马哥教育
最专业的Linux培训机构

高级文件系统管理

http://www.magedu.com

本章内容

- 设定文件系统配额
- 设定和管理软RAID设备
- 配置逻辑卷
- 设定LVM快照
- btrfs文件系统

配置配额系统

- 综述
  - 在内核中执行
  - 以文件系统为单位启用
  - 对不同组或者用户的策略不同
    根据块或者节点进行限制
    - 执行软限制（soft limit）
    - 硬限制（hard limit）
- 初始化
  - 分区挂载选项：usrquota、grpquota
  - 初始化数据库：quotacheck

为用户设定配额

- 执行
  - 开启或者取消配额：quotaon、quotaoff
  - 直接编辑配额：edquota username
  - 在shell中直接编辑：

    setquota usename 4096 5120 40 50 /foo
  - 定义原始标准用户

    edquota -p user1 user2

报告配额状态

- 报告
  - 用户调查：quota　username
  - 配额概述：repquota　/mountpoint
  - 其它工具：warnquota

什么是RAID

- RAID:Redundant Arrays of Inexpensive（Independent） Disks
- 1988年由加利福尼亚大学伯克利分校（University of California-Berkeley） “A Case for Redundant Arrays of Inexpensive Disks”。
- 多个磁盘合成一个“阵列”来提供更好的性能、冗余，或者两者都提供

- 提高IO能力：
  磁盘并行读写
- 提高耐用性；
  磁盘冗余来实现
- 级别：多块磁盘组织在一起的工作方式有所不同
- RAID实现的方式：
  外接式磁盘阵列：通过扩展卡提供适配能力
  内接式RAID：主板集成RAID控制器
    安装OS前在BIOS里配置
  软件RAID：通过OS实现

RAID级别

- RAID-0：条带卷，strip
- RAID-1：镜像卷，mirror
- RAID-2
- ..
- RAID-5
- RAID-6
- RAID-10
- RAID-01

RAID级别

- RAID-0:

  读、写性能提升；

  可用空间：N*min(S1,S2,...)

  无容错能力

  最少磁盘数：2，2

- RAID-1：

  读性能提升、写性能略有下降；

  可用空间：1*min(S1,S2,...)

  有冗余能力

  最少磁盘数：2，2N

- RAID-4：

  多块数据盘异或运算值，存于专用校验盘

RAID级别

- RAID-5：
  - 读、写性能提升
  - 可用空间：(N-1)*min(S1,S2,...)
  - 有容错能力：允许最多1块磁盘损坏
  - 最少磁盘数：3，3+
- RAID-6：
  - 读、写性能提升
  - 可用空间：(N-2)*min(S1,S2,...)
  - 有容错能力：允许最多2块磁盘损坏
  - 最少磁盘数：4，4+

RAID混合类型级别

- RAID-10：

  读、写性能提升

  可用空间：N*min(S1,S2,...)/2

  有容错能力：每组镜像最多只能坏一块

  最少磁盘数：4，4+

- RAID-01、RAID-50
- RAID7： 可以理解为一个独立存储计算机，自身带有操作系统和管理工具，可以独立运行，理论上性能最高的RAID模式
- JBOD：Just a Bunch Of Disks

  功能：将多块磁盘的空间合并一个大的连续空间使用

  可用空间：sum(S1,S2,...)
- 常用级别：RAID-0，RAID-1，RAID-5，RAID-10，RAID-50，JBOD

软RAID

- mdadm：为软RAID提供管理界面
- 为空余磁盘添加冗余
- 结合内核中的md(multi devices)
- RAID设备可命名为/dev/md0、/dev/md1、/dev/md2、/dev/md3等等

软件RAID的实现

- mdadm：模式化的工具
- 命令的语法格式：mdadm [mode] <raiddevice> [options] <component-devices>
- 支持的RAID级别：LINEAR, RAID0, RAID1, RAID4, RAID5, RAID6, RAID10
- 模式：
  - 创建：-C
  - 装配：-A
  - 监控：-F
  - 管理：-f, -r, -a
- <raiddevice>：/dev/md#
- <component-devices>：任意块设备

