我们在生产中会经常遇到磁盘空间不够用的情况，无论你前期规划的多精细，但最终还是会遇到这样的问题（所以磁盘的分区就没有一个标准的格式，只能是根据自己的服务器应用来适当的进行划分）。那么遇到这样的问题怎么办，如何来解决，这就是我们今天的主题:逻辑卷管理(LVM)。

# 1介绍

LVM 的全名是Logical Volume Manager。通过他可以随意扩大或缩小磁盘或分区的容量。LVM的实现是将磁盘、分区或RAID设备通过软件组合成一块独立的VG，然后将这个VG再进行划分出LV，这时候的LV就相当于一个分区了，你需要先格式化，然后挂载到系统中使用。LV能自由伸缩关键是底层的PE在起作用。下面就让我们来了解一下这些名词的相关内容。

LVM相关基础

Physical Extend：PE

PE就像是我们前面所说的磁盘的BLOCK，而这个的大小同样会影响到VG的大小。

Physical Volume：PV

我们还记得在分区的时候，把分区转换类型里面有个LVM的标识（8e）不，我们要做LVM，就必需先把这里的分区类型转换成8e。然后再用pvcreate将分区转换成PV，这一步是下一步的前提。

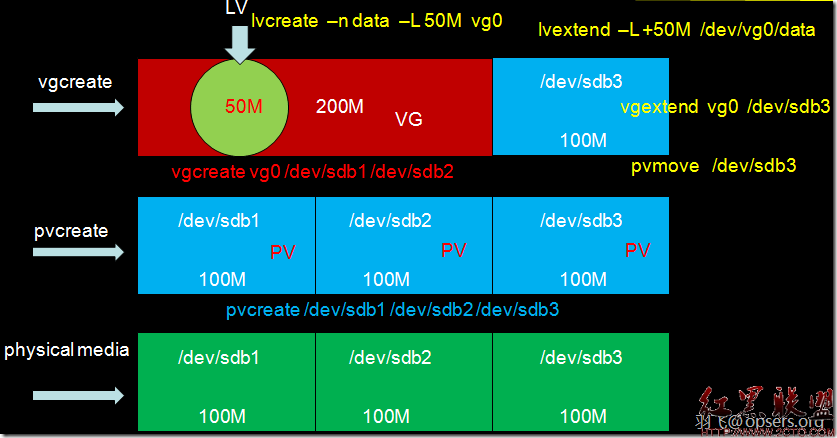
Volume Group：VG

所谓的VG，就是把多个PV组合成一个更大的磁盘，这就是VG。

Logical Volume：LV

我们要想使用VG，就必需把VG分成LV，这个LV你可以看作是分区了，当然分区后需要格式化才能挂载使用。

为了方便大家理解上面的内容，我做了一张图，希望对大家的理解带来点帮助。



## 2、LVM的创建

本次实验的环境是RHEL6系统，一个分区，一块磁盘，一个RAID0设备。所以你的环境至少有三个磁盘，一个分区，为了后面的实验，你还需要有更多的空间，为了扩大LVM。

关于环境的搭建，我相信通过前面的学习，因为不会有问题了！如果你不太了解RAID创建的话，参考前面的课程：http://www.opsers.org/base/learning-linux-the-day-that-the-system-configuration-in-the-rhel6-disk-array-raid.html

我这里就快速的搭建一下。

### 创建个分区

[root@yufei ~]# fdisk /dev/sdb

Command (m for help): n

Command action

   e   extended

   p   primary partition (1-4)

p

Partition number (1-4): 1

First cylinder (1-261, default 1):

Using default value 1

Last cylinder, +cylinders or +size{K,M,G} (1-261, default 261):

Using default value 261

Command (m for help): w

创建个RAID0

[root@yufei ~]# mdadm -C /dev/md0 -l0 -n2 /dev/sd{c,d}

mdadm: Defaulting to version 1.2 metadata

mdadm: array /dev/md0 started.

