

Centos7系列（七）逻辑卷详解

原创

Mr大表哥

2017-05-17 09:08:53

评论(1)

216人阅读

博主QQ: 819594300

博客地址: <http://zpf666.blog.51cto.com/>

有什么疑问的朋友可以联系博主，博主会帮你们解答，谢谢支持！

Centos7可以用xfs_growfs来扩大XFS文件系统，用resize2fs 来扩大ext4文件系统，

注意的是 XFS系统只能增长，不能减少！因此如果需要减少LVM的话，分区只能使用ext4了。

```
[root@localhost ~]# pvcreate /dev/sd[bc]
Physical volume "/dev/sdb" successfully created
Physical volume "/dev/sdc" successfully created
[root@localhost ~]# vgcreate vg00 /dev/sd[bc]
Volume group "vg00" successfully created
[root@localhost ~]# lvcreate -L +110M -n lv00 vg00
Rounding up size to full physical extent 112.00 MiB
Logical volume "lv00" created.
[root@localhost ~]# mkfs.xfs /dev/vg00/lv00
meta-data=/dev/vg00/lv00          isize=256    agcount=4, agsize=7168 blks
=                               sectsz=512   attr=2, projid32bit=1
=                               crc=0      finobt=0
data           =                bsize=4096   blocks=28672, imaxpct=25
=                               sunit=0     swidth=0 blks
naming         =version 2        bsize=4096   ascii-ci=0 ftype=0
log            =internal log     bsize=4096   blocks=853, version=2
=                               sectsz=512   sunit=0 blks, lazy-count=1
realtime       =none            extsz=4096   blocks=0, rtextents=0
[root@localhost ~]# mkdir /data
[root@localhost ~]# blkid /dev/vg00/lv00 >> /etc/fstab
[root@localhost ~]# vim /etc/fstab
```

<http://zpf666.blog.51cto.com>

```
#
# /etc/fstab
# Created by anaconda on Sun Dec 25 03:43:23 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=7ccfb9c6-5599-41ba-96f4-9897b6efe0af /boot xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
UUID=3a17ba26-d914-4970-a5a9-3760bcb75e1b /data xfs defaults 0 0
```

<http://zpf666.blog.51cto.com>

```
[root@localhost ~]# mount -a
[root@localhost ~]# df -hT
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	46G	3.4G	43G	8%	/
devtmpfs	devtmpfs	898M	0	898M	0%	/dev
tmpfs	tmpfs	913M	144K	913M	1%	/dev/shm
tmpfs	tmpfs	913M	9.0M	904M	1%	/run
tmpfs	tmpfs	913M	0	913M	0%	/sys/fs/cgroup
/dev/sda1	xfs	497M	157M	340M	32%	/boot
tmpfs	tmpfs	183M	16K	183M	1%	/run/user/0
/dev/sr0	iso9660	4.1G	4.1G	0	100%	/run/media/root/CentOS 7 x86_64
/dev/mapper/vg00-lv00	xfs	109M	5.8M	103M	6%	/data

[root@localhost ~]#

<http://zpf666.blog.51cto.com>

逻辑卷已经创建并且挂在上挂载点了，下面我们来用一个例子来做一下快照：

```
[root@localhost ~]# cp /boot/* /data/
```

cp: 略过目录"/boot/grub"

cp: 略过目录"/boot/grub2"

cp: 写入"/data/initrd-plymouth.img" 出错: 设备上没有空间

cp: 扩展"/data/initrd-plymouth.img" 失败: 设备上没有空间

cp: 无法创建普通文件"/data/symvers-3.10.0-327.el7.x86_64.gz": 设备上没有空间

cp: 无法创建普通文件"/data/System.map-3.10.0-327.el7.x86_64": 设备上没有空间

cp: 无法创建普通文件"/data/vmlinuz-0-rescue-d18250f3aac84e0fbc0c83fec80c5be3": 设备上没有空间

cp: 无法创建普通文件"/data/vmlinuz-3.10.0-327.el7.x86_64": 设备上没有空间

