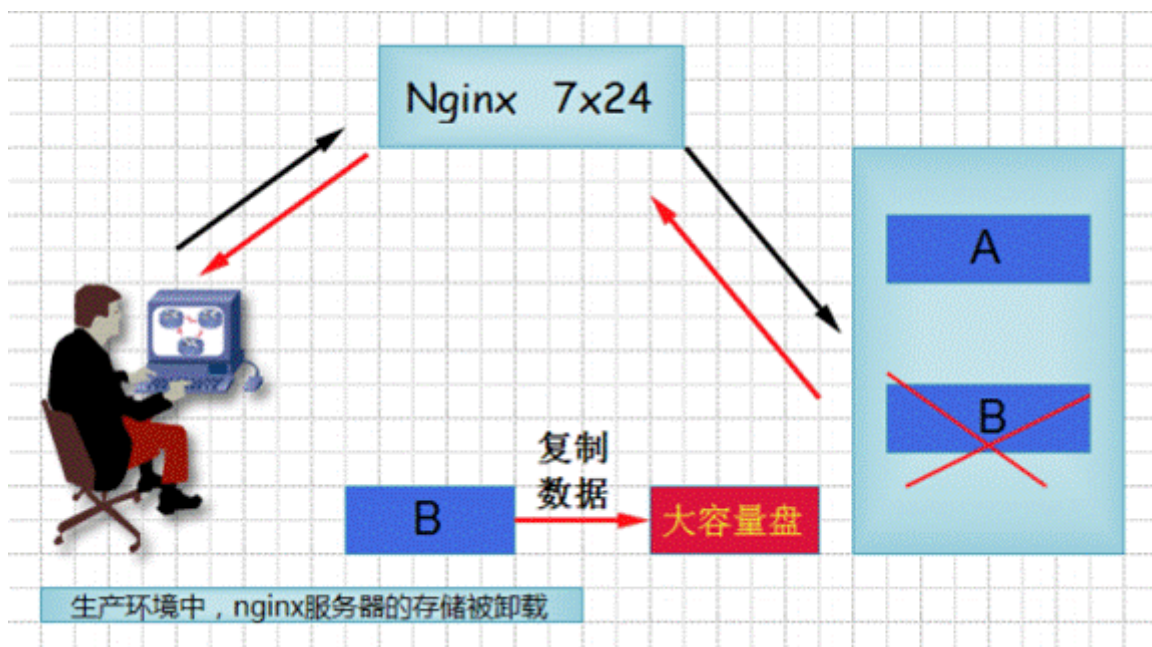


1.LVM 的基本概念

实战场景：

对于生产环境下的服务器来说,如果存储数据的分区磁盘空间不够了怎么办?



答：只能换一个更大的磁盘。如果用了一段时间后，空间又不够了，怎么办？再加一块更大的？换磁盘的过程中，还需要把数据从一个硬盘复制到另一个硬盘，过程太慢了。

解决方案: 使用LVM在线动态扩容

2.LVM的工作原理

LVM (LogicalVolumeManager) 逻辑卷管理，是在磁盘分区和文件系统之间添加的一个逻辑层，来为文件系统屏蔽下层磁盘分区布局，提供一个抽象的盘卷，在盘卷上建立文件系统。管理员利用LVM可以在磁盘不用重新分区的情况下动态调整文件系统的大小，并且利用LVM管理的文件系统可以跨越磁盘，当服务器添加了新的磁盘后，管理员不必将原有的文件移动到新的磁盘上，而是通过LVM可以直接扩展文件系统跨越磁盘

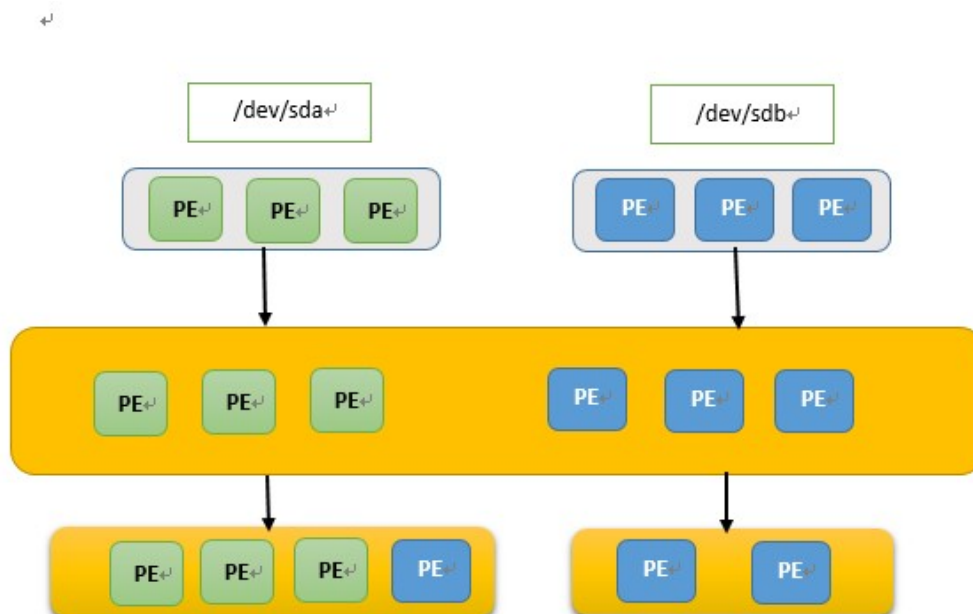
它就是通过将底层的物理硬盘封装起来，然后以逻辑卷的方式呈现给上层应用。在LVM中，其通过对底层的硬盘进行封装，当我们对底层的物理硬盘进行操作时，其不再是针对于分区进行操作，而是通过一个叫做逻辑卷的东西来对其进行底层的磁盘管理操作。

LVM的原理

要想理解好LVM的原理，我们必须首先要掌握4个基本的逻辑卷概念。

- ① **PE** (Physical Extend) 物理拓展
- ② **PV** (Physical Volume) 物理卷
- ③ **VG** (Volume Group) 卷组
- ④ **LV** (Logical Volume) 逻辑卷

我们知道在使用LVM对磁盘进行动态管理以后，我们是以逻辑卷的方式呈现给上层的服务的。所以我们所有的操作目的，其实就是去创建一个**LV(Logical Volume)**,逻辑卷就是用来取代我们之前的分区，我们通过对逻辑卷进行格式化，然后进行挂载操作就可以使用了。那么LVM的工作原理是什么呢？所谓无图无真相，咱们下面通过图来对逻辑卷的原理进行解释！！



1.将我们的物理硬盘格式化成**PV**(Physical Volume)

我们看到，这里有两块硬盘，一块是sda，另一块是sdb，在LVM磁盘管理里，我首先要将这两块硬盘格式化为我们的PV(Physical Volume),也就是我们的物理卷，其实格式化物理卷的过程中LVM是将底层的硬盘划分为了一个一个的**PE**(Physical Extend),我们的**LVM磁盘管理中PE的默认大小是4M大小**，其实PE就是我们逻辑卷管理的最基本单位。比如说我有一个

400M的硬盘，那么在将其格式化PV的时候，其实际就是将这块物理硬盘划分成了100个的PE，因为PE默认的大小就是4M。这个就是我们的第一步操作。

2.创建一个VG(Volume Group)

在将硬盘格式化PV以后，我们第二步操作就是创建一个卷组，也就是**VG(Volume Group)**，卷组在这里我们可以将其抽象化成一个空间池，VG的作用就是用来装PE的，我们可以把一个或者多个PV加到VG当中，因为在第一步操作时就已经将该硬盘划分成了多个PE，所以将多个PV加到VG里面后，VG里面就存放了许许多多来自不同PV中的PE，我们通过上面的图片就可以看到，我们格式化了两块硬盘，每个硬盘分别格式化成了3个PE，然后将两块硬盘的PE都加到了我们的VG当中，那么我们的VG当中就包含了6个PE，这6个PE就是两个硬盘的PE之和。通常创建一个卷组的时候我们会为其取一个名字，也就是该VG的名字。

3.基于VG创建我们最后要使用的LV(Logical Volume)