[root@yufei ~]# mdadm -Ds >> /etc/mdadm.conf

现在我们用三个设备来做，一个分区sdb1 一个RAID设备md0 一个磁盘sde

下面就来演示LVM的的创建过程

### 1、创建PV

[root@yufei ~]# pvcreate /dev/sdb1 /dev/sde /dev/md0

  Physical volume "/dev/sdb1" successfully created

  Physical volume "/dev/sde" successfully created

  Physical volume "/dev/md0" successfully created

三个设备类型：分区、磁盘、RAID

查看PV创建出来的结果

[root@yufei ~]# pvdisplay

  "/dev/md0" is a new physical volume of "4.00 GiB"

  --- NEW Physical volume ---

  PV Name               /dev/md0

  VG Name

  PV Size               4.00 GiB

  Allocatable           NO

  PE Size               0

  Total PE              0

  Free PE               0

  Allocated PE          0

  PV UUID               66E15n-AkNA-5mu8-dy4S-TGTZ-8UoK-dVH8Gd

  "/dev/sdb1" is a new physical volume of "2.00 GiB"

  --- NEW Physical volume ---

  PV Name               /dev/sdb1

  VG Name

  PV Size               2.00 GiB

  Allocatable           NO

  PE Size               0

  Total PE              0

  Free PE               0

  Allocated PE          0

  PV UUID               gDhuiG-Rfkk-rwoR-D2ux-jlhj-tpRS-ANbrGC

  "/dev/sde" is a new physical volume of "2.00 GiB"