```
[root@localhost ~]# df -hT
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	46G	3.4G	43G	8%	/
devtmpfs	devtmpfs	898M	0	898M	0%	/dev
tmpfs	tmpfs	913M	144K	913M	1%	/dev/shm
tmpfs	tmpfs	913M	9.0M	904M	1%	/run
tmpfs	tmpfs	913M	0	913M	0%	/sys/fs/cgroup
/dev/sda1	xfs	497M	157M	340M	32%	/boot
tmpfs	tmpfs	183M	16K	183M	1%	/run/user/0
/dev/sr0	iso9660	4.1G	4.1G	0	100%	/run/media/root/CentOS 7 x86_64
/dev/mapper/vg00-lv00	xfs	109M	109M	20K	100%	/data

[root@localhost ~]#

拷贝的/boot目录下的所有文件到挂载点/data下，由于/boot目录下的数据大于lvm卷的空间大小，所以出现了cp报错，正好我们可以通过一个知识点lvm的扩展来解决该问题。

```
[root@localhost ~]# lvextend -L +300M /dev/vg00/lv00
```

Size of logical volume vg00/lv00 changed from 112.00 MiB (28 extents) to 412.00 MiB (103 extents).

Logical volume lv00 successfully resized.

```
[root@localhost ~]# xfs_growfs /dev/vg00/lv00
```

```
meta-data=/dev/mapper/vg00-lv00  isize=256    agcount=4, agsize=7168 blks
      =                       sectsz=512   attr=2, projid32bit=1
      =                       crc=0        finobt=0
data      =                       bsize=4096   blocks=28672, imaxpct=25
      =                       sunit=0      swidth=0 blks
naming    =version 2           bsize=4096   ascii-ci=0 ftype=0
log       =internal          bsize=4096   blocks=853, version=2
      =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096   blocks=0, rtextents=0
data blocks changed from 28672 to 105472
```

```
[root@localhost ~]# df -hT
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	46G	3.4G	43G	8%	/
devtmpfs	devtmpfs	898M	0	898M	0%	/dev
tmpfs	tmpfs	913M	144K	913M	1%	/dev/shm
tmpfs	tmpfs	913M	9.0M	904M	1%	/run
tmpfs	tmpfs	913M	0	913M	0%	/sys/fs/cgroup
/dev/sda1	xfs	497M	157M	340M	32%	/boot
tmpfs	tmpfs	183M	16K	183M	1%	/run/user/0
/dev/sr0	iso9660	4.1G	4.1G	0	100%	/run/media/root/CentOS 7 x86_64
/dev/mapper/vg00-lv00	xfs	409M	109M	300M	27%	/data

[root@localhost ~]#

通过lvextend命令可以扩展lvm卷，但是这还没完呢，扩展完了lvm卷≠扩展完了文件系统，所有我们还需要扩展文件系统，因为是xfs格式，我们需要用到xfs_growfs命令

<http://zpf666.blog.51cto.com>

下面我们再来拷贝一次/boot目录下的所有数据到挂载点/data下：

cp命令

目录基本操作

cp命令用来将一个或多个源文件或者目录复制到指定的目的文件或目录。它可以将单个源文件复制成一个指定文件名的具体的文件或一个已经存在的目录下。cp命令还支持同时复制多个文件，当一次复制多个文件时，目标文件参数必须是一个已经存在的目录，否则将出现错误。

语法

cp(选项)(参数)

选项

- a: 此参数的效果和同时指定“-dpR”参数相同;
- d: 当复制符号连接时，把目标文件或目录也建立为符号连接，并指向与源文件或目录连接的原始文件或目录;
- f: 强行复制文件或目录，不论目标文件或目录是否已存在;
- i: 覆盖既有文件之前先询问用户;
- l: 对源文件建立硬连接，而非复制文件;
- p: 保留源文件或目录的属性;
- R/r: 递归处理，将指定目录下的所有文件与子目录一并处理;
- s: 对源文件建立符号连接，而非复制文件;
- u: 使用这项参数后只会在源文件的更改时间较目标文件更新时或是名称相对应的目标文件并不存在时，才复制文件;
- S: 在备份文件时，用指定的后缀“SUFFIX”代替文件的默认后缀;
- b: 覆盖已存在的文件目标前将目标文件备份;
- v: 详细显示命令执行的操作。

<http://zpf666.blog.51cto.com>