软件RAID的实现

- -C：创建模式
  - -n #：使用#个块设备来创建此RAID
  - -l #：指明要创建的RAID的级别
  - -a {yes|no}：自动创建目标RAID设备的设备文件
  - -c CHUNK_SIZE：指明块大小
  - -x #：指明空闲盘的个数
- -D：显示raid的详细信息；

  mdadm -D /dev/md#
- 管理模式：
  - -f：标记指定磁盘为损坏
  - -a：添加磁盘
  - -r：移除磁盘
- 观察md的状态：

  cat /proc/mdstat

软RAID配置示例

- 使用mdadm创建并定义RAID设备

```
#mdadm -C  /dev/md0 -a yes -l 5 -n 3 -x 1  /dev/sdb1
   /dev/sdc1  /dev/sdd1 /dev/sde1
```

- 用文件系统对每个RAID设备进行格式化

```
#mke2fs -j  /dev/md0
```

- 测试RAID设备
- 使用mdadm检查RAID设备的状况

```
#mdadm --detail|D  /dev/md0
```

- 增加新的成员

```
#mdadm -G /dev/md0 -n4 -a /dev/sdf1
```

软RAID测试和修复

- 模拟磁盘故障

  #mdadm /dev/md0 -f /dev/sda1

- 移除磁盘

  #mdadm /dev/md0 -r /dev/sda1

- 从软件RAID磁盘修复磁盘故障
  - 替换出故障的磁盘然后开机
  - 在备用驱动器上重建分区
  - #mdadm /dev/md0 -a /dev/sda1
- mdadm、/proc/mdstat及系统日志信息

软RAID管理

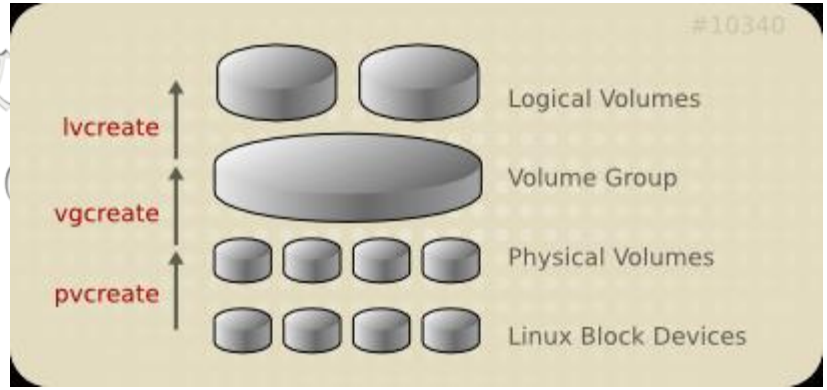
- 生成配置文件：mdadm -D -s >> /etc/mdadm.conf
- 停服务：mdadm -S /dev/md0
- 激活：mdadm -A -s /dev/md0 激活
- 强制启动：mdadm -R /dev/md0
- 删除raid信息：mdadm --zero-superblock /dev/sdb1

练习

- 1：创建一个可用空间为1G的RAID1设备，文件系统为ext4，有一个空闲盘，开机可自动挂载至/backup目录
- 2：创建由三块硬盘组成的可用空间为2G的RAID5设备，要求其chunk大小为256k，文件系统为ext4，开机可自动挂载至/mydata目录

逻辑卷管理器（LVM）

- 允许对卷进行方便操作的抽象层，包括重新设定文件系统的大小
- 允许在多个物理设备间重新组织文件系统
  - 将设备指定为物理卷
  - 用一个或者多个物理卷来创建一个卷组
  - 物理卷是用固定大小的物理区域（Physical Extent，PE）来定义的
  - 在物理卷上创建的逻辑卷是由物理区域（PE）组成
  - 可以在逻辑卷上创建文件系统



LVM介绍

- LVM: Logical Volume Manager, Version: 2
- dm: device mapper：将一个或多个底层块设备组织成一个逻辑设备的模块
- 设备名：/dev/dm-#
- 软链接：