  --- NEW Physical volume ---

  PV Name               /dev/sde

  VG Name

  PV Size               2.00 GiB

  Allocatable           NO

  PE Size               0

  Total PE              0

  Free PE               0

  Allocated PE          0

  PV UUID               zz1nd4-JMZm-DN0O-25Ae-QIQ8-HfM7-nGQJEY

RAID0是4G，分区sdb1是2G，磁盘sde是2G

### 2、创建VG

[root@yufei ~]# vgcreate VG0 /dev/sdb1 /dev/sde /dev/md0

  Volume group "VG0" successfully created

参数说明：

VG0：创建的VG设备的名字，随便起。

后面接的是设备

整个命令的意思就是：把三个设备组合成一个VG0设备

查看VG0的信息

[root@yufei ~]# vgdisplay

  --- Volume group ---

  VG Name               VG0

  System ID

  Format                lvm2

  Metadata Areas        3

  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                3

  Act PV                3

  VG Size               7.99 GiB

  PE Size               4.00 MiB

  Total PE              2045

  Alloc PE / Size       0 / 0

  Free  PE / Size       2045 / 7.99 GiB

  VG UUID               690fRw-0HIo-rlXZ-WpOZ-5xjv-Em36-ZqtKDA

三个设备加起来是8G，我们这里也是显示的8G啊。这里面PE的大小是4M，如果想设置这个值，可以在创建VG的时候加上-s并指定大小

### 3、从VG中划分出LV

[root@yufei ~]# lvcreate -L 1G -n lv1 VG0

  Logical volume "lv1" created

参数说明：

-L 指定创建LV大小

-l 指定创建LV的PE数量

-n LV的名字

上面命令的意思是：从VG0中分出1G的空间给LV1使用

查看LV的信息

[root@yufei ~]# lvdisplay

  --- Logical volume ---

  LV Name                /dev/VG0/lv1

  VG Name                VG0

  LV UUID                U4yTiy-s25E-Qori-oKTd-TUam-ylA2-nv02Ce

  LV Write Access        read/write

  LV Status              available

  # open                 0

  LV Size                1.00 GiB

  Current LE             256

  Segments               1

  Allocation             inherit

  Read ahead sectors     auto

  - currently set to     256

  Block device           253:0

这个LV设备的路径，是系统方便我们管理，而自动设置此格式的。

### 4、格式化LV

[root@yufei ~]# mkfs.ext4 /dev/VG0/lv1

5、挂载使用

[root@yufei ~]# mount /dev/VG0/lv1 /mnt

[root@yufei ~]# df

Filesystem           1K-blocks      Used Available Use% Mounted on

/dev/sda1             15118728   7082808   7267920  50% /

tmpfs                   255784         0    255784   0% /dev/shm

/dev/mapper/VG0-lv1    1032088     34052    945608   4% /mnt

把挂载信息加入到/etc/fstab

/dev/VG0/lv1            /mnt                    ext4    defaults        0 0

为了继续下面的实验，我们向此设备中增加一些数据

[root@yufei ~]# touch /mnt/testfile

[root@yufei ~]# touch /mnt/testfile1

[root@yufei ~]# touch /mnt/testfile2

[root@yufei ~]# ls /mnt

lost+found  testfile  testfile1  testfile2

## 3、LVM的扩容操作

假设：我的这个LV1空间不够用了，我要怎么办呢？

因为我们的VG0空间还有剩余，所以我们可以从VG0中再分配点空间给LV

下面看如何实现

[root@yufei ~]# lvextend -L +1G /dev/VG0/lv1

  Extending logical volume lv1 to 2.00 GiB

  Logical volume lv1 successfully resized

这里面的-L/-l和上面是一样的意思，就是一个是PE值，一个是空间大小

而“+”就是在原有的基础上加多少

如果不加“+”就表示这个LV的空间是多少

注意看上面的信息是说，lv1已经扩展到2G的大小了

但我们用df命令查看还是1G，为什么呢？

[root@yufei ~]# df -H

Filesystem             Size   Used  Avail Use% Mounted on

/dev/sda1               16G   7.3G   7.5G  50% /

tmpfs                  262M      0   262M   0% /dev/shm

/dev/mapper/VG0-lv1    1.1G    35M   969M   4% /mnt

继续看下面的操作

[root@yufei ~]# resize2fs /dev/VG0/lv1

resize2fs 1.41.12 (17-May-2010)

Filesystem at /dev/VG0/lv1 is mounted on /mnt; on-line resizing required

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

Performing an on-line resize of /dev/VG0/lv1 to 524288 (4k) blocks.

The filesystem on /dev/VG0/lv1 is now 524288 blocks long.

这时候再看

[root@yufei ~]# df -H

Filesystem             Size   Used  Avail Use% Mounted on

/dev/sda1               16G   7.3G   7.5G  50% /

tmpfs                  262M      0   262M   0% /dev/shm

/dev/mapper/VG0-lv1    2.2G    36M   2.0G   2% /mnt

哈哈已经变成了2G了

我们再来观察一下看看里面的数据也是正常的

注意，我们整个操作都是在线的哦，没有umount过文件系统啊。

## 4、增VG空间

假设：我的VG0空间不够用了，那要怎么办？

一种方法：通过PV来增加

你可以增加硬盘或者分区，然后再创建PV设备，然后把他加入VG中就可以了。下面我们以硬盘为例/dev/sdf。

1、创建PV

[root@yufei ~]# pvcreate /dev/sdf

  Physical volume "/dev/sdf" successfully created

2、拉升VG

[root@yufei ~]# vgextend VG0 /dev/sdf

  Volume group "VG0" successfully extended

[root@yufei ~]# vgdisplay

  --- Volume group ---

  VG Name               VG0

  System ID

  Format                lvm2

  Metadata Areas        4

  Metadata Sequence No  4

  VG Access             read/write

  VG Status             resizable

  MAX LV                0

  Cur LV                1

  Open LV               1

  Max PV                0

  Cur PV                4

  Act PV                4

  VG Size               9.98 GiB

  PE Size               4.00 MiB

  Total PE              2556

  Alloc PE / Size       512 / 2.00 GiB

  Free  PE / Size       2044 / 7.98 GiB

  VG UUID               690fRw-0HIo-rlXZ-WpOZ-5xjv-Em36-ZqtKDA

此时的VG0已经由8G变成了10G了。

第二种方法，通过扩展RAID设备的容量，来增加VG的容量

（强烈建议先路过此内容，最后再进行此实验）由于前面我们是用RAID0来做的实验，但这个RAID0只能重新创建才能正常扩容（为什么是这样，请参考：http://www.opsers.org/base/add-on-to-the-hard-disk-raid0-invalid-argument-error-message-to-the-construction.html)所以我们需要就把RAID0重新做成RAID5。