```
[root@localhost ~]# /bin/cp -rf /boot/* /data
[root@localhost ~]# df -hT
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	46G	3.4G	43G	8%	/
devtmpfs	devtmpfs	898M	0	898M	0%	/dev
tmpfs	tmpfs	913M	144K	913M	1%	/dev/shm
tmpfs	tmpfs	913M	9.0M	904M	1%	/run
tmpfs	tmpfs	913M	0	913M	0%	/sys/fs/cgroup
/dev/sda1	xfs	497M	157M	340M	32%	/boot
tmpfs	tmpfs	183M	16K	183M	1%	/run/user/0
/dev/sr0	iso9660	4.1G	4.1G	0	100%	/run/media/root/CentOS 7 x86_64
/dev/mapper/vg00-lv00	xfs	409M	138M	271M	34%	/data

在cp前面加个/bin
可以在拷贝过程中
省略输入很多“y”
的麻烦。

可以看见新拷过来的/boot目录下的数据占用了138M的空间大小。现在我们就做快照，创建的快照的大小不能低于这个138M。

<http://zpf666.blog.51cto.com>

```
[root@localhost ~]# lvcreate --size 140M --snapshot --name lvsp00 /dev/vg00/lv00
Logical volume "lvsp00" created.
[root@localhost ~]# lvscan
```

ACTIVE	Logical Volume	Size	Origin	Copy-on-Write	Permissions
ACTIVE	/dev/centos/swap	4.00 GiB	inherit		
ACTIVE	/dev/centos/root	45.51 GiB	inherit		
ACTIVE	Original /dev/vg00/lv00	412.00 MiB	inherit		
ACTIVE	Snapshot /dev/vg00/lvsp00	140.00 MiB	inherit		

我们给快照140M
的空间，其实快照
也是一个lvm卷。

<http://zpf666.blog.51cto.com>

快照也创建完毕，下面我们破坏挂载点的数据，通过恢复快照来恢复数据。

```

root@localhost ~]# rm -rf /data/*
root@localhost ~]# umount /dev/vg00/lv00
root@localhost ~]# lvconvert --merge /dev/vg00/lv00
Merging of volume lv00 started.
lv00: Merged: 100.0%
root@localhost ~]# mount /dev/vg00/lv00 /data/
root@localhost ~]# cd /data/
root@localhost data]# ls
config-3.10.0-327.el7.x86_64
grub
grub2
initramfs-0-rescue-d18250f3aac84e0fbc0c83fec80c5be3.img
initramfs-3.10.0-327.el7.x86_64dump.img
initrd-plymouth.img
sysvers-3.10.0-327.el7.x86_64.gz
System.map-3.10.0-327.el7.x86_64
vmlinuz-0-rescue-d18250f3aac84e0fbc0c83fec80c5be3
vmlinuz-3.10.0-327.el7.x86_64

```

用快照恢复丢失的数据的方法思路是先卸载挂载，然后用lvconvert命令将恢复与原lvm卷合并，最后再把lvm卷挂载到原挂载点就恢复数据了。

<http://zpf666.blog.51cto.com>

一旦恢复了快照，快照就自动消失了，如果有必要的话还可以重新建一次快照。

逻辑卷快照说明：

LVM提供一个极妙的设备，它是snapshot。允许管理员创建一个新的块装置，在某个时间点提供了一个精确的逻辑卷副本，快照提供原始卷的静态视图LVM 快照通过把文件系统的改变记录到一个快照分区，因此当你创建一个快照分区时，你不需要使用和你正创建快照的分区一样大小的分区，所需的空间大小取决于快照的使用，所以没有可循的方法来设置此大小。如果快照的大小等于原始卷的大小那么快照永远可用。快照是特殊的逻辑卷，只可以对逻辑卷做快照。逻辑卷快照和需要做快照的逻辑卷必须在同一个卷组里面。

