

## Linux 云计算集群架构师

**学神 IT 教育：从零基础到实战，从入门到精通！**

### 版权声明：

本系列文档为《学神 IT 教育》内部使用教材和教案，只允许 VIP 学员个人使用，禁止私自传播。否则将取消其 VIP 资格，追究其法律责任，请知晓！

### 免责声明：

本课程设计目的只用于教学，切勿使用课程中的技术进行违法活动，学员利用课程中的技术进行违法活动，造成的后果与讲师本人及讲师所属机构无关。倡导维护网络安全人人有责，共同维护网络文明和谐。

### 联系方式：

学神 IT 教育官方网站: <http://www.xuegod.cn>

Linux 云计算架构师进阶学习群 QQ 群: 1072932914



学习顾问：小语老师

学习顾问：边边老师

学神微信公众号

微信扫码添加学习顾问微信，同时扫码关注学神公众号了解最新动态，获取更多学习资料及答疑就业服务！

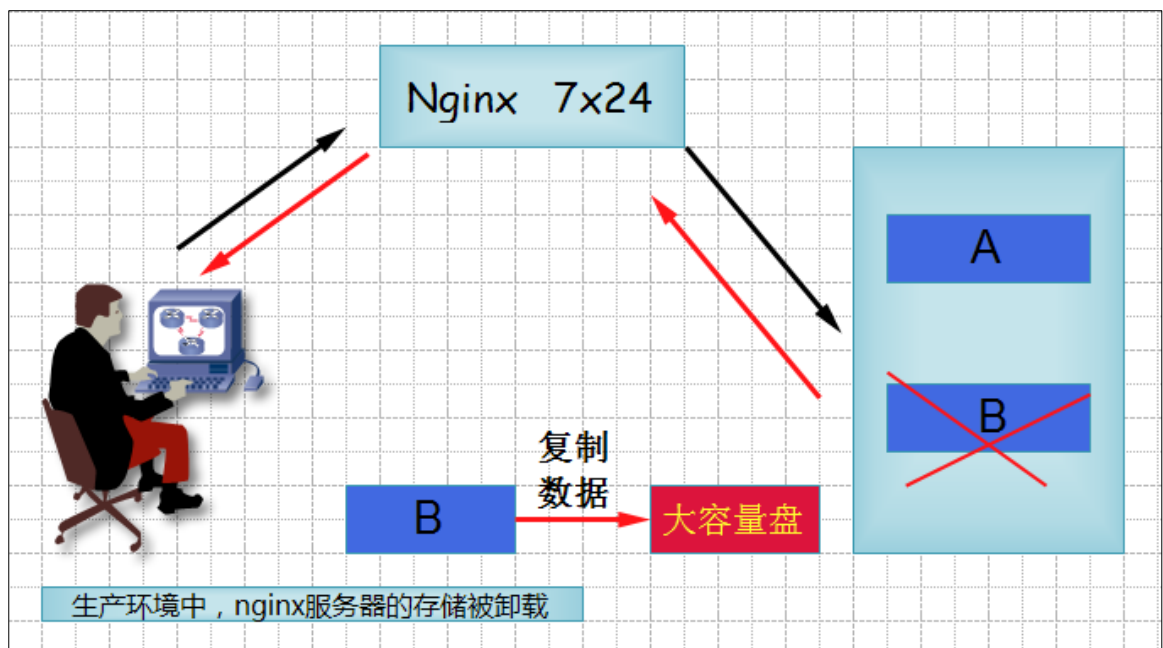
## LVM 管理和 ssm 存储管理器使用

### 本节所讲内容:

- 15.1 LVM 的工作原理
- 15.2 创建 LVM 的基本步骤
- 15.3 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

#### 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

### 15.1 LVM 的工作原理

LVM ( Logical Volume Manager) 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。

#### 15.1.1 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** (physical volume): 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** (Volume Group): 一个 LVM 卷组由一个或多个物理卷组成

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

PE (physical extents): **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 (logical extent): **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)