[注意] PV以及VG创建好以后我们是不能够直接使用的，因为PV、VG是我们逻辑卷底层的東西，我们其实最后使用的是在VG基础上创建的**LV(Logical Volume)**，所以第三步操作就是基于VG来创建我们最终要使用的LV。

当我们创建好我们的VG以后，这个时候我们创建LV其实就是从VG中拿出我们指定数量的PE，还是拿上图来说，我们看到我们此时的VG里面已经拥有了6个PE，这时候我们创建我们的第一个逻辑卷，它的大小是4个PE的大小，也就是16M(**因为一个PE的默认大小是4M**)，而这4个PE有三个是来自于第一块硬盘，而另外一个PE则是来自第二块硬盘。当我们创建第二个逻辑卷时，它的大小就最多只有两个PE的大小了，因为其中的4个PE已经分配给了我们的第一个逻辑卷。

所以创建逻辑卷其实就是从VG中拿出我们指定数量的PE，VG中的PE可以来自不同的PV，我们可以创建的逻辑卷的大小取决于VG当中PE存在的数量，并且我们创建的逻辑卷其大小一定是PE的整数倍(即逻辑卷的大小一定要是4M的整数倍)。

4.将我们创建好的LV进行文件系统的格式化，然后挂载使用

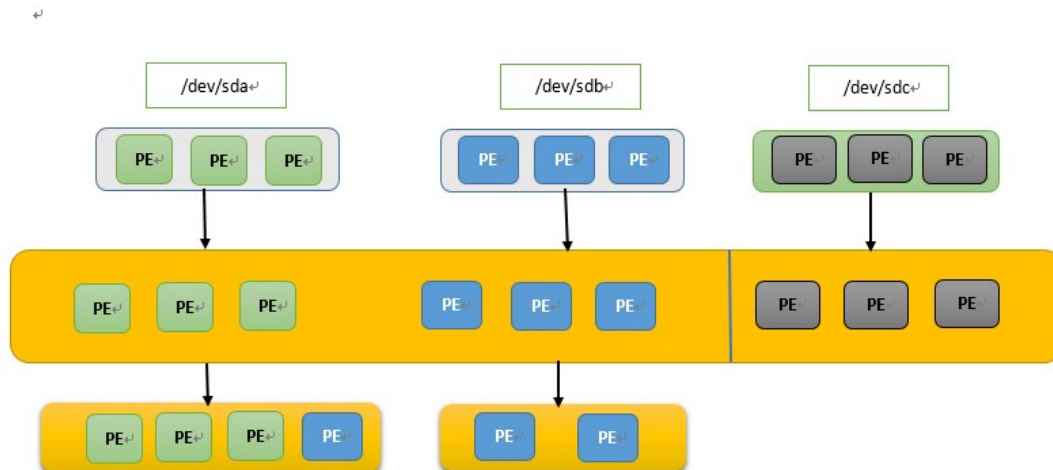
在创建好LV以后，这个时候我们就能够对其进行文件系统的格式化了，我们最终使用的就是我们刚创建好的LV，其就相当于传统的文件管理的分区，我们首先要对其进行文件系统的格式化操作，然后通过mount命令对其进行挂载，这个时候我们就能够像使用平常的分区一样来使用我们的逻辑

辑卷了。

我们在创建好LV以后，我们会在 /dev 目录下看到我们的LV信息，例如 **/dev/vgname/lvname**，我们每创建一个VG，其会在/dev目录下创建一个以该VG名字命名的文件夹，在该VG的基础上创建好LV以后，我们会在这个VG目录下多出一个以LV名字命名的逻辑卷。

下面我们来对整个LVM的工作原理进行一个总结：

- (1)物理磁盘被格式化为PV，空间被划分为一个个的PE
- (2)不同的PV加入到同一个VG中，不同PV的PE全部进入到了VG的PE池内
- (3)LV基于PE创建，大小为PE的整数倍，组成LV的PE可能来自不同的物理磁盘
- (4)LV现在就直接可以格式化后挂载使用了
- (5)LV的扩充缩减实际上就是增加或减少组成该LV的PE数量，其过程不会丢失原始数据



我们看到，我们这里如果要对LV进行扩充，直接加进来一块sdc硬盘，然后将其格式化成PE，然后将该PV加入到了VG当中，这个时候我们就可以通过增加LV中PE的数量来动态的对LV进行扩充了，只要我们的LV的大小不要超过我们VG空余空间的大小就行了！！

3.LVM常用的术语

物理存储介质（The physical media）：LVM存储介质可以是磁盘分区，整个磁盘，RAID阵列或SAN磁盘，设备必须初始化为LVM物理卷，才能与LVM结合使用

物理卷**PV** (physicalvolume)：物理卷就是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,创建物理卷它可以用硬盘分区，也可以用硬盘本身；

卷组**VG** (VolumeGroup)：一个LVM卷组由一个或多个物理卷组成

逻辑卷**LV** (logicalvolume)：LV建立在VG之上，可以在LV之上建立文件系统

PE (physicalextents)：**PV物理卷**中可以分配的最小存储单元，PE的大小是可以指定的，默认为4MB

LE (logicaextent)：**LV逻辑卷**中可以分配的最小存储单元，在同一个卷组中，LE的大小和PE是相同的，并且一一对应

最小存储单位总结：

名称 最小存储单位

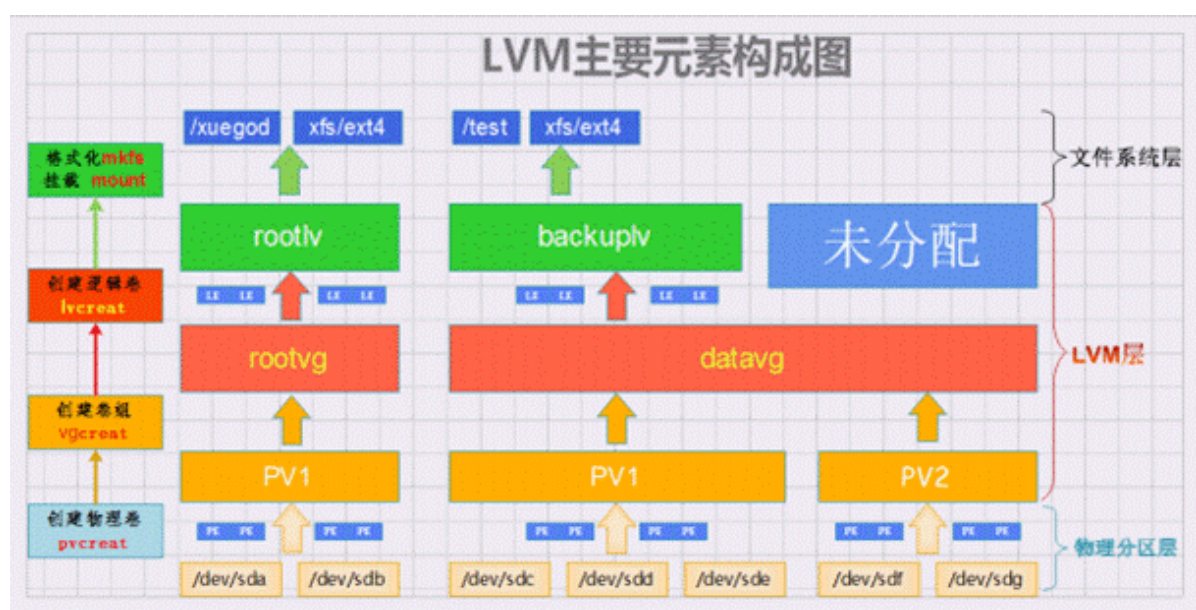
硬盘 扇区 (512字节)

文件系统 block (1K或4K) #mkfs.ext4-b2048/dev/sdb1，最大支持到4096

raid chunk (512K) #mdadm-C-v/dev/md5-l5-n3-c512-x1/dev/sde{1,2,3,5}

LVM PE (4M) #vgcreate-s4Mvg1/dev/sdb{1,2}

4.LVM主要元素构成：



总结：多个磁盘/分区/raid > 多个物理卷PV > 合成卷组VG > 从VG划出逻辑卷LV > 格式化LV挂载使用

5.LVM优点:

使用卷组，使多个硬盘空间看起来像是一个大的硬盘

使用逻辑卷，可以跨多个硬盘空间的分区sdb1 sdb2 sdc1 sdd2 sdf

在使用逻辑卷时，它可以在空间不足时动态调整它的大小

在调整逻辑卷大小时，不需要考虑逻辑卷在硬盘上的位置，不用担心没有可用的连续空间

可以在线对LV,VG进行创建，删除，调整大小等操作。LVM上的文件系统也需要重新调整大小。

允许创建快照，可以用来保存文件系统的备份。

RAID+LVM一起用：LVM是软件的卷管理方式，而RAID是磁盘管理的方法。对于重要的数据，使用RAID用来保护物理的磁盘不会因为故障而中断业务，再用LVM用来实现对卷的良性的管理，更好的利用磁盘资源。

6.创建LVM的基本步骤:

1)物理磁盘被格式化为PV，(空间被划分为一个个的PE) **#PV包含PE**

2)不同的PV加入到同一个VG中，(不同PV的PE全部进入到了VG的PE池内)
#VG包含PV

3)在VG中创建LV逻辑卷，基于PE创建，(组成LV的PE可能来自不同的物理磁盘) **#LV基于PE创建**

4)LV直接可以格式化后挂载使用 **#格式化挂载使用**

5)LV的扩充缩减实际上就是增加或减少组成该LV的PE数量，其过程不会丢失原始数据

7.lvm常用的命令

功能	PV管理命令	VG管理命令	LV管理命令
scan扫描	pvscan	vgscan	lvscan
create创建	pvcreate	vgcreate	lvcreate
display显示	pvdisplay	vgdisplay	lvdisplay

display显示 功能 remove移除	pvdisplay PV管理命令 pvremove	vgdisplay VG管理命令 vgremove	lvdisplay LV管理命令 lvremove
extend扩展		vgextend	lvextend
reduce减少		vgreduce	lvreduce

下面的操作会用的一些查看命令

查看 卷名	简单对应卷信息 的查看	扫描相关的所有的 对应卷	详细对应卷信息 的查看
物理 卷	pvs	pvscan	pvdisplay
卷组	vgs	vgscan	vgdisplay
逻辑 卷	lvs	lvscan	lvdisplay

8.创建并使用LVM逻辑卷

创建PV

添加一个sdb磁盘

需要注意，给一个磁盘分区到准备分第四个区的时候默认选项不再是p(即不再是主分区)而是e(即是扩展分区)

```
命令(输入 m 获取帮助): n
Partition type:
   p   primary (3 primary, 0 extended, 1 free)
   e   extended
Select (default e): p
已选择分区 4
起始 扇区 (6293504-41943039, 默认为 6293504):
将使用默认值 6293504
Last 扇区, +扇区 or +size{K,M,G} (6293504-41943039, 默认为 41943039): +1G
分区 4 已设置为 Linux 类型, 大小设为 1 GiB
```

```
1 [root@exercise1 ~]# fdisk /dev/sdb    #创建4个主分区，每个分区1G
2 [root@exercise1 ~]# ls /dev/sdb*
3 /dev/sdb  /dev/sdb1  /dev/sdb2  /dev/sdb3  /dev/sdb4
```

设定分区类型代码：`fdisk /dev/sdb=>t=>选择分区号==>8e==>w`
注：现在系统已经很智能了，直接使用默认的83Linux分区，也可以创建pv的。

需要安装软件包lvm2

```
1 [root@exercise1 ~]# yum -y install lvm2
```

```
[root@exercise1 ~]# pvcreate /dev/sdb{1,2,3,4} #创建pv
Physical volume "/dev/sdb1" successfully created.
Physical volume "/dev/sdb2" successfully created.
Physical volume "/dev/sdb3" successfully created.
Physical volume "/dev/sdb4" successfully created.
[root@exercise1 ~]#
```

```
[root@exercise1 ~]# pvdisplay /dev/sdb1 #查看物理卷信息
"/dev/sdb1" is a new physical volume of "1.00 GiB"
--- NEW Physical volume ---
PV Name          /dev/sdb1
VG Name
PV Size          1.00 GiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          jKJdlf-jOpy-8PaP-h3Wu-H9Jd-p5zg-ouxlPr
```

创建vg卷组

语法：`vgcreate vg名字 pv的名字`


```
1 [root@exercise1 ~]# vgcreate vg01 /dev/sdb1 #创建vg
2 volume group "vg01" successfully created
```

[root@exercise1 ~]# vgs #查看卷组信息

VG	#PV	#LV	#SN	Attr	VSize	VFree
vg01	1	0	0	wz--n-	1020.00m	1020.00m

[root@exercise1 ~]# vgdisplay vg01 #查看具体某卷组信息

--- Volume group ---

VG Name vg01

System ID

Format lvm2

Metadata Areas 1

Metadata Sequence No 1

VG Access read/write

VG Status resizable

MAX LV 0

Cur LV 0

Open LV 0

Max PV 0

Cur PV 1

Act PV 1

VG Size 1020.00 MiB

PE Size 4.00 MiB

Total PE 255

Alloc PE / Size 0 / 0

Free PE / Size 255 / 1020.00 MiB

VG UUID 7PGG2H-96ct-Jc2p-SDY2-0tZR-ow11-SjcAXD

创建LV

语法: **lvcreate -n 指定新逻辑卷的名称 -L指定lv大小的SIZE(M,G)(-l:小指定PE的数量) vg名**

```
1 [root@exercise1 ~]# lvcreate -n lv01 -L16M vg01
2 Logical volume "lv01" created.
```

```
[root@exercise1 ~]# lvcreate -n lv02 -l4 vg01 #默认一个PE 4M
Logical volume "lv02" created.
```

```
[root@exercise1 ~]# lvs #查看逻辑卷的信息
LV VG Attr LSize Pool Origin Data% Meta% Move Log
Cpy%Sync Convert
lv01 vg01 -wi-a----- 16.00m
lv02 vg01 -wi-a----- 16.00m
```

```
[root@exercise1 ~]# pvdisplay /dev/sdb1 #查看物理卷信息
--- Physical volume ---
PV Name          /dev/sdb1
VG Name          vg01
PV Size          1.00 GiB / not usable 4.00 MiB
Allocatable      yes
PE Size          4.00 MiB
Total PE         255
Free PE          247
Allocated PE     8
PV UUID          jKJdlf-jOpy-8PaP-h3Wu-H9Jd-p5zg-ouxlPr
```

```
[root@exercise1 ~]#
```

9. 文件系统格式与挂载

```
1 [root@exercise1 ~]# mkdir /opt/test1
```

互动：lv01逻辑卷的路径在哪？

/dev/mapper/vg01-lv01

```
1 [root@exercise1 ~]# ls /dev/vg01    #查看逻辑卷
2 lv01  lv02
```

```
[root@exercise1 ~]# ll /dev/vg*    #其实lv01是dm-0的软链接
crw-----. 1 root root 10, 63 2月  3 10:28 /dev/vga_arbiter
```

```
/dev/vg01:
```

```
总用量 0
```

```
lrwxrwxrwx. 1 root root 7 2月  3 10:52 lv01 -> ../dm-0
```

```
lrwxrwxrwx. 1 root root 7 2月  3 10:52 lv02 -> ../dm-1
```

```
[root@exercise1 ~]#
```

```
[root@exercise1 ~]# ls /dev/vg01
lv01  lv02
[root@exercise1 ~]# ll /dev/vg*
crw-----. 1 root root 10, 63 2月  3 10:28 /dev/vga_arbiter

/dev/vg01:
总用量 0
lrwxrwxrwx. 1 root root 7 2月  3 10:52 lv01 -> ../dm-0
lrwxrwxrwx. 1 root root 7 2月  3 10:52 lv02 -> ../dm-1
```

```

1 [root@exercise1 ~]# mkfs.xfs /dev/vg01/lv01
2 meta-data=/dev/vg01/lv01          isize=512
   agcount=1, agsize=4096 blks
3           =                        sectsz=512   attr=2,
   projid32bit=1
4           =                        crc=1        finobt=0,
   sparse=0
5 data      =                        bsize=4096
   blocks=4096, imaxpct=25
6           =                        sunit=0      swidth=0
   blks
7 naming    =version 2              bsize=4096   ascii-
   ci=0 ftype=1
8 log       =internal log          bsize=4096
   blocks=855, version=2
9           =                        sectsz=512   sunit=0
   blks, lazy-count=1
10 realtime =none                  extsz=4096   blocks=0,
   rtextents=0