注意：这个快照卷建好之后，是不需要格式化也不需要进行挂载的。格式化或挂载都会出现的错误提示的。

注意：当我们把原逻辑卷里面的数据给删除了，逻辑卷快照里面的数据还在，所以可以用快照恢复数据。而当我们在逻辑卷里面添加数据，快照是不会发生改变的，是没有这个文件的。因为快照只会备份当时逻辑卷的一瞬间。

使用ssm（系统存储管理器）进行逻辑管理

逻辑卷管理器（LVM）是一种极其灵活的磁盘管理工具，它让用户可以从多个物理硬驱创建逻辑磁盘卷，并调整大小，根本没有停机时间。最新版本的CentOS/RHEL 7现在随带系统存储管理器（又叫ssm），这是一种统一的命令行界面，由红帽公司开发，用于管理各种各样的存储设备。目前，有三种可供ssm使用的卷管理后端：LVM、Btrfs和Crypt

准备ssm，在CentOS/RHEL 7上，你需要首先安装系统存储管理器。可以通过rpm或yum工具安装

```

[root@localhost ~]# rpm -q system-storage-manager
未安装软件包 system-storage-manager
[root@localhost ~]#

```

该包默认是未安装的

```

[root@localhost ~]# mount /dev/sr0 /media
mount: /dev/sr0 写保护。将以只读方式挂载
[root@localhost ~]# rpm -ivh /media/Packages/system-storage-manager-0.4-5.el7.noarch.rpm
警告：/media/Packages/system-storage-manager-0.4-5.el7.noarch.rpm: 头V3 RSA/SHA256 Signature, 密钥 ID f4a80eb5: NOKEY
准备中...
正在升级/安装...
  1: system-storage-manager-0.4-5.el7

```

<http://zpf666.blog.51cto.com>


```
[root@localhost ~]# ssm list
```

Device	Free	Used	Total	Pool	Mount point
/dev/sda			50.00 GB		PARTITIONED
/dev/sda1			500.00 MB		/boot
/dev/sda2	0.00 KB	49.51 GB	49.51 GB	centos	
/dev/sdb	19.59 GB	412.00 MB	20.00 GB	vg00	
/dev/sdc	20.00 GB	0.00 KB	20.00 GB	vg00	
/dev/sdd			20.00 GB		
/dev/sde			20.00 GB		

Pool	Type	Devices	Free	Used	Total
centos	lvm	1	0.00 KB	49.51 GB	49.51 GB
vg00	lvm	2	39.59 GB	412.00 MB	39.99 GB

Volume	Pool	Volume size	FS	FS size	Free	Type	Mount point
/dev/centos/root	centos	45.51 GB	xfs	45.49 GB	42.15 GB	linear	/
/dev/centos/swap	centos	4.00 GB				linear	
/dev/vg00/lv00	vg00	412.00 MB	xfs	408.67 MB	255.90 MB	linear	/data
/dev/sda1		500.00 MB	xfs	496.67 MB	339.74 MB	part	/boot

[root@localhost ~]# █

<http://zpf666.blog.51cto.com>

“ssm list” 命令可以查看当前现有磁盘存储设备、存储池、LVM卷和快照信息，因为刚才恢复快照了（一旦恢复快照了，原快照就会自动显示），所以我这没有快照信息。