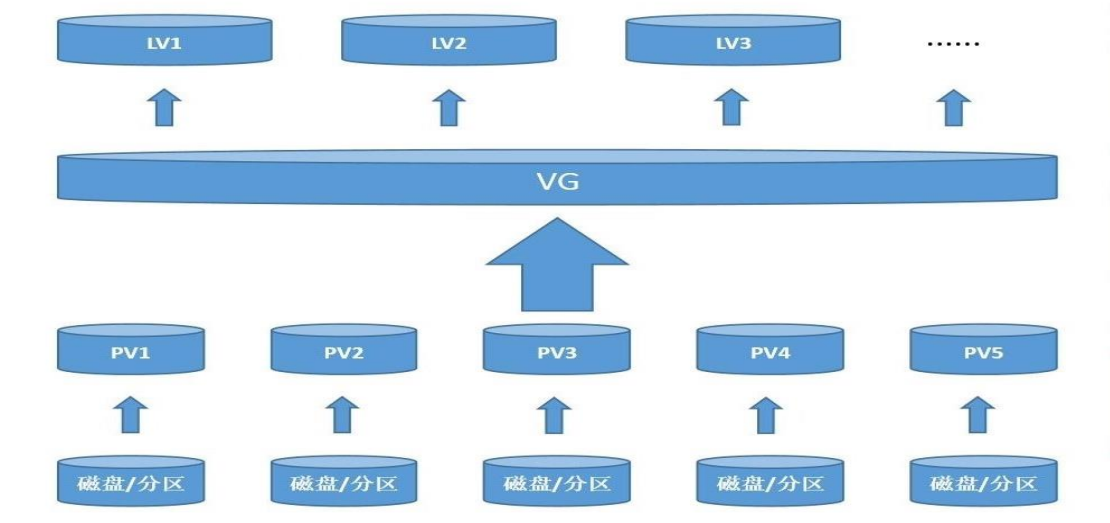
文件系统 block (1K 或 4K ) #mkfs.ext4 -b 2048 /dev/sdb1 , 最大支持到 4096

raid chunk (512K) #mdadm -C -v /dev/md5 -l 5 -n 3 -c 512 -x 1

/dev/sde{1,2,3,5}

LVM PE (4M) #vgcreate -s 4M vg1 /dev/sdb{1,2}

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

### 15.1.2 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。

## 15.2 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

### 15.2.1 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	pvdisk	vgdisplay	lvdisplay
remove 移除	pvremove	vgremove	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	pvdisk
卷组	vgs	vgscan	vgdisplay
逻辑卷	lvs	lvscan	lvdisplay

### 15.2.2 创建并使用 LVM 逻辑卷

1、创建 PV

添加一个 sdb 磁盘

[root@xuegod63 ~]# fdisk /dev/sdb #创建 4 个主分区, 每个分区 1G

[root@xuegod63 ~]# ls /dev/sdb\*

```
/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`

8e 代表是 Linux LVM 卷这样一个分区 (centos7 的序号是 31)

注: 现在系统已经很智能了, 直接使用默认的 83 Linux 分区 (centos7 是 20), 也可以创建 pv 的。

```
[root@xuegod63 ~]# yum -y install lvm2
```

```
[root@xuegod63 ~]# pvcreate /dev/sdb{1,2,3,4} -y #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@xuegod63 ~]# 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          SHKFwf-WsLr-kkox-wlee-dAXc-5eL0-hyhaTV
```

创建 vg 卷组:

语法: `vgcreate vg 名字 pv 的名字`

```
[root@xuegod63 ~]# vgcreate vg01 /dev/sdb1
```

```
Volume group "vg01" successfully created
```

```
[root@xuegod63 ~]# vgs
```

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

```
vg01 1 0 0 wz--n- 1020.00m 1020.00m
```

```
[root@xuegod63 ~]# vgdisplay vg01
```

```
--- 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
```

#### 创建 LV

lvcreate -n 指定新逻辑卷的名称 -L 指定 lv 大小的 SIZE(M,G) (-l: 小 l 指定 LE 的数量)

vgname 卷组名

```
[root@xuegod63 ~]# lvcreate -n lv01 -L 16M vg01 -y #大写 L 指定逻辑卷大小 (常用)
```

Logical volume "lv01" created.

```
[root@xuegod63 ~]# lvcreate -n lv02 -l 4 vg01 #小写 l 指定 LE 的数量 (了解)
```

Logical volume "lv02" created.

```
[root@xuegod63 ~]# lvs
```

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@xuegod63 ~]# pvdisplay /dev/sdb1
```

--- 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 (同一卷组中, LE 大小等于 PE 的大小)

Total PE 255

Free PE 247

Allocated PE 8 # Allocated ['æləkeɪtɪd] 分配, 已经使用了 8 个 PE

```
[root@xuegod63 ~]# vgdisplay vg01
```

...