```

```
[root@exercise1 ~]# mount /dev/vg01/lv01 /opt/test1/
```

```
[root@exercise1 ~]# df -h
```

文件系统	容量	已用	可用	已用%	挂载点
/dev/sda3	18G	2.3G	16G	13%	/
devtmpfs	479M	0	479M	0%	/dev
tmpfs	489M	0	489M	0%	/dev/shm
tmpfs	489M	6.8M	482M	2%	/run
tmpfs	489M	0	489M	0%	/sys/fs/cgroup
/dev/sr0	4.3G	4.3G	0	100%	/mnt
/dev/sda1	197M	97M	100M	50%	/boot
tmpfs	98M	0	98M	0%	/run/user/0
/dev/mapper/vg01-lv01	13M	896K	12M	7%	/opt/test1

```

[root@exercise1 ~]# echo "/dev/mapper/vg01-lv01 /opt/test1 xfs
defaults 0 0" >> /etc/fstab

```

10.指定PE大小用

指定PE大小用的参数: -s,如果存储的数据都是大文件, 那么PE尽量调大, 读取速度快

PE的大小只有为2的幂数, 且最大为512M

```
1 [root@exercise1 ~]# vgcreate -s 16M vg02 /dev/sdb2
2   volume group "vg02" successfully created
```

```
[root@exercise1 ~]# vgdisplay vg02
--- Volume group ---
VG Name                vg02
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                 0
Cur PV                 1
Act PV                 1
VG Size                 1008.00 MiB
PE Size                 16.00 MiB
Total PE                63
Alloc PE / Size         0 / 0
Free PE / Size          63 / 1008.00 MiB
VG UUID                 ZYR8n5-BTmK-6bAn-zsPW-gDOL-YmOe-IGmoco
```

11.LV扩容

首先，确定一下是否有可用的扩容空间，因为空间是从VG里面创建的，并且LV不能跨VG扩容

```
1 [root@exercise1 ~]# vgs
2   VG      #PV #LV #SN Attr   VSize   VFree
3   vg01     1   2   0 wz--n- 1020.00m 988.00m
4   vg02     1   0   0 wz--n- 1008.00m 1008.00m
```

用的命令如下

扩展	vg扩容	lv扩容
extend扩展	vgextend	lvextend

扩容逻辑卷

```
1 [root@exercise1 ~]# lvs
2   LV      VG      Attr      LSize   Pool Origin Data%  Meta%
3   lv01    vg01    -wi-ao---- 48.00m
4   lv02    vg01    -wi-a----- 32.00m
```

```
[root@exercise1 ~]# lvextend -L +50M /dev/vg01/lv01
Rounding size to boundary between physical extents: 52.00 MiB.
Size of logical volume vg01/lv01 changed from 48.00 MiB (12
extents) to 100.00 MiB (25 extents).
Logical volume vg01/lv01 successfully resized.
```

```
[root@exercise1 ~]# lvs
LV      VG      Attr      LSize   Pool Origin Data%  Meta%  Move Log
Cpy%Sync Convert
lv01    vg01    -wi-ao---- 100.00m
lv02    vg01    -wi-a----- 32.00m
```

```
[root@exercise1 ~]# lvextend -L 50M /dev/vg01/lv02
Rounding size to boundary between physical extents: 52.00 MiB.
Size of logical volume vg01/lv02 changed from 32.00 MiB (8
extents) to 52.00 MiB (13 extents).
Logical volume vg01/lv02 successfully resized.

[root@exercise1 ~]# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log
Cpy%Sync Convert
lv01 vg01 -wi-ao---- 100.00m
lv02 vg01 -wi-a----- 52.00m
[root@exercise1 ~]#
```

说明：在指定大小的时候，扩容50m和扩容到50m是不一样的写法

扩容50m==>-L +50M

扩容到50m====>-L 50M

而且是按照2的倍数呈现，并不是直接这样加上去，通过上面例子可以看出

```
1 [root@exercise1 ~]# df -Th /dev/mapper/vg01-lv01
2 文件系统                类型  容量  已用  可用  已用% 挂载点
3 /dev/mapper/vg01-lv01 xfs      13M   896K   12M    7%
  /opt/test1
```

注：可以看到LV虽然扩展了，但是文件系统大小还是原来的，下面开始扩容文件系统

ext4文件系统扩容使用命令语法：resize2fs 逻辑卷名称；xfs文件系统

扩容使用命令语法：xfs_growfs 挂载点

resize2fs和xfs_growfs两者的区别是传递的参数不一样的，xfs_growfs是采用的挂载点；resize2fs是逻辑卷名称，而且resize2fs命令不能对xfs类型文件系统使用

```

1 [root@exercise1 ~]# xfs_growfs /opt/test1/
2 meta-data=/dev/mapper/vg01-lv01  isize=512
   agcount=1, agsize=4096 blks
3         =                               sectsz=512   attr=2,
   projid32bit=1
4         =                               crc=1       finobt=0
   spinodes=0
5 data      =                               bsize=4096
   blocks=4096, imaxpct=25
6         =                               sunit=0     swidth=0
   blks
7 naming    =version 2                     bsize=4096   ascii-
   ci=0 ftype=1
8 log       =internal                      bsize=4096
   blocks=855, version=2
9         =                               sectsz=512   sunit=0
   blks, lazy-count=1
10 realtime =none                          extsz=4096   blocks=0,
   rtextents=0
11 data blocks changed from 4096 to 25600

```

```
[root@exercise1 ~]# df -h /opt/test1/
```

```
文件系统      容量  已用  可用  已用%  挂载点
```

```
/dev/mapper/vg01-lv01  97M  1.1M  96M   2% /opt/test1
```

```
[root@exercise1 ~]#
```

[root@base ~]# lvextend -L80M -r /dev/vg01/lv01 # -r 参数:直接扩容到80M空间，一步到位，不用再扩文件系统了

[root@base ~]# lvextend -L80M -r /dev/vg01/lv01 #直接扩容到80M空间

```
[root@base2 ~]# df -h /opt/test1
```

```
文件系统      容量  已用  可用  已用%  挂载点
```

```
/dev/mapper/vg01-lv01  97M  1.1M  96M   2% /opt/test1
```

注意:

上面例子使用了df -T,其中-T, --print-type 显示文件系统的形式

df -TH 查看磁盘大小，解决：删除比较大无用的文件

df -i 查看inode：文件的字节数，拥有者id，组id，权限，改动时间，链接数，数据block的位置，解决：删除数量过多的小文件

12.VG扩容

```
1 [root@exercise1 ~]# vgs
2   VG      #PV #LV #SN Attr   VSize   VFree
3   vg01    1   2   0 wz--n- 1020.00m 868.00m
4   vg02    1   0   0 wz--n- 1008.00m 1008.00m
5 [root@exercise1 ~]#
```

vg扩容的场景：vg卷组中的空间不够，需要添加新的硬盘进来

```
1 [root@exercise1 ~]# pvcreate /dev/sdb3
2   Physical volume "/dev/sdb3" successfully created.
```

```
[root@exercise1 ~]# vgextend vg01 /dev/sdb3
Volume group "vg01" successfully extended
```

```
[root@exercise1 ~]# vgs
VG      #PV #LV #SN Attr   VSize   VFree
vg01    2   2   0 wz--n- 1.99g   1.84g
vg02    1   0   0 wz--n- 1008.00m 1008.00m
[root@exercise1 ~]#
```

13.LVM缩小(需要用到ext4)

13.1 lv缩小

互动：LVM可以动态增加，可以动态缩小吗？

答：LVM可以动态增加，也可以动态缩小，但是XFS不支持动态缩小，所以我们无法实现基于xfs的动态缩小。btrfs文件系统支持在线缩小。

```

1 [root@exercise1 ~]# lvreduce -L -20M /dev/vg01/lv01
2   WARNING: Reducing active and open logical volume to
   80.00 MiB.
3   THIS MAY DESTROY YOUR DATA (filesystem etc.)
4 Do you really want to reduce vg01/lv01? [y/n]: y
5   Size of logical volume vg01/lv01 changed from 100.00
   MiB (25 extents) to 80.00 MiB (20 extents).
6   Logical volume vg01/lv01 successfully resized.