下面通过ssm创建lvm卷、扩展lvm空间大小以及创建lvm快照：

①创建lvm卷：

```
[root@localhost ~]# mkdir /mnt/test
[root@localhost ~]# ssm create -s 500M -n lv00 --fstype xfs -p mypool /dev/sd[ef]
File descriptor 7 (/dev/urandom) leaked on lvm invocation. Parent PID 13605: /usr/bin/python
Physical volume "/dev/sde" successfully created
Volume group "mypool" successfully created
File descriptor 7 (/dev/urandom) leaked on lvm invocation. Parent PID 13605: /usr/bin/python
Logical volume "lv00" created.
meta- data=/dev/mypool/lv00      isize=256      agcount=4, agsize=32000 blks
      =                  sectsz=512      attr=2, projid32bit=1
      =                  crc=0          finobt=0
data      =                  bsize=4096      blocks=128000, imaxpct=25
      =                  sunit=0        swidth=0 blks
naming    =version 2           bsize=4096      ascii-ci=0 ftype=0
log       =internal log       bsize=4096      blocks=853, version=2
      =                  sectsz=512    sunit=0 blks, lazy-count=1
realtime  =none              extsz=4096      blocks=0, rtextents=0
[root@localhost ~]# blkid /dev/mypool/lv00 >> /etc/fstab
[root@localhost ~]# vim /etc/fstab
```

通过ssm创建lvm卷非常的方便，可以使原来的lvcreate创建lvm的方法的几个步骤只通过一个命令就全部搞定，这一个命令=原方法的（建pv、建vg、建lv、格式化）四个步骤，挂载LVM卷还是需要我们另外手动的。上面那条命令里面的pool(池)也就是卷组是mypool，lvm卷是lv00

<http://zpf666.blog.51cto.com>

```
#
# /etc/fstab
# Created by anaconda on Sun Dec 25 03:43:23 2016
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
/dev/mapper/centos-root / xfs defaults 0 0
UUID=7ccfb9c6-5599-41ba-96f4-9897b6efe0af /boot xfs defaults 0 0
/dev/mapper/centos-swap swap swap defaults 0 0
UUID=07a5c4c3-5dac-4ebe-bbd0-c8e0eccc80f1 /data xfs defaults 0 0
UUID=dbc11bb4-2620-46d3-a576-7074e37c5f51 /mnt/test xfs defaults 0 0
```

<http://zpf666.blog.51cto.com>

```
[root@localhost ~]# mount -a
[root@localhost ~]# df -hT
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	46G	3.4G	43G	8%	/
devtmpfs	devtmpfs	898M	0	898M	0%	/dev
tmpfs	tmpfs	913M	144K	913M	1%	/dev/shm
tmpfs	tmpfs	913M	9.0M	904M	1%	/run
tmpfs	tmpfs	913M	0	913M	0%	/sys/fs/cgroup
/dev/sda1	xfs	497M	157M	340M	32%	/boot
tmpfs	tmpfs	183M	12K	183M	1%	/run/user/0
/dev/sr0	iso9660	4.1G	4.1G	0	100%	/media
/dev/mapper/vg00-lv00	xfs	409M	153M	256M	38%	/data
/dev/mapper/mypool-lv00	xfs	497M	26M	472M	6%	/mnt/test

[root@localhost ~]#

<http://zpf666.blog.51cto.com>

②扩展lvm空间

```
[root@localhost ~]# ssm resize -s +300M /dev/mypool/lv00
```

File descriptor 7 (/dev/urandom) leaked on lvm invocation, Parent PID 49607: /usr/bin/python
Size of logical volume mypool/lv00 changed from 500.00 MiB (125 extents) to 800.00 MiB (200 extents).
Logical volume lv00 successfully resized.

```
meta-data=/dev/mapper/mypool-lv00 isize=256    agcount=4, agsize=32000 blks
        =                               sectsz=512   attr=2, projid32bit=1
        =                               crc=0        finobt=0
data      =                               bsize=4096   blocks=128000, imaxpct=25
        =                               sunit=0      swidth=0 blks
naming    =version 2                   bsize=4096   ascii-ci=0 ftype=0
log       =internal                   bsize=4096   blocks=853, version=2
        =                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                       extsz=4096   blocks=0, rtextents=0
```

data blocks changed from 128000 to 204800