Alloc PE / Size 8 / 32.00 MiB #已经使用 8 个 PE/ 32M 大小 8\*4=32

Free PE / Size 247 / 988.00 MiB # 空闲的 PE 数量 / 空闲大小 247\*4=988

df -

互动: lv01 逻辑卷的路径在哪?

```
[root@xuegod63 ~]# ls /dev/vg01/ #查看逻辑卷
```

lv01 lv02

```
[root@xuegod63 ~]# ll /dev/vg01/lv01 #其实 lv01 是 dm-0 的软链接
```

```
[root@xuegod63 ~]# mkfs.ext4 /dev/vg01/lv01
```

```
lrwxrwxrwx 1 root root 7 5月 18 19:02 /dev/vg01/lv01 -> ../dm-0
```

```
[root@xuegod63 ~]# mount /dev/vg01/lv01 /lv01
```

```
[root@xuegod63 ~]# df -Th /lv01
文件系统          类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vg01-lv01 ext4   15M 268K  14M   2% /lv01
[root@xuegod63 ~]# echo  "/dev/vg01/lv01 /lv01 ext4 defaults 0 0" >> /etc/fstab
```

### 15.2.3 指定 PE 大小

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

```
[root@xuegod63 ~]# vgcreate -s 16M vg02 /dev/sdb2
```

Volume group "vg02" successfully created

PE 的大小只有为 2 的幂数, 且最大为 512M

```
[root@xuegod63 ~]# 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    #已经是 16MB
```

### 15.2.4 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

```
[root@xuegod63 ~]# vgs
VG   #PV #LV #SN Attr   VSize   VFree
vg01   1   2   0 wz--n- 1020.00m 988.00m
vg02   1   0   0 wz--n- 1008.00m 1008.00m
```

用的命令如下:

extend 扩展    `vgextend`    `lvextend`

扩容逻辑卷

```
[root@xuegod63 ~]# lvextend -L +30m /dev/vg01/lv01
```

说明: 在指定大小的时候, 扩容 30m 和扩容到 30m 是不一样的写法

扩容 30m =====> `-L +30M`

扩容到 30m =====> `-L 30M`    使用扩容到, 那么新大小不能低于现有大小

```
[root@xuegod63 ~]# lvextend -L +30m /dev/vg01/lv01
```

Rounding size to boundary between physical extents: 32.00 MiB.



Size of logical volume vg01/lv01 changed from 16.00 MiB (4 extents) to 48.00 MiB (12 extents).

Logical volume vg01/lv01 successfully resized.

```
[root@xuegod63 ~]# lvs
LV VG Attr LSize Pool Origin Data% Meta% Move Log Cpy%Sync
Convert
lv01 vg01 -wi-ao---- 48.00m #LV 已经扩容成功
lv02 vg01 -wi-a----- 16.00m
```

```
[root@xuegod63 ~]# df -Th /lv01
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/vg01-lv01	ext4	15M	268K	14M	2%	/lv01

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** **resize2fs /dev/vg01/lv01**

xfs 文件系统扩容使用命令语法: **xfs\_growfs 挂载点** **xfs\_growfs /lv01**

**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**

```
[root@xuegod63 ~]# pvcreate /dev/sdc
[root@xuegod63 ~]# vgcreate vg05 /dev/sdc
[root@xuegod63 ~]# lvcreate -n lv05 -L 1G vg05
[root@xuegod63 ~]# mkfs.xfs /dev/vg05/lv05
[root@xuegod63 ~]# blkid /dev/vg05/lv05
[root@xuegod63 ~]# mkdir /lv05
[root@xuegod63 ~]# mount /dev/vg05/lv05 /lv05
[root@xuegod63 ~]# lvextend -L +100M -r /dev/vg05/lv05
[root@xuegod63 ~]# df -hT && lvs
[root@xuegod63 ~]# lvextend -L +100M /dev/vg01/lv01
[root@xuegod63 ~]# df -hT && lvs
[root@xuegod63 ~]# xfs_growfs /lv01
[root@xuegod63 ~]# df -hT && lvs
```

```
[root@xuegod63 ~]# resize2fs /dev/vg01/lv01
```

resize2fs 1.42.9 (28-Dec-2013)

Filesystem at /dev/vg01/lv01 is mounted on /lv01; on-line resizing required

old\_desc\_blocks = 1, new\_desc\_blocks = 1

The filesystem on /dev/vg01/lv01 is now 49152 blocks long.