```

但是文件系统没有缩小成功：

```

1 [root@exercise1 ~]# df -h /opt/test1/
2 文件系统              容量  已用   可用 已用% 挂载点
3 /dev/mapper/vg01-lv01  97M   1.1M   96M    2% /opt/test1
   #发现文件系统上空间没有变

```

```
[root@exercise1 ~]# lvextend -L10M -r /dev/vg01/lv01
```

Rounding size to boundary between physical extents: 12.00 MiB.

New size given (3 extents) not larger than existing size (20 extents)

[root@exercise1 ~]# xfs_growfs /dev/vg01/lv01 #这两个命令也是不能执行成功的

meta-data=/dev/mapper/vg01-lv01 isize=512 agcount=7,
agsize=4096 blks

= sectsz=512 attr=2, projid32bit=1

= crc=1 finobt=0 spinodes=0

data = bsize=4096 blocks=25600, imaxpct=25

= sunit=0 swidth=0 blks

naming =version 2 bsize=4096 ascii-ci=0 ftype=1

log =internal bsize=4096 blocks=855, version=2

= sectsz=512 sunit=0 blks, lazy-count=1

realtime =none extsz=4096 blocks=0, rtextents=0

data size 20480 too small, old size is 25600

13.2 vg缩小

VG的缩减，要保证你的物理卷是否被使用，是因为它无法缩减一个正在使用的PV


```
1 [root@exercise1 ~]# umount /opt/test1/ #卸载挂载
```

```
[root@exercise1 ~]# df
```

文件系统	1K-块	已用	可用	已用%	挂载点
/dev/sda3	18658304	2312184	16346120	13%	/
devtmpfs	490016	0	490016	0%	/dev
tmpfs	499848	0	499848	0%	/dev/shm
tmpfs	499848	6892	492956	2%	/run
tmpfs	499848	0	499848	0%	/sys/fs/cgroup
/dev/sr0	4414592	4414592	0	100%	/mnt
/dev/sda1	201380	99128	102252	50%	/boot
tmpfs	99972	0	99972	0%	/run/user/0

```
[root@exercise1 ~]# lvreduce -L -20m /dev/vg01/lv01 #这样就是成功
```

WARNING: Reducing active logical volume to 60.00 MiB.

THIS MAY DESTROY YOUR DATA (filesystem etc.)

Do you really want to reduce vg01/lv01? [y/n]: y

Size of logical volume vg01/lv01 changed from 80.00 MiB (20 extents) to 60.00 MiB (15 extents).

Logical volume vg01/lv01 successfully resized.

```
1 [root@exercise1 ~]# vgs
2   VG      #PV #LV #SN Attr   VSize   VFree
3   vg01     2   2   0 wz--n-   1.99g   1.86g
4   vg02     1   0   0 wz--n- 1008.00m 1008.00m
```

```
[root@exercise1 ~]# pvs
```

PV	VG	Fmt	Attr	PSize	PFree
/dev/sdb1	vg01	lvm2	a--	1020.00m	888.00m
/dev/sdb2	vg02	lvm2	a--	1008.00m	1008.00m
/dev/sdb3	vg01	lvm2	a--	1020.00m	1020.00m
/dev/sdb4		lvm2	---	1.00g	1.00g

```
[root@exercise1 ~]# cp -r /boot/grub /opt/test1/ #复制一些测试数据
```

```
[root@exercise1 ~]# vgreduce vg01 /dev/sdb1 #将sdb1移出失败,
因sdb1正在被使用
```

```
Physical volume "/dev/sdb1" still in use
```

```
[root@exercise1 ~]#
```

互动：如果sdb1是一个磁盘阵列，而这个磁盘阵列使用年代太久，我们必须移出怎么办？(数据迁移例子)

```
1 [root@exercise1 ~]# pvmove /dev/sdb1 /dev/sdb3 #将sdb1
  上数据移到新增加sdb3上
2 /dev/sdb1: Moved: 14.29%
3 /dev/sdb1: Moved: 42.86%
4 /dev/sdb1: Moved: 53.57%
5 /dev/sdb1: Moved: 67.86%
6 /dev/sdb1: Moved: 82.14%
7 /dev/sdb1: Moved: 100.00%
```

```
[root@exercise1 ~]# vgreduce vg01 /dev/sdb1
Removed "/dev/sdb1" from volume group "vg01"
```

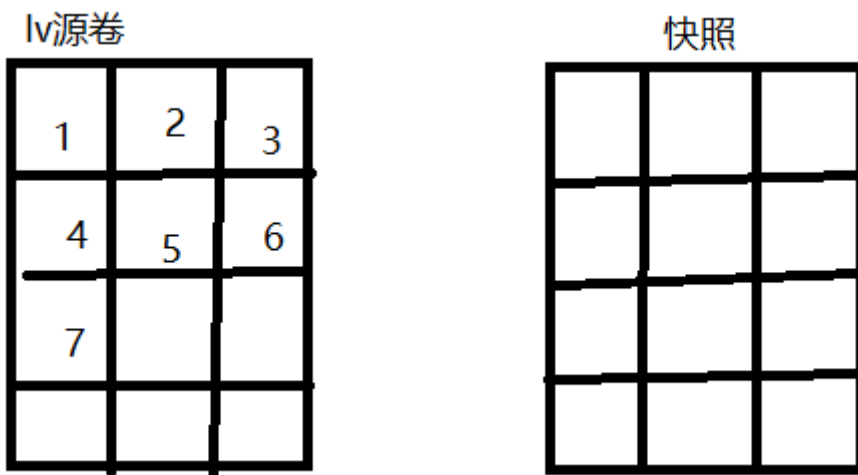
```
[root@exercise1 ~]# pvs
PV      VG  Fmt Attr PSize  PFree
/dev/sdb1    lvm2 ---  1.00g  1.00g
/dev/sdb2 vg02 lvm2 a-- 1008.00m 1008.00m
/dev/sdb3 vg01 lvm2 a-- 1020.00m 908.00m
/dev/sdb4    lvm2 ---  1.00g  1.00g
```

```
[root@exercise1 ~]#
```

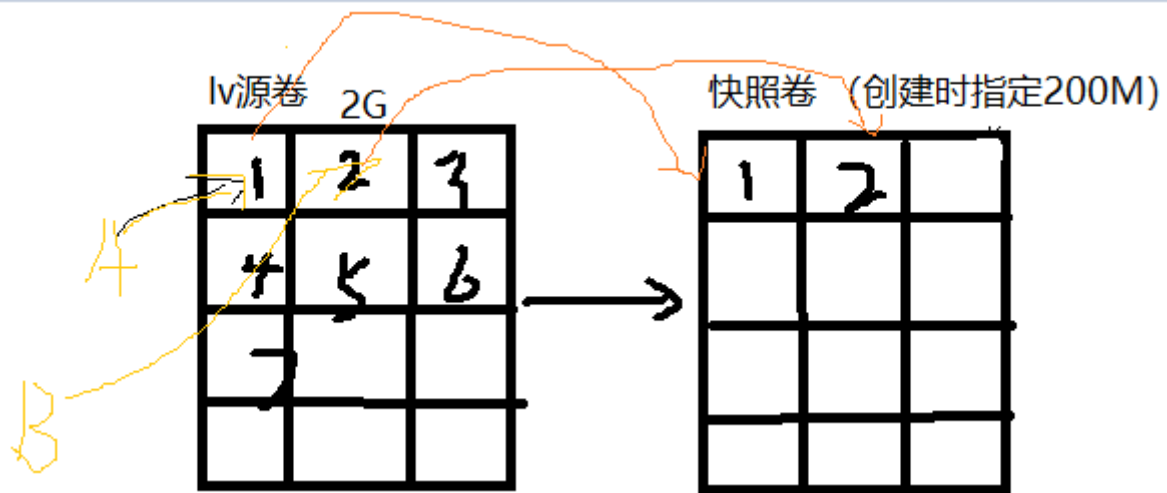
```
[root@exercise1 ~]# pvmove /dev/sdb1 /dev/sdb3
/dev/sdb1: Moved: 14.29%
/dev/sdb1: Moved: 42.86%
/dev/sdb1: Moved: 53.57%
/dev/sdb1: Moved: 67.86%
/dev/sdb1: Moved: 82.14%
/dev/sdb1: Moved: 100.00%
[root@exercise1 ~]# vgreduce vg01 /dev/sdb1
Removed "/dev/sdb1" from volume group "vg01"
[root@exercise1 ~]# pvs
PV          VG      Fmt  Attr PSize   PFree
/dev/sdb1           lvm2  ---   1.00g   1.00g
/dev/sdb2  vg02  lvm2  a--  1008.00m 1008.00m
/dev/sdb3  vg01  lvm2  a--  1020.00m  908.00m
/dev/sdb4           lvm2  ---   1.00g   1.00g
[root@exercise1 ~]#
```