```
[root@localhost ~]# df -hT
```

文件系统	类型	容量	已用	可用	已用%	挂载点
/dev/mapper/centos-root	xfs	46G	3.4G	43G	8%	/
devtmpfs	devtmpfs	898M	0	898M	0%	/dev
tmpfs	tmpfs	913M	144K	913M	1%	/dev/shm
tmpfs	tmpfs	913M	9.0M	904M	1%	/run
tmpfs	tmpfs	913M	0	913M	0%	/sys/fs/cgroup
/dev/sda1	xfs	497M	157M	340M	32%	/boot
tmpfs	tmpfs	183M	12K	183M	1%	/run/user/0
/dev/sr0	iso9660	4.1G	4.1G	0	100%	/media
/dev/mapper/vg00-lv00	xfs	409M	153M	256M	38%	/data
/dev/mapper/mypool-lv00	xfs	797M	26M	772M	4%	/mnt/test

[root@localhost ~]#

“ssm resize”命令扩展lvm卷的空间大小=原来的lvextend命令的两个步骤，用ssm这一次命令即扩展了lvm卷的大小，也扩展了文件系统的大小，比较省事。

<http://zpf666.blog.51cto.com>

③创建lvm快照

```
[root@localhost ~]# cp /etc/{passwd,shadow,group} /mnt/test
[root@localhost ~]# ssm snapshot /dev/mypool/lv00
```

File descriptor 7 (/dev/urandom) leaked on lvm invocation, Parent PID 49732: /usr/bin/python
Logical volume "snap20170108T153054" created.

```
[root@localhost ~]#
```

我们为了测试快照是否有效，拷贝点测试数据，用过cp把三个文件一下子全拷过来。这里在说两个小技巧：

mkdir -p /data/{a,b,c,d} //一条命令就一下子创建了四个目录

useradd zpf && echo "123456" | passwd --stdin zpf //一下子就创建了用户zpf，并且还设置了密码

通过“ssm snapshot”命令创建快照不需要指定快照的名字，快照名字会根据“年月日T十分秒”格式自动生成；另外也不需要分配快照的空间大小，它会自动根据lvm卷的大小的1/5从卷组里面自动分配相应的空间大小。

<http://zpf666.blog.51cto.com>


```
[root@localhost ~]# ssm list
```

Device	Free	Used	Total	Pool	Mount point
/dev/sda			50.00 GB		PARTITIONED
/dev/sda1			500.00 MB		/boot
/dev/sda2	0.00 KB	49.51 GB	49.51 GB	centos	
/dev/sdb	19.59 GB	412.00 MB	20.00 GB	vg00	
/dev/sdc	20.00 GB	0.00 KB	20.00 GB	vg00	
/dev/sdd			20.00 GB		
/dev/sde	19.06 GB	960.00 MB	20.00 GB	mypool	

Pool	Type	Devices	Free	Used	Total
centos	lvm	1	0.00 KB	49.51 GB	49.51 GB
mypool	lvm	1	19.06 GB	960.00 MB	20.00 GB
vg00	lvm	2	39.59 GB	412.00 MB	39.99 GB

Volume	Pool	Volume size	FS	FS size	Free	Type	Mount point
/dev/centos/root	centos	45.51 GB	xfs	45.49 GB	42.12 GB	linear	/
/dev/centos/swap	centos	4.00 GB				linear	
/dev/vg00/lv00	vg00	412.00 MB	xfs	408.67 MB	255.89 MB	linear	/data
/dev/mypool/lv00	mypool	800.00 MB	xfs	496.67 MB	496.54 MB	linear	/mnt/test
/dev/sda1		500.00 MB	xfs	496.67 MB	339.74 MB	part	/boot

Snapshot	Origin	Pool	Volume size	Size	Type
/dev/mypool/snap20170108T153054	lv00	mypool	160.00 MB	16.38 KB	linear

[root@localhost ~]#

<http://zpf666.blog.51cto.com>

说明：一旦lvm卷挂载在挂载点的数据有丢失，可以通过前面的方法我们可以恢复快照(lvconvert --merge)。

版权声明：原创作品，如需转载，请注明出处。否则将追究法律责任