```
[root@xuegod63 ~]# df -Th /lv01
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/vg01-lv01	ext4	46M	(扩容成功)	522K	43M	2% /lv01

```
[root@xuegod63 ~]# lvextend -L 80m -r /dev/vg01/lv01
```

**#直接扩容到 80M 空间, 增加-r 参数, 不用再扩文件系统了**

```
[root@xuegod63 ~]# df -T /lv01/
```

文件系统	类型	1K-块	已用	可用	已用%	挂载点
------	----	------	----	----	-----	-----



```
/dev/mapper/vg01-lv01 ext4 78303 776 73761 2% /lv01
[root@xuegod63 ~]# df -Th /lv01/
文件系统 类型 容量 已用 可用 已用% 挂载点
/dev/mapper/vg01-lv01 ext4 77M 776K 73M 2% /lv01
```

#### 15.2.5 VG 扩容

```
[root@xuegod63 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
vg01 1 2 0 wz--n- 1020.00m 924.00m
vg02 1 0 0 wz--n- 1008.00m 1008.00m
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
[root@xuegod63 ~]# pvcreate /dev/sdb3 # 创建 pv
[root@xuegod63 ~]# vgextend vg01 /dev/sdb3 #扩容成功
Volume group "vg01" successfully extended
[root@xuegod63 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
vg01 2 2 0 wz--n- 1.99g <1.90g
vg02 1 0 0 wz--n- 1008.00m 1008.00m
```

#### 15.2.6 LVM 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, 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。

实际生产中, 不推荐使用缩减, 扩的时候给小点, 不够再继续扩, 不推荐使用缩减, 可能会丢失数据

```
[root@xuegod63 ~]# lvreduce -L -20m -r /dev/vg01/lv01
WARNING: Reducing active and open 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. #缩小成功
[root@xuegod63 ~]# lvs #缩小成功
[root@xuegod63 ~]# df -hT /lv01/
[root@xuegod63 ~]# resize2fs /dev/vg01/lv01 #这个命令相当于扩容时的-r 选项
[root@xuegod63 ~]# lvreduce -L -100M -r /dev/vg05/lv05
```

```
fsadm: Xfs filesystem shrinking is unsupported.
/usr/sbin/fsadm failed: 1
Filesystem resize failed.
```

xfs 文件系统 缩小是不支持的

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

```
[root@xuegod63 ~]# vgs
VG #PV #LV #SN Attr VSize VFree
vg01 2 2 0 wz--n- 1.99g <1.92g
```

```
vg02 1 0 0 wz--n- 1008.00m 1008.00m
[root@xuegod63 ~]# pvs
PV          VG   Fmt  Attr PSize   PFree
/dev/sdb1   vg01 lvm2 a-- 1020.00m 944.00m
/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@xuegod63 ~]# cp -r /boot/grub /lv01/      #复制一些测试数据
[root@xuegod63 ~]# vgreduce vg01 /dev/sdb1      #将 sdb1 移出失败, 因 sdb1 正在被使用
```

Physical volume "/dev/sdb1" still in use

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?

移动数据:

```
[root@xuegod63 ~]# pvs
[root@xuegod63 ~]# pvmove /dev/sdb1 /dev/sdb3 #必须保证 sdb3 物理卷和 sdb1 物理卷属于一个卷组
```

**#将物理卷 sdb1 上的数据移到新增加物理卷 sdb3 上**

/dev/sdb1: Moved: 23.53%

/dev/sdb1: Moved: 76.47%

/dev/sdb1: Moved: 100.00%

```
[root@xuegod63 ~]# vgreduce vg01 /dev/sdb1
```

**#移完数据再把 sdb1 物理卷从 vg01 卷组中移除**

Removed "/dev/sdb1" from volume group "vg01"

```
[root@xuegod63 ~]# 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	952.00m

**#vg01 中只有 sdb3 了**

### 15.2.7 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

```
[root@xuegod63 ~]# umount /lv01
```

```
[root@xuegod63 ~]# lvremove /dev/vg01/lv01
```

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

Logical volume "lv01" successfully removed

```
[root@xuegod63 ~]# lvs
```

LV	VG	Attr	LSize	Pool	Origin	Data%	Meta%	Move	Log	Cpy%	Sync
lv02	vg01	-wi-a-----	16.00m								

Convert

**#已经看不到 lv01**

```
[root@xuegod63 ~]# vgremove vg01    #直接移出卷组
Do you really want to remove volume group "vg01" containing 1 logical volumes?
[y/n]: y
Do you really want to remove active logical volume vg01/lv02? [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 "lv02" successfully removed
Volume group "vg01" successfully removed
[root@xuegod63 ~]# vgs
VG    #PV #LV #SN Attr   VSize   VFree
vg02   1   0   0 wz--n- 1008.00m 1008.00m    #没有 vg01

移出 pv sdb1
[root@xuegod63 ~]# 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    lvm2 ---   1.00g   1.00g
/dev/sdb4    lvm2 ---   1.00g   1.00g
[root@xuegod63 ~]# pvremove /dev/sdb1    #已经移出
Labels on physical volume "/dev/sdb1" successfully wiped.
[root@xuegod63 ~]# 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
```

## 15.3 实战-使用 SSM 工具创建可动态扩容的存储池

安装 SSM:

```
[root@xuegod63 ~]# 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。

### 15.3.1 查看磁盘信息

列出设备信息

```
root@xuegod63 ~]# ssm list dev
```

```
-----
Device          Free    Used    Total Pool Mount point
-----
/dev/fd0                4.00 KB
/dev/sda                20.00 GB    PARTITIONED
/dev/sda1              200.00 MB    /boot
/dev/sda2                1.00 GB    SWAP
/dev/sda3              10.00 GB    /
```

```
/dev/sdb          20.00 GB
/dev/sdb1         1.00 GB
/dev/sdb2 1008.00 MB 0.00 KB   1.00 GB  vg02
/dev/sdb3         1.00 GB
/dev/sdb4         1.00 GB
```

#### 存储池信息

```
[root@xuegod63 ~]# ssm list pool
```

```
-----
Pool Type Devices      Free    Used    Total
-----
vg02  lvm    1        1008.00 MB 0.00 KB 1008.00 MB
-----
```

### 15.3.2 实战: 为公司的邮件服务器创建基于 LVM 的邮件存储

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:

```
[root@xuegod63 ~]# vgremove vg02
```

```
[root@xuegod63 ~]# pvremove /dev/sdb{1,2,3,4}
```

#### 创建目录

```
[root@xuegod72 ~]# mkdir /mail-lv
```

用的命令如下:

```
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
```

不需要手动创建物理卷, 卷组, 逻辑卷, 格式化文件系统, 挂载等操作

这条命令自动把设备变成 pv, 创建 vg, lv, 格式化文件系统, 自动挂载

```
[root@xuegod63 ~]# ssm create -s 1G -n mail-lv --fstype xfs -p mail /dev/sdb[1-4]
```

/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: ext4 signature detected on /dev/mail/mail-lv at offset 1080. Wipe it? [y/n]:

y

Wiping ext4 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
```

```
[root@xuegod63 ~]# df -h /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% /mail-lv
```

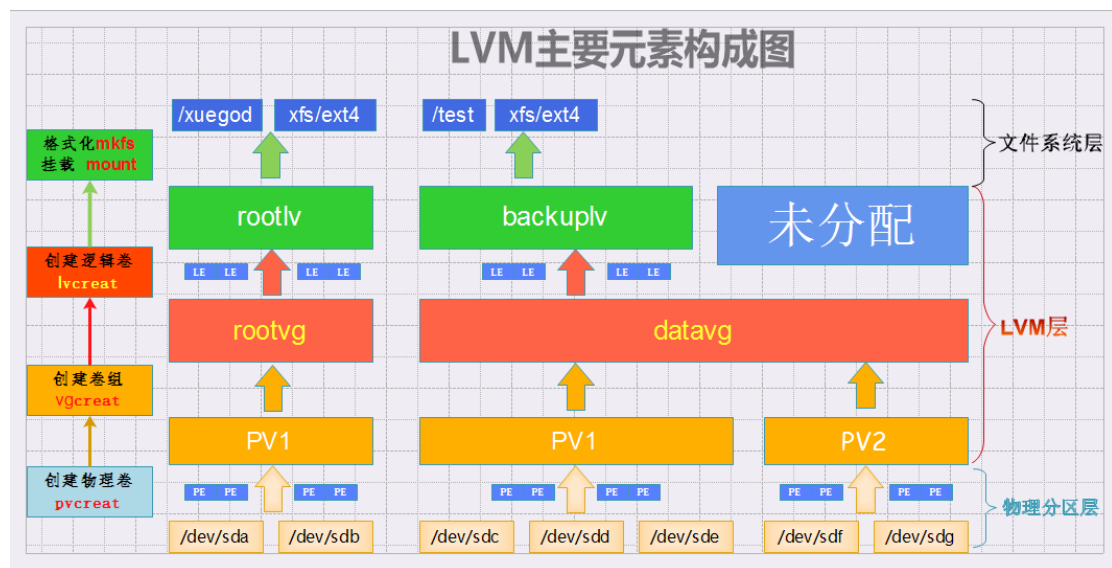
```
[root@xuegod63 ~]# pvs
```

```
[root@xuegod63 ~]# lvs
```

```
[root@xuegod63 ~]# lvextend -L +500m -r /dev/mail/mail-lv
```

```
[root@xuegod64 ~]# pvs
```

```
[root@xuegod64 ~]# lvs
```



## 总结:

### 15.1 LVM 的工作原理

### 15.2 创建 LVM 的基本步骤

### 15.3 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