  /dev/mapper/VG_NAME-LV_NAME

  /dev/mapper/vol0-root

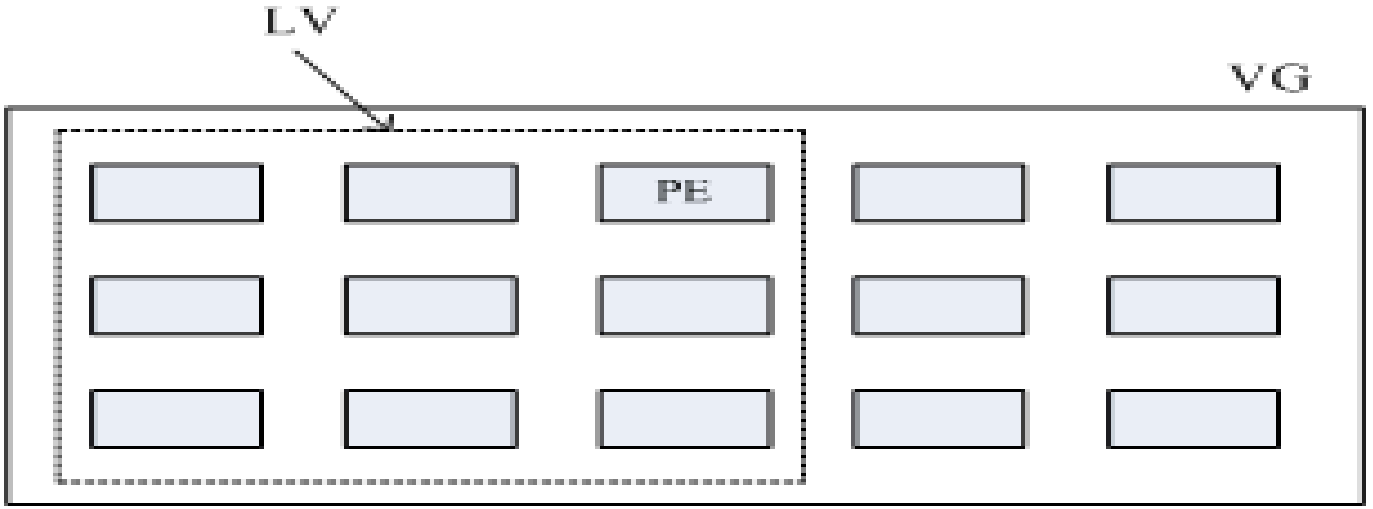
  /dev/VG_NAME/LV_NAME

  /dev/vol0/root

LVM更改文件系统的容量