0、删除以前做的LVM（参见本文章）,删除RAID0（参见http://www.opsers.org/base/learning-linux-the-day-that-the-system-configuration-in-the-rhel6-disk-array-raid.html）

1、创建RAID5

[root@yufei ~]# mdadm -C /dev/md5 -l5 -n3 /dev/sd{c,d,e}

[root@yufei ~]# mdadm -Ds >> /etc/mdadm.conf

[root@yufei ~]# mdadm -D /dev/md5

/dev/md5:

        Version : 1.2

  Creation Time : Thu Jun  2 17:06:33 2011

     Raid Level : raid5

     Array Size : 4191232 (4.00 GiB 4.29 GB)

这里MD5是4G的空间

2、创建PV

[root@yufei ~]# pvcreate /dev/sdb1 /dev/md5

  Physical volume "/dev/sdb1" successfully created

  Physical volume "/dev/md5" successfully created

3、创建VG

[root@yufei ~]# vgcreate VG0 /dev/sdb1 /dev/md5

  Volume group "VG0" successfully created

4、查看VG大小情况

[root@yufei ~]# vgdisplay VG0 |grep "VG Size"

  VG Size               5.99 GiB

因为后面的LV我们不需要，所以LV就不创建了，直接演示通过向RAID5中增加磁盘来实现VG的扩容

5、向RAID5中增加一磁盘

[root@yufei ~]# mdadm /dev/md5 -a /dev/sdf

mdadm: added /dev/sdf

[root@yufei ~]# mdadm -G /dev/md5 -n4

mdadm: Need to backup 3072K of critical section..

[root@yufei ~]# mdadm -D /dev/md5

/dev/md5:

        Version : 1.2

  Creation Time : Thu Jun  2 17:06:33 2011

     Raid Level : raid5

     Array Size : 6286848 (6.00 GiB 6.44 GB)

这时候RAID5的空间增加了2G了

6、扩展VG

再来查看一下VG的大小，发现还是6G

[root@yufei ~]# vgdisplay VG0 |grep "VG Size"

  VG Size               5.99 GiB

因为/dev/md5的大小变化了，所以我们需要调整PV大小

[root@yufei ~]# pvresize /dev/md5

  Physical volume "/dev/md5" changed

  1 physical volume(s) resized / 0 physical volume(s) not resized

再来看一下VG的容量，发现增加了

[root@yufei ~]# vgdisplay VG0 |grep "VG Size"

  VG Size               7.99 GiB

LVM的减少操作

  既然可以增加，当然也可以减少，下面我们还是分两个方面来给大家讲解

缩小LV

这个和增加LV空间不同，他需要离线来处理，下面是操作步骤，不要把步骤或操作搞错哦，如果系统的根在LVM上的话，可能会导致进入不了系统。所以缩小一定要慎重。

1、umount文件系统

[root@yufei ~]# umount /mnt

2、检测LV

[root@yufei ~]# e2fsck -f /dev/VG0/lv1

e2fsck 1.41.12 (17-May-2010)

Pass 1: Checking inodes, blocks, and sizes

Pass 2: Checking directory structure

Pass 3: Checking directory connectivity

Pass 4: Checking reference counts

Pass 5: Checking group summary information

/dev/VG0/lv1: 15/131072 files (0.0% non-contiguous), 16812/524288 blocks

3、设置缩小到的空间大小

[root@yufei ~]# resize2fs /dev/VG0/lv1 500M

resize2fs 1.41.12 (17-May-2010)

Resizing the filesystem on /dev/VG0/lv1 to 128000 (4k) blocks.