14.lvm创建快照与恢复

快照在拍摄的一瞬间，系统会记录那个时间点逻辑卷的状态、数据等，此时拍下的快照相当于一张白纸。如图所示



快照做好后，随着时间的推移，源卷里的东西会发生改变。如数据1、2改写成了A、B，如图：



此时lv源卷里发生改变的数据会转移到快照卷里面去。当你恢复快照时，源卷会和快照进行合并，源卷里没有改变的数据+快照卷，就恢复到最初的状态。

准备测试文件

```
1 [root@exercise1 ~]# cd /opt/test1/
```

```
[root@exercise1 test1]# touch file{1..20}
```

```
[root@exercise1 test1]# ls
```

```
file1  file11  file13  file15  file17  file19  file20  file4  file6  file8  
file10  file12  file14  file16  file18  file2   file3   file5  file7  file9
```

```
[root@exercise1 test1]#
```

创建快照

```
1 [root@exercise1 test1]# lvcreate -L100M -s -n lv01_snap  
   /dev/vg01/lv01    #(源卷路径)  
2   Logical volume "lv01_snap" created.
```

```
[root@exercise1 test1]# lvdisplay
```

```
--- Logical volume ---
```

```
LV Path          /dev/vg01/lv01
```

```
LV Name          lv01
```

```
VG Name          vg01
```

```
LV UUID          6tjn24-B6j1-91SA-fLqr-xyld-HtO2-uLCIjl
```

LV Write Access read/write

LV Creation host, time exercise1, 2022-02-03 14:56:50 +0800

LV snapshot status source of
 lv01_snap [active]

LV Status available

open 1

LV Size 112.00 MiB

Current LE 28

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 8192

Block device 253:0

--- Logical volume ---

LV Path /dev/vg01/lv01_snap

LV Name lv01_snap

VG Name vg01

LV UUID siERVQ-KXBb-ucjj-aTiU-6fmV-0YME-eM0Mxd

LV Write Access read/write

LV Creation host, time exercise1, 2022-02-03 15:01:29 +0800

LV snapshot status active destination for lv01

LV Status available

open 0

LV Size 112.00 MiB

Current LE 28

COW-table size 100.00 MiB

COW-table LE 25

Allocated to snapshot 0.00%

Snapshot chunk size 4.00 KiB

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 8192

Block device 253:3


```
[root@exercise1 test1]#
```

```
[root@exercise1 test1]# lvdisplay
```

```
--- Logical volume ---
```

```
LV Path          /dev/vg01/lv01
```

```
LV Name          lv01
```

```
VG Name          vg01
```

```
LV UUID          6tjn24-B6j1-91SA-fLqr-xyld-HtO2-uLCIjI
```

```
LV Write Access   read/write
```

```
LV Creation host, time exercise1, 2022-02-03 14:56:50 +0800
```

```
LV snapshot status source of
```

```
lv01_snap [active]
```

```
LV Status         available
```

```
# open            1
```

```
LV Size           112.00 MiB
```

```
Current LE        28
```

```
Segments          1
```

```
Allocation         inherit
```

```
Read ahead sectors auto
```

```
- currently set to 8192
```

```
Block device       253:0
```

```
--- Logical volume ---
```

```
LV Path          /dev/vg01/lv01_snap
```

```
LV Name          lv01_snap
```

```
VG Name          vg01
```

```
LV UUID          siERVQ-KXBb-ucjj-aTiU-6fmV-0YME-eM0Mxd
```

```
LV Write Access   read/write
```

```
LV Creation host, time exercise1, 2022-02-03 15:01:29 +0800
```

```
LV snapshot status active destination for lv01
```

```
LV Status         available
```

```
# open            0
```

```
LV Size           112.00 MiB
```

```
Current LE        28
```

```
COW-table size    100.00 MiB #快照大小（备份文件的变动大小变过  
100M，该快照自动失效）
```

COW-table LE 25

Allocated to snapshot 0.00% #此时使用率为0

Snapshot chunk size 4.00 KiB

Segments 1

Allocation inherit

Read ahead sectors auto

- currently set to 8192

Block device 253:3

挂载快照到对比目录

```
1 [root@exercise1 test1]# mkdir /opt/test2
```

举例：挂载快照出错

```
[root@exercise1 test1]# mount /dev/vg01/lv01_snap /opt/test2
mount: 文件系统类型错误、选项错误、/dev/mapper/vg01-lv01_snap
上有坏超级块、
```

缺少代码页或助手程序，或其他错误

有些情况下在 syslog 中可以找到一些有用信息- 请尝试
dmesg | tail 这样的命令看看。

```
[root@exercise1 test1]# tail -100 /var/log/messages
```

.....

```
Feb 3 15:05:46 exercise1 kernel: XFS (dm-3): Filesystem has
duplicate UUID 0739dfe4-57eb-49cc-9cf9-d188a5c80dc4 - can't mount
Feb 3 15:06:01 exercise1 systemd: Created slice User Slice of lin05.
Feb 3 15:06:01 exercise1 systemd: Starting User Slice of lin05.
Feb 3 15:06:01 exercise1 systemd: Started Session 14 of user lin05.
Feb 3 15:06:01 exercise1 systemd: Starting Session 14 of user lin05.
Feb 3 15:06:01 exercise1 systemd: Removed slice User Slice of lin05.
Feb 3 15:06:01 exercise1 systemd: Stopping User Slice of lin05.
```

```
[root@exercise1 test1]# mount -o nouuid /dev/vg01/lv01_snap
/opt/test2
```

```
[root@exercise1 test1]# df -h
文件系统          容量 已用 可用 已用% 挂载点
/dev/sda3          18G  2.3G  16G   13% /
devtmpfs           479M   0  479M   0% /dev
tmpfs              489M   0  489M   0% /dev/shm
tmpfs              489M  6.8M  482M   2% /run
tmpfs              489M   0  489M   0% /sys/fs/cgroup
/dev/sr0            4.3G  4.3G   0 100% /mnt
/dev/sda1           197M  97M  100M   50% /boot
tmpfs              98M   0   98M   0% /run/user/0
/dev/mapper/vg01-lv01 109M  1.1M  108M   1% /opt/test1
/dev/mapper/vg01-lv01_snap 109M  5.9M  103M   6% /opt/test2
[root@exercise1 test1]#
```

为了防止快照失效，有两种办法，第一是创建完快照之后，挂载到某个目录，马上将里面的内容打包备份出来。另一个是，建立一个和原来的lv一样大小或者更大的快照，这样的快照也不会失效。

恢复快照

```
[root@exercise1 ~]# rm -rf /opt/test1/*
```

注：要恢复快照，需要快照与备分区都要取消挂载

```
1 [root@exercise1 ~]# umount /opt/test1
2 [root@exercise1 ~]# umount /opt/test2
```

```
[root@exercise1 ~]# lvconvert --merge /dev/vg01/lv01_snap
Merging of volume vg01/lv01_snap started.
vg01/lv01: Merged: 100.00%
```