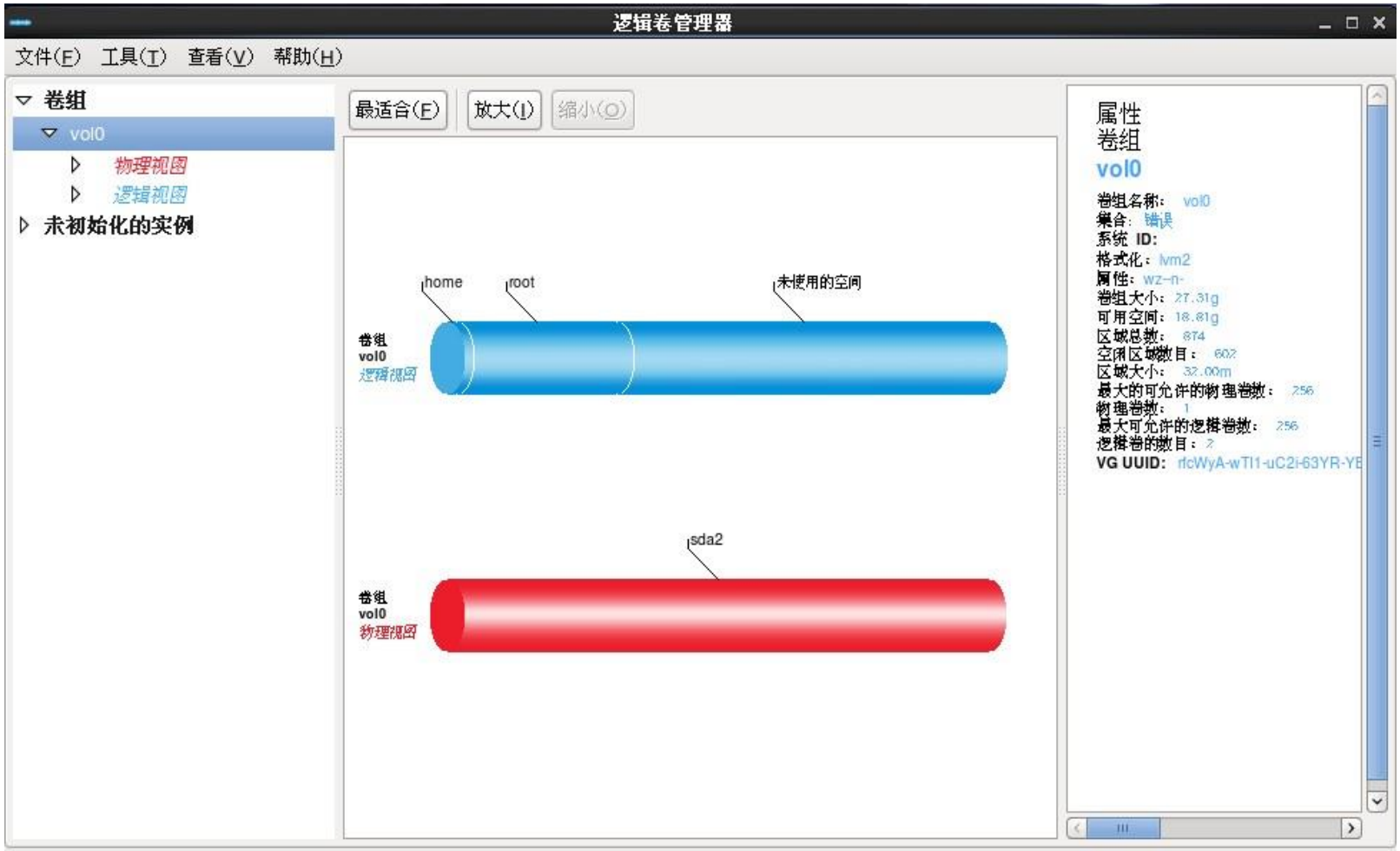
- LVM可以弹性的更改LVM的容量

通过交换PE来进行资料的转换，将原来LV内的PE转移到其他的设备中以降低LV的容量，或将其他设备中的PE加到LV中以加大容量



查看LVM的使用情况

- 点击“系统”->“管理”->“逻辑卷管理器”