The filesystem on /dev/VG0/lv1 is now 128000 blocks long.

4、缩小LV的大小

这时候，LV的大小其实还没有改变，可以通过lvdisplay查看，我们需要缩小LV的大小

[root@yufei ~]# lvreduce /dev/VG0/lv1 -L 500M

  WARNING: Reducing active logical volume to 500.00 MiB

  THIS MAY DESTROY YOUR DATA (filesystem etc.)

Do you really want to reduce lv1? [y/n]: y

  Reducing logical volume lv1 to 500.00 MiB

  Logical volume lv1 successfully resized

再用lvdisplay查看一次LV的情况，发现此时已经缩小到了500M了

注意：这里面可以用-1500M来操作，就是减少多少的意思，而不加“-”就表示到多少的意思，和上面的类似。

5、重新挂载上查看数据情况

[root@yufei ~]# mount -a

[root@yufei ~]# df -H

Filesystem             Size   Used  Avail Use% Mounted on

/dev/sda1               16G   7.3G   7.5G  50% /

tmpfs                  262M      0   262M   0% /dev/shm

/dev/mapper/VG0-lv1    516M    35M   456M   8% /mnt

[root@yufei ~]# ls /mnt

lost+found  testfile  testfile1  testfile2

一切正常

注意：

1、第2步和第3步不能颠倒。

2、第3步设置缩小到的空间大小，不能小于里面数据的大小

3、第4步缩小LV的大小应与第2步设置的一样

缩小VG

1、umount lv设备

[root@yufei ~]# umount /mnt

1、用pvdisplay查看VG中的设备

[root@yufei ~]# pvdisplay

省略

  --- Physical volume ---

  PV Name               /dev/sdf

  VG Name               VG0

  PV Size               2.00 GiB / not usable 4.00 MiB

  Allocatable           yes

  PE Size               4.00 MiB

  Total PE              511

  Free PE               511

  Allocated PE          0

  PV UUID               YvnosK-Hi2U-21YE-wRFU-63C7-KUdo-1ul6oy

我就以此PV为例来演示

2、移除里面的pv设备

先查看VG的大小

[root@yufei ~]# vgreduce VG0  /dev/sdf

  Removed "/dev/sdf" from volume group "VG0"

再次查看VG的大小，发现已经变小了

删除LVM，并还原系统

如果说要彻底的来移除LVM的话，你需要把创建的步骤反过来操作。

1、umount

2、移除LV

[root@yufei ~]# lvremove /dev/VG0/lv1

Do you really want to remove active logical volume lv1? [y/n]: y

  Logical volume "lv1" successfully removed

3、移除VG

[root@yufei ~]# vgremove VG0

  Volume group "VG0" successfully removed

4、移除PV

[root@yufei ~]# pvdisplay

针对查找出来的，进行清理

[root@yufei ~]# pvremove /dev/sdf /dev/sde /dev/md0 /dev/sdb1

  Labels on physical volume "/dev/sdf" successfully wiped

  Labels on physical volume "/dev/sde" successfully wiped

  Labels on physical volume "/dev/md0" successfully wiped

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

5、 如果是分区的话，使用fdisk 把ID 改回来

6、如果有RAID设备的话，也同样清除掉吧

OK，看到这里，你可以回到 （第二种方法，通过扩展RAID设备的容量，来增加VG的容量） 这里来看了。因为我们下面也需要这里建立的快照。

  逻辑卷(LVM)快照

快照这个词对我们学习Linux的人来说，应该是不会陌生。他就是在某个时刻，为系统做个照片。方便用于还原操作。而我们的LVM的快照和我们[虚拟机](http://www.2cto.com/os/xuniji/)上的快照功能差不多，但他更常用于数据非常大的时候，比如说[数据库](http://www.2cto.com/database/)需要备份，但这时候数据库又在使用中，而这时候的备份肯定是不完全的或者是有问题的。这个时候就需要LVM的快照来帮助我们了。LVM快照还有一种应用，就是用在资源共享的时候，我可以把某个做好的东西做个快照，然后把这个快照共享出来供大家使用，如果这个快照资源损坏了，我们可以迅速创建，并投入使用。