#恢复成功，快照消失

```
[root@exercise1 ~]# lvdisplay
--- Logical volume ---
LV Path          /dev/vg01/lv01
LV Name           lv01
VG Name           vg01
LV UUID           6tjn24-B6j1-91SA-fLqr-xyld-HtO2-uLCjlj
LV Write Access   read/write
```

LV Creation host, time exercise1, 2022-02-03 14:56:50 +0800

LV Status available
open 0
LV Size 112.00 MiB
Current LE 28
Segments 1
Allocation inherit
Read ahead sectors auto
- currently set to 8192
Block device 253:0

#重新挂载lv01, 验证数据是否恢复

```
[root@exercise1 ~]# mount /dev/vg01/lv01 /opt/test1
```

```
[root@exercise1 ~]# ls /opt/test1/
```

```
file1  file11  file13  file15  file17  file19  file20  file4  file6  file8  
file10  file12  file14  file16  file18  file2   file3   file5  file7  file9
```

```
[root@exercise1 ~]#
```

15.LVM删除

创建LVM流程:

pvcreate创建pv->vgcreate创建卷组->lvcreate创建逻辑卷->mkfs.xfs

lv 格式化->mount挂载

删除LVM流程:

umount卸载->lvremove lv 移出卷组中所有逻辑卷->vgremove vg移出卷组->pvremove移出pv

```
1 [root@exercise1 ~]# umount /opt/test1
```

```
[root@exercise1 ~]# lvremove /dev/vg01/lv01
```

```
Do you really want to remove active logical volume vg01/lv01? [y/n]:
```

y

```
#如果卷组中还有lv, 移出时, 会提示, 是否也移出, 咱们这里直接移出  
Logical volume "lv01" successfully removed
```

```
[root@exercise1 ~]# lvs #已经看不到lv01
```

```
[root@exercise1 ~]# vgremove vg01
```

```
Volume group "vg01" successfully removed
```

```
[root@exercise1 ~]# vgs
```

```
VG #PV #LV #SN Attr VSize VFree
```

```
vg02 1 0 0 wz--n- 1008.00m 1008.00m #没有vg01
```

```
[root@exercise1 ~]#
```

移出pvs db1:

```
1 [root@exercise1 ~]# pvs
2 PV VG Fmt Attr PSize PFree
3 /dev/sdb1 lvm2 --- 1.00g 1.00g
4 /dev/sdb2 vg02 lvm2 a-- 1008.00m 1008.00m
5 /dev/sdb3 lvm2 --- 1.00g 1.00g
6 /dev/sdb4 lvm2 --- 1.00g 1.00g
```

```
[root@exercise1 ~]# pvremove /dev/sdb1 #已经移出
```

```
Labels on physical volume "/dev/sdb1" successfully wiped.
```

```
[root@exercise1 ~]# pvs
```

```
PV VG Fmt Attr PSize PFree
```

```
/dev/sdb2 vg02 lvm2 a-- 1008.00m 1008.00m
```

```
/dev/sdb3 lvm2 --- 1.00g 1.00g
```

```
/dev/sdb4 lvm2 --- 1.00g 1.00g
```

```
[root@exercise1 ~]#
```

16、实战-使用SSM工具为公司的邮件服务器创建可动态扩容的存储池

安装SSM **ssm工具了一下**

```
1 [root@exercise1 ~]# yum -y install system-storage-manager
```

SSM：检查关于可用硬驱和LVM卷的信息。显示关于现有磁盘存储设备、存储池、LVM卷和存储快照的信息。

查看磁盘信息

列出设备信息

```
1 [root@exercise1 ~]# ssm list dev
2 -----
3 Device              Free      Used      Total  Pool  Mount
4 point              -----
5 /dev/sda              20.00 GB
6 /dev/sda1             200.00 MB      /boot
7 /dev/sda2              2.00 GB      SWAP
8 /dev/sda3             17.80 GB      /
9 /dev/sdb              20.00 GB
10 /dev/sdb1             1.00 GB
11 /dev/sdb2  1008.00 MB  0.00 KB    1.00 GB  vg02
```

```

12 /dev/sdb3                                1.00 GB
13 /dev/sdb4                                1.00 GB
14 -----
15 [root@exercise1 ~]#

```

存储池信息

```

1 [root@exercise1 ~]# ssm list pool
2 -----
3 Pool  Type  Devices      Free      Used      Total
4 -----
5 vg02  lvm    1            1008.00 MB  0.00 KB   1008.00 MB
6 -----
7 [root@exercise1 ~]#

```

实战场景：公司要搭建一台邮件服务器，考虑到后期公司发展规模扩张，需要你创建一个名为mail的LVM存储池，并在其上创建一个名为mail-lv，初始大小为1G的lvm卷，格式化为xfs文件系统，并将其挂载/mail-lv目录下。此存储池中的空间后期要可以动态扩容。

将sdb上所有卷组信息删除

```

1 [root@exercise1 ~]# vgremove vg02
2    volume group "vg02" successfully removed

```

```

[root@exercise1 ~]# pvremove /dev/sdb{1,2,3,4}
No PV found on device /dev/sdb1.
Labels on physical volume "/dev/sdb2" successfully wiped.
Labels on physical volume "/dev/sdb3" successfully wiped.
Labels on physical volume "/dev/sdb4" successfully wiped.

```

#创建目录

```

[root@exercise1 ~]# mkdir /opt/mail-lv

```

ssm create -s lv大小 -n lv名称 --fstype lv文件系统类型 -p 卷组名 设备 挂载点

自动把设备变成pv, 创建vg,lv,格式化文件系统,自动挂载

```
1 [root@exercise1 ~]# ssm create -s 1G -n mail-lv --fstype xfs -p mail /dev/sdb[1-4] /opt/mail-lv/
```

```
[root@exercise1 ~]# ssm create -s 1G -n mail-lv --fstype xfs -p mail /dev/sdb[1-4] /opt/mail-lv/
Physical volume "/dev/sdb1" successfully created.
Physical volume "/dev/sdb2" successfully created.
Physical volume "/dev/sdb3" successfully created.
Physical volume "/dev/sdb4" successfully created.
Volume group "mail" successfully created
WARNING: xfs signature detected on /dev/mail/mail-lv at offset 0. Wipe it? [y/n]: y
Wiping xfs signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 16384. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 17408. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 18432. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 19456. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 21504. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 30208. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 31232. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 32256. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 33280. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 34304. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 35328. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
WARNING: xfs_external_log signature detected on /dev/mail/mail-lv at offset 36352. Wipe it? [y/n]: y
Wiping xfs_external_log signature on /dev/mail/mail-lv.
Logical volume "mail-lv" created.
meta-data=/dev/mail/mail-lv isize=512    agcount=4, agsize=65536 blks
          =                  sectsz=512   attr=2, projid32bit=1
          =                  crc=1        finobt=0, sparse=0
data      =                  bsize=4096   blocks=262144, imaxpct=25
          =                  sunit=0      swidth=0 blks
naming    =version 2          bsize=4096   ascii-ci=0 ftype=1
log        =internal log     bsize=4096   blocks=2560, version=2
          =                  sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
```

```
1 [root@exercise1 ~]# df -h /opt/mail-lv/
2 文件系统                      容量  已用   可用 已用% 挂载点
3 /dev/mapper/mail-mail--lv 1014M    33M   982M    4%
  /opt/mail-lv
4 [root@exercise1 ~]#
```

```
y[root@exercise1 ~]# df -h /opt/mail-lv/
文件系统                      容量  已用   可用 已用% 挂载点
/dev/mapper/mail-mail--lv 1014M    33M   982M    4% /opt/mail-lv
[root@exercise1 ~]#
```