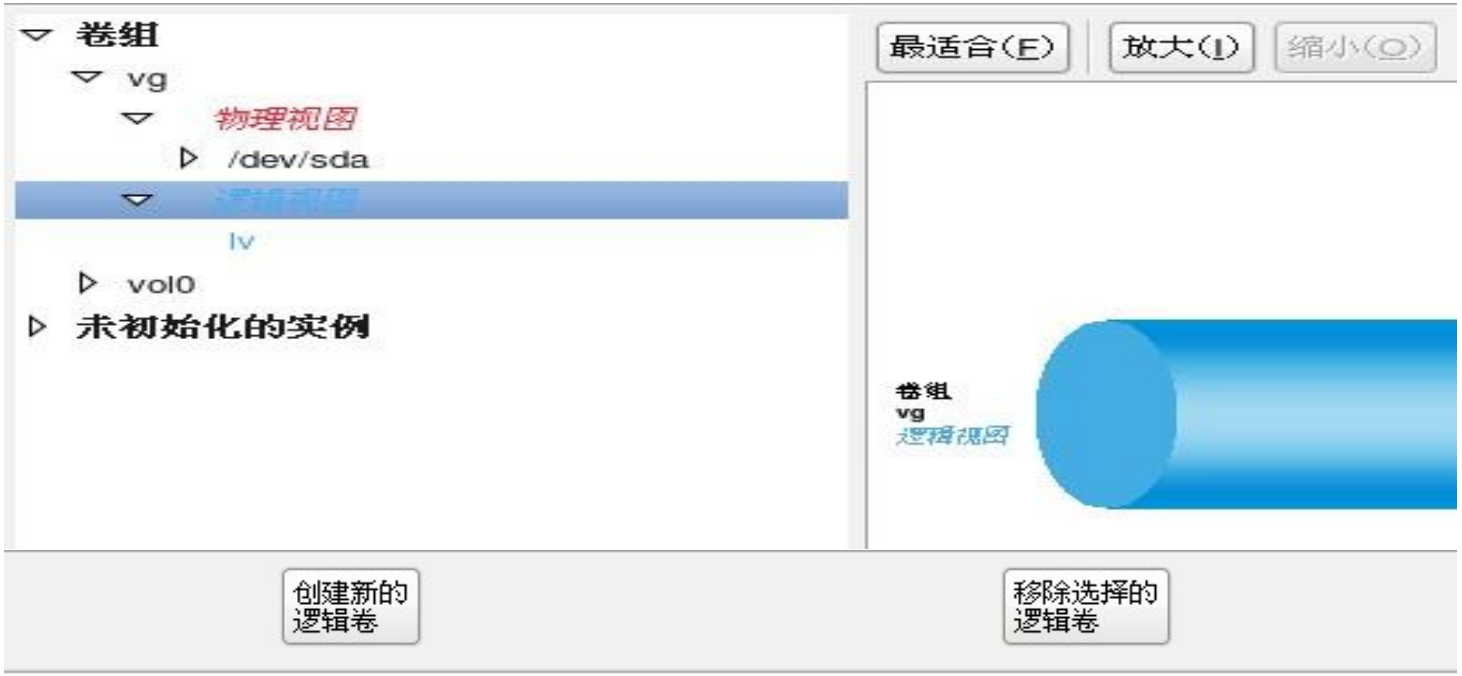


增大或减小逻辑卷

- 打开逻辑卷管理器后，点击“编辑属性”，打开LVM属性对话框：

删除逻辑卷

- 删除逻辑卷必须先删除LV，再删除VG，最后删除PV
- 点击逻辑卷管理器的“卷组”->“逻辑视图”的LV逻辑卷
- 点击“移除选择的逻辑卷”，再删除VG，最后删除PV。



pv管理工具

- 显示pv信息

  pvs：简要pv信息显示

  pvdisplay

- 创建pv

  pvcreate /dev/DEVICE

vg管理工具

- 显示卷组

  vgs

  vgdisplay

- 创建卷组

  vgcreate [-s #[kKmMgGtTpPeE]] VolumeGroupName PhysicalDevicePath [PhysicalDevicePath...]

- 管理卷组

  vgextend VolumeGroupName PhysicalDevicePath [PhysicalDevicePath...]

  vgreduce VolumeGroupName PhysicalDevicePath [PhysicalDevicePath...]

- 删除卷组

  先做pvmove，再做vgremove

lv管理工具

- 显示逻辑卷

  lvs

  Lvdisplay

- 创建逻辑卷

  lvcreate -L #[mMgGtT] -n NAME VolumeGroup

- 删除逻辑卷

  lvremove /dev/VG_NAME/LV_NAME

- 重设文件系统大小

  fsadm [options] resize device [new_size[BKMGTEP]]

  resize2fs [-f][-F][-M][-P][-p] device [new_size]

扩展和缩减逻辑卷

- 扩展逻辑卷：

```
# lvextend -L [+]#[mMgGtT] /dev/VG_NAME/LV_NAME
# resize2fs /dev/VG_NAME/LV_NAME
```

- 缩减逻辑卷：

```
# umount /dev/VG_NAME/LV_NAME
# e2fsck -f /dev/VG_NAME/LV_NAME
# resize2fs /dev/VG_NAME/LV_NAME #[mMgGtT]
# lvreduce -L [-]#[mMgGtT] /dev/VG_NAME/LV_NAME
# mount
```