快照的建立

因为我们还没有建立LV，所以我们要建立个LV，然后对其进行格式化、挂载

[root@yufei ~]# l[vc](http://www.2cto.com/kf/ware/vc/)reate -L 500M -n lv1 VG0

  Logical volume "lv1" created

[root@yufei ~]# mkfs.ext4 /dev/VG0/lv1

[root@yufei ~]# mount /dev/VG0/lv1 /mnt

[root@yufei ~]# cp -a /etc/\* /mnt

[root@yufei ~]# df -H

Filesystem             Size   Used  Avail Use% Mounted on

/dev/sda1               16G   7.3G   7.5G  50% /

tmpfs                  262M      0   262M   0% /dev/shm

/dev/mapper/VG0-lv1    508M    38M   445M   8% /mnt

这时候我们的LV就已经建立好，并在里面写入了相应的数据了。下面我们就为这个LV建立个快照。

[root@yufei ~]# lvcreate -L 10M -s -n lv1snap /dev/VG0/lv1

  Rounding up size to full physical extent 12.00 MiB

  Logical volume "lv1snap" created

上述命令的意思就是，为/dev/VG0/lv1 创建一个10M大小，名字为lv1snap的快照。

[root@yufei ~]# ls /dev/VG0/

lv1      lv1snap

我们把他们挂载起来对比一下看看

[root@yufei ~]# mkdir /lv1

[root@yufei ~]# mkdir /lv1snap

[root@yufei ~]# mount /dev/VG0/lv1 /lv1

[root@yufei ~]# mount /dev/VG0/lv1snap /lv1snap/

[root@yufei ~]# df

Filesystem           1K-blocks      Used Available Use% Mounted on

/dev/sda1             15118728   7084428   7266300  50% /

tmpfs                   255784         0    255784   0% /dev/shm

/dev/mapper/VG0-lv1     495844     35018    435226   8% /mnt

/dev/mapper/VG0-lv1     495844     35018    435226   8% /lv1

/dev/mapper/VG0-lv1snap

                        495844     35018    435226   8% /lv1snap

/mnt是我们之前挂载的，后面两个是刚才挂载的，我们发现他们三个是完全一样的大小。

查看里面的文件也是一样的，其实这个快照只记录了当时那一刻LV1的情况，而如果我们再向LV1里面增加数据，这些数据是不会显示在LV1的快照lv1snap中的，只要向LV1中写入的内容不超过我们创建LV1快照的大小（我们指定的是10M），就不会破坏lv1snap，一旦超过10M，那么这个快照就损坏了，失去了他的作用了。这个大家可以自行测试一下吧。

大家可能会这样一个疑问，为什么一个很小的空间（10M）能快照出一个很大的空间（30M）的内容呢？网上的很多资料说的可能都不能让大家彻底明白，而我这么理解的，快照的空间是用来记录源的更改情况，如果源的改动超过了快照的空间，当然快照就不起作用了啊。当一个snapshot创建的时候，仅拷贝原始卷里数据的元数据(meta-data)，创建的时候，并不会有数据的物理拷贝，因此snapshot的创建几乎是实时的。

删除快照

1、umount快照

[root@yufei ~]# umount /dev/mapper/VG0-lv1snap

2、删除快照

[root@yufei ~]# lvremove /dev/VG0/lv1snap

Do you really want to remove active logical volume lv1snap? [y/n]: y

  Logical volume "lv1snap" successfully removed