注意

减少/缩小逻辑卷是数据损坏的最高风险。

所以，如果可能的话，尽量避免这种情况，但如果没有其他选择的话，那就继续。

缩减 LVM 之前，建议先做一个备份。

当你在 LVM 中的磁盘空间耗尽时，你可以通过缩小现有的没有使用全部空间的 LVM，而不是增加一个新的物理磁盘，在卷组上腾出一些空闲空间。

需要注意的是：在 GFS2 或者 XFS 文件系统上不支持缩小

减少逻辑卷涉及以下步骤：

1. 卸载文件系统
2. 检查文件系统是否有任何错误(e2fsck 命令)
3. 缩小文件系统的大小(resize2fs命令)**resize2fs [现有逻辑卷名] [新的文件系统大小]**
4. 缩小逻辑卷的大小
5. 重新检查文件系统是否存在错误（可选）
6. 挂载文件系统
7. 检查减少后的文件系统大小

e2fsck命令

实例

```
1 #检查 /dev/hda5 是否正常，如果有异常便自动修复，并且设定若有问
   答，均回答[是]
2 e2fsck -a -y /dev/hda5
3
4 #注意：
5
6 大部份使用 e2fsck 来检查硬盘 partition 的情况时，通常都是情形特
   殊，因此最好先将该 partition umount，然后再执行 e2fsck 来做检
   查，若是要非要检查 / 时，则请进入 singal user mode 再执行
```

练习：

题目1：

在虚拟机上添加1个硬盘，大小为2G；

在新增的硬盘上新建1个1G的分区，格式化该分区为xfs文件系统；

创建目录/data，将该分区挂载到/data目录，要求开机自动挂载；

将在新增的硬盘上的剩余空间分区为交换空间（swap），并启用它，且要求开机自动生效。（额外题目）

在虚拟机上添加2个硬盘，大小分别为1G；

将以上的2个1G的硬盘组建成为卷组vg01，并创建大小为1500M的逻辑卷lv01；

格式化为ext4文件系统并挂载到/lv01使用；

为上述逻辑卷创建大小为200M的快照。

题目2：

配置存储服务器，要求使用RAID+LVM技术结合在一起，创建/dev/md10，从中取20G存储资源出来创建存储卷/dev/webData/lv_web（xfs文件系统）并挂载到/webData，用/webData做自定义yum源仓库，建快照。注意：存储服务器至少要有4块可用硬盘！

题目3：

假如凌晨3点，根“/”满了，如何处理和预防，请说出你的方案（自由发挥）

分两种情况：

根“/”如果是lvm的，直接扩容

扩展RHEL7根分区

查看当前根分区容量

```
1 [root@localhost ~]# df -h
2 Filesystem              Size  Used Avail Use% Mounted on
3 /dev/mapper/rhel-root    17G   1.2G   16G    7% /
4 devtmpfs                 902M    0   902M    0% /dev
5 tmpfs                    912M    0   912M    0% /dev/shm
6 tmpfs                    912M   8.7M   904M    1% /run
7 tmpfs                    912M    0   912M    0%
8 /sys/fs/cgroup
9 /dev/sr0                  3.6G   3.6G    0 100% /yum
10 /dev/sda1                1014M  139M   876M   14% /boot
11 tmpfs                    183M    0   183M    0%
12 /run/user/0
```

把存储设备挂载到 RHEL7 系统并把它做成物理卷

```
1 [root@localhost ~]# fdisk -l
2
3 Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040
  sectors
4 Units = sectors of 1 * 512 = 512 bytes
5 Sector size (logical/physical): 512 bytes / 512 bytes
6 I/O size (minimum/optimal): 512 bytes / 512 bytes
7 Disk label type: dos
8 Disk identifier: 0x000c39f9
9
10    Device Boot      Start         End      Blocks   Id
11    /dev/sda1  *          2048       2099199       1048576    83
12    /dev/sda2             2099200       41943039       19921920    8e
13    Linux LVM
14 Disk /dev/mapper/rhel-root: 18.2 GB, 18249416704 bytes,
  35643392 sectors
```

```

15 Units = sectors of 1 * 512 = 512 bytes
16 Sector size (logical/physical): 512 bytes / 512 bytes
17 I/O size (minimum/optimal): 512 bytes / 512 bytes
18
19
20 Disk /dev/mapper/rhel-swap: 2147 MB, 2147483648 bytes,
  4194304 sectors
21 Units = sectors of 1 * 512 = 512 bytes
22 Sector size (logical/physical): 512 bytes / 512 bytes
23 I/O size (minimum/optimal): 512 bytes / 512 bytes
24
25
26 Disk /dev/sdb: 15.6 GB, 15610576896 bytes, 30489408
  sectors
27 Units = sectors of 1 * 512 = 512 bytes
28 Sector size (logical/physical): 512 bytes / 512 bytes
29 I/O size (minimum/optimal): 512 bytes / 512 bytes
30 [root@localhost ~]# ls /dev/sd*
31 /dev/sda /dev/sda1 /dev/sda2 /dev/sdb
32 [root@localhost ~]# pvcreate /dev/sdb
33   Physical volume "/dev/sdb" successfully created.
34

```

扩展卷组

```

1 [root@localhost ~]# pvdisplay
2   --- Physical volume ---
3   PV Name                /dev/sda2
4   VG Name                rhel
5   PV Size                19.00 GiB / not usable 3.00 MiB
6   Allocatable            yes (but full)
7   PE Size                4.00 MiB
8   Total PE               4863
9   Free PE                0
10  Allocated PE            4863
11  PV UUID                 I9U34V-LKAQ-n4EZ-y3ab-up7c-
    Ph89-PhBI9C
12

```

```

13  "/dev/sdb" is a new physical volume of "14.54 GiB"
14  --- NEW Physical volume ---
15  PV Name                /dev/sdb
16  VG Name
17  PV Size                14.54 GiB
18  Allocatable           NO
19  PE Size                0
20  Total PE              0
21  Free PE               0
22  Allocated PE          0
23  PV UUID                8nWGYZ-Btjh-DRBd-3JuZ-wXRr-
vXTa-BywgBI
24  [root@localhost ~]# vgdisplay
25  --- volume group ---
26  VG Name                rhel
27  System ID
28  Format                 lvm2
29  Metadata Areas         2
30  Metadata Sequence No   4
31  VG Access              read/write
32  VG Status              resizable
33  MAX LV                 0
34  Cur LV                 2
35  Open LV                2
36  Max PV                 0
37  Cur PV                 2
38  Act PV                 2
39  VG Size                33.53 GiB
40  PE Size                4.00 MiB
41  Total PE              8584
42  Alloc PE / Size        4863 / 19.00 GiB
43  Free PE / Size         3721 / 14.54 GiB
44  VG UUID                ds3VQ9-f3Tx-MJhs-a36Q-gSHD-
gGiR-Cv7ika
45  [root@localhost ~]# vgextend rhel /dev/sdb
46  volume group "rhel" successfully extended
47

```


扩展逻辑卷

```

1 #扩展根分区存储量
2 [root@localhost ~]# lvextend -l +100%FREE
/dev/mapper/rhel-root #FREE也可以小写
3   Size of logical volume rhel/root changed from 17.00
GiB (4351 extents) to 31.53 GiB (8072 extents).
4   Logical volume rhel/root successfully resized.
5 #查看当前根分区容量
6 [root@localhost ~]# df -h
7 Filesystem                Size      Used Avail Use% Mounted on
8 /dev/mapper/rhel-root    17G       1.2G    16G     7% /
9 devtmpfs                  902M         0   902M     0% /dev
10 tmpfs                     912M         0   912M     0% /dev/shm
11 tmpfs                     912M      8.7M   904M     1% /run
12 tmpfs                     912M         0   912M     0%
/sys/fs/cgroup
13 /dev/sr0                   3.6G      3.6G        0 100% /yum
14 /dev/sda1                 1014M     139M    876M    14% /boot
15 tmpfs                     183M         0   183M     0%
/run/user/0
16 #查看根分区文件系统类型
17 [root@localhost ~]# blkid /dev/mapper/rhel-root
18 /dev/mapper/rhel-root: UUID="8d40a78a-0db6-4dfa-8ba6-
0cd2700ed8db" TYPE="xfs"
19 #扩展文件系统
20 [root@localhost ~]# xfs_growfs /dev/mapper/rhel-root
21 meta-data=/dev/mapper/rhel-root  isize=512
   agcount=4, agsize=1113856 blks
22           =                       sectsz=512   attr=2,
   projid32bit=1
23           =                       crc=1        finobt=0
   spinodes=0
24 data      =                       bsize=4096
   blocks=4455424, imaxpct=25
25           =                       sunit=0      swidth=0
   blks

```