创建逻辑卷实例

- 创建物理卷

  pvcreate /dev/sda3

- 为卷组分配物理卷

  vgcreate vg0 /dev/sda3

- 从卷组创建逻辑卷

  lvcreate -L 256M -n data vg0

  mke2fs -j /dev/vg0/data

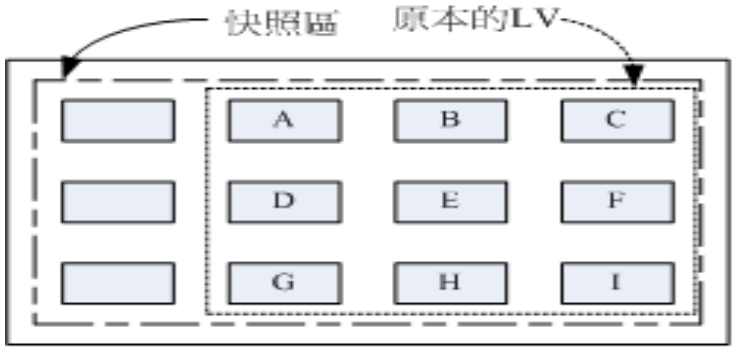
- mount /dev/vg0/data /mnt/data

逻辑卷管理器快照

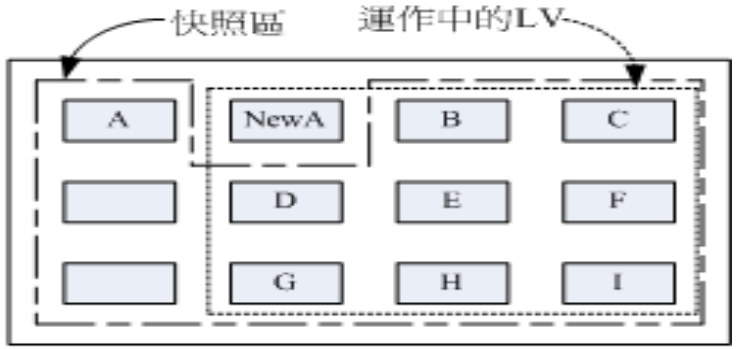
- 快照是特殊的逻辑卷，它是在生成快照时存在的逻辑卷的准确拷贝
- 对于需要备份或者复制的现有数据集临时拷贝以及其它操作来说，快照是最合适的选择。
- 快照只有在它们和原来的逻辑卷不同时才会消耗空间。
  - 在生成快照时会分配给它一定的空间，但只有在原来的逻辑卷或者快照有所改变才会使用这些空间
  - 当原来的逻辑卷中有所改变时，会将旧的数据复制到快照中。
  - 快照中只含有原来的逻辑卷中更改的数据或者自生成快照后的快照中更改的数据
  - 建立快照的卷大小只需要原始逻辑卷的15%～20%就够了。也可以使用lvextend放大快照。

逻辑卷管理器快照

- 快照就是将当时的系统信息记录下来，就好像照相一般，若将来有任何数据改动了，则原始数据会被移动到快照区，没有改动的区域则由快照区和文件系统共享。



此時的A～I的PE爲共用區域



A更動過，快照區保留舊A，未更動的B～I部分的PE爲共用區

由于快照区与原本的**LV**共用很多**PE**的区块，因此快照去与被快照的**LV**必须要要在同一个**VG**上！系统恢复的时候的文件数量不能高于快照区的实际容量。

使用LVM快照

- 为现有逻辑卷创建快照

```
#lvcreate -l 64 -s -n snap-data -p r /dev/vg0/data
```

- 挂载快照

```
#mkdir -p /mnt/snap
#mount -o ro /dev/vg0/snap-data /mnt/snap
```

- 删除快照

```
#umount /mnt/databackup
#lvremove /dev/vg0/databackup
```

练习

- 1、创建一个至少有两个PV组成的大小为20G的名为testvg的VG；要求PE大小为16MB，而后在卷组中创建大小为5G的逻辑卷testlv；挂载至/users目录
- 2、 新建用户archlinux，要求其家目录为/users/archlinux，而后su切换至archlinux用户，复制/etc/pam.d目录至自己的家目录
- 3、扩展testlv至7G，要求archlinux用户的文件不能丢失
- 4、收缩testlv至3G，要求archlinux用户的文件不能丢失
- 5、对testlv创建快照，并尝试基于快照备份数据，验正快照的功能

btrfs文件系统

- 技术预览版
- Btrfs (B-tree, Butter FS, Better FS), GPL, Oracle, 2007, CoW
- 核心特性：
  - 多物理卷支持：btrfs可由多个底层物理卷组成，支持RAID，以及联机“添加”、“移除”，“修改”
  - 写时复制更新机制(CoW)：复制、更新及替换指针，而非“就地”更新
  - 数据及元数据校验码：checksum
  - 子卷：sub_volume
  - 快照：支持快照的快照
  - 透明压缩

btrfs文件系统

- 文件系统创建：
- mkfs.btrfs
  -L 'LABEL'
  -d <type>: raid0, raid1, raid5, raid6, raid10, single
  -m <profile>: raid0, raid1, raid5, raid6, raid10, single, dup
  -O <feature>
    -O list-all: 列出支持的所有feature
  mkfs.btrfs -L mydata -f /dev/sdb /dev/sdc
- 属性查看：
  btrfs filesystem show ; blkid
  btrfs filesystem show –mounted|all-devices
- 挂载文件系统：
  mount -t btrfs /dev/sdb MOUNT_POINT

btrfs文件系统

- 透明压缩机制：

  mount -o compress={lzo|zlib} DEVICE MOUNT_POINT
- 在线修改文件系统大小
- man btrfs

  btrfs filesystem resize -10G /mydata

  btrfs filesystem resize +5G /mydata

  btrfs filesystem resize max /mydata
- 查看

  df -lh; btrfs filesytem df /mydata
- 添加设备:man btrfs-device

  btrfs device add /dev/sdd /mydata

  btrfs filesystem show mydata;df

btrfs文件系统

- 平衡数据:man btrfs-banlance

  btrfs banlance status /mydata

  btrfs banlance start /mydata

  btrfs banlance pause /mydata

  btrfs banlance cancel /mydata

  btrfs banlance resume /mydata

- 删除设备

  btrfs device delete /dev/sdb /mydata

  btrfs filesystem show

- 修改raid级别:注意raid对成员数量的要求

  btrfs balance start -mconvert=raid1|raid0|raid5 /mydata

  btrfs balance start -dconvert=raid1|raid0|raid5 /mydata

Btrfs文件系统

- 子卷管理:man btrfs-subvolume
  btrfs subvolume list /mydata 查看子卷ID等信息
  btrfs subvolume create /mydata/subv1
  umount /mydata
  mount -o subvol=subv1 /dev/sdd /mnt/subv1
  btrfs subvolume show /mnt/subv1
  mount /dev/sdb /mydata 挂父卷，子卷自动挂载
  mount -o subvolid=### /dev/sdd /mnt/subv1

Btrfs文件系统

- 子卷管理

```
btrfs subvolume show /mnt/subv1
btrfs subvolume delete /mydata/subv1
```

- 创建快照：

```
btrfs subvolume snapshot /mydata/subv1 \
/mydata/snapshot_subv1
btrfs subvolume list /mydata
```

- 删除快照

```
btrfs subvolume delete /mydata/snapshot_subv1
```

- 对文件启用CoW（写时复制）

```
cd /mydata/subv1
cp --reflink testfile  testfile2
```

实验：ext4和btrfs互转

- btrfs balance start -dconvert=single /mydata
- btrfs balance start -mconvert=raid1 /mydata
- btrfs device delete /dev/sdd /mydata
- fdisk /dev/sdd分区
- mkfs.ext4 /dev/sdd1
- mount /dev/sdd1 /mnt
- cp /etc/fstab /mnt
- umount /mnt/
- fsck -f /dev/sdd1
- btrfs-convert /dev/sdd1 转化ext4为btrfs
- btrfs filesystem show
- mount /dev/sdd1 /mnt

实验ext4和btrfs互转

- btrfs转化ext4文件系统
- umount /mnt
- btrfs-convert -r /dev/sdd1
- blkid /dev/sdd1

- 再转换成btrfs
- btrfs-convert /dev/sdd1

关于马哥教育

- 博客：http://mageedu.blog.51cto.com
- 主页：http://www.magedu.com
- QQ：1661815153，113228115
- QQ群：203585050，279599283

马哥教育
最专业的Linux培训机构

Thank You!