

4.1 项目导引

天地网络公司在操作系统的选择上采用 Linux 操作系统作为网络操作系统平台。当购买服务器后网络工程师小张需要对磁盘进行分区并格式化,为了对磁盘进行优化、安全管理还需要实现逻辑卷管理、磁盘阵列及磁盘配额管理。

这一案例表明:Linux 操作系统的安全性和稳定性是需要建立在对磁盘合理配置和管理的基础上,以此才能保障操作系统健康稳定地运行。

4.2 项目分析

公司 Red Hat Enterprise Linux AS5 版本的 Linux 已经安装好并成为公司的服务器操作系统,在配置 Linux 时,应重点关注 Linux 系统如何进行硬盘分区、格式化、文件系统的挂载和卸载、逻辑卷管理、磁盘阵列配置和文件系统的配额管理。

通过本章的学习,我们将实现 Linux 磁盘分区操作,可以完成 Linux 磁盘管理中的 LVM、RAID 和磁盘配额设置,并且在掌握 Linux 磁盘基本操作的基础上进行技术拓展,实现 Linux 系统在安装中的 RAID 和 LVM 设置。

4.3 技术准备

4.3.1 文件系统

1. Linux 磁盘分区

在 Linux 中,每一个硬件设备都映射成一个系统文件,对于硬盘、光驱等 IDE 或 SCSI 设备也不例外。Linux 把各种 IDE 设备分配了一个由 hd 前缀组成的文件;而对于各种 SCSI 设备,则分配了一个由 sd 前缀组成的文件。

对于 IDE 硬盘,驱动器标识符为“hdxy”,其中“hd”表明分区所在设备的类型,这里是指 IDE 接口类型硬盘。“x”为盘号,即代表分区所在磁盘是当前接口的第几个设备,以“a”、“b”、“c”等字母标识。“y”代表分区,前四个分区用数字 1 到 4 表示,它们是主分区或扩展分区,从 5 开始就是逻辑分区。例如:hda3 表示为第一个 IDE 硬盘上的第三个主分区或扩展分区,hdb2 表示为第二个 IDE 硬盘上的第二个主分区或扩展分区。对于 SCSI 硬盘则标识为“sdxy”,SCSI 硬盘是用“sd”来表示分区所在设备的类型的,其余则和 IDE 硬盘的表示方法一样。此部分知识在第一章系统安装中已经详细讲述,读者可以自行参阅。

Red Hat Enterprise LinuxAS5 提供一种功能强大的磁盘分区工具 FDISK,FDISK 是 Linux 下的硬盘分区工具,相当重要。

FDISK 命令基本用法为:“fdisk [-l] [设备名称]”,这个命令是查看当前系统磁盘分区情况。FDISK 基本使用方法如下:

```
[root@Server2 ~]# fdisk -l
```

```
Disk /dev/sda: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	1288	10241437+	83	Linux
/dev/sda3		1289	1925	5116702+	83	Linux
/dev/sda4		1926	2610	5502262+	5	Extended
/dev/sda5		1926	2056	1052226	82	Linux swap / Solaris

```
Disk /dev/sdb: 21.4 GB, 21474836480 bytes
255 heads, 63 sectors/track, 2610 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Disk /dev/sdb doesn't contain a valid partition table
```

FDISK 命令可以以交互方式进行磁盘操作,命令格式为“fdisk 设备名称”。

```
[root@Server2 ~]# fdisk /dev/sda
```

```
The number of cylinders for this disk is set to 2610.
```

```
There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:
```

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs

```
(e.g., DOS FDISK, OS/2 FDISK)
```

```
Command (m for help):
```

在 Command 后面可以输入一些交互方式的参数,最常用的是下面几个参数:

- (1)m:显示帮助;
- (2)p:打印当前磁盘分区表;
- (3)n:建立一个新的分区;
- (4)l:显示已知分区类型;
- (5)t:改变分区系统 id;
- (6)d:删除分区;
- (7>w:保存操作;
- (8)q:退出。

2. 创建文件系统

FDISK 完成分区的创建后,接下来就需要进行格式化操作,从而实现在分区创建文件系统。只有创建完文件系统后,此分区才能存取文件。

使用命令 `mkfs` 对分区进行格式化。其基本用法为:`mkfs.ext2 / mkfs.ext3 /mke2fs`
常用的选项有:

- (1)-b:分区时设定每个数据区块占用空间大小,目前支持每个块大小为 1024、2048、4096 字节;
- (2)-i:设定 inode 大小;
- (3)-N:设定 inode 数量,有时使用默认的 inode 数不够用,所以要自行设定 inode 数量;
- (4)-c:在格式化前先检测一下磁盘是否有问题,加上这个选项后会非常慢;
- (5)-L:预设该分区的标签 label;
- (6)-j:建立 ext3 格式的分区,如果使用 `mkfs.ext3` 就不用加这个选项了。

注:进行格式化时,会删除此分区上的所有数据,所以应注意安全,留意备份。`Mkfs.ext3` 基本使用方法如下:

```
[root@Server2 ~]# mkfs.ext3 /dev/sdb1
mke2fs 1.39 (29-May-2008)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
124928 inodes, 497980 blocks
24899 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=67633152
61 block groups
8192 blocks per group, 8192 fragments per group
2048 inodes per group
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409

Writing inode tables: done
Creating journal (8192 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

```
This filesystem will be automatically checked every 26 mounts or
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

3. 检查硬盘有没有坏道

用 FSCK 命令可以在格式化后进行检查硬盘有没有坏道。

当使用 FSCK 检查磁盘有无坏道时,会提示用户“使用这个任务可能会导致某些挂载的文件系统损坏”,所以这个命令不要轻易运行。否则真的遇到问题,系统甚至都不能启动。

(1)-A :加该参数时,后面不需要跟分区名作为参数。它会自动检查/etc/fstab 文件下的所有分区(开机过程中就会执行一次该操作);

(2)-a :自动修复检查有问题的分区;

(3)-r :当检查到有坏道的分区时会让用户决定是否修复。FSCK 命令基本使用如下:

```
[root@Server2 Server]# fsck -a /dev/sdb1
fsck 1.39 (29-May-2006)
/dev/sdb1: clean, 11/124928 files, 26310/497980 blocks
```

4.3.2 挂载、卸载设备

在 Linux 系统中当对一块磁盘进行分区和格式化操作之后,该分区只有在挂载之后才可以给用户使用。下面我们介绍 Linux 提供的几种挂载分区的方法。

1. 挂载磁盘分区

使用 Mount 命令挂载磁盘分区,挂载点必须是一个已经存在的目录。

Mount 命令主要用于挂载设备和显示系统的挂载情况。它的命令语法如下:

```
Mount [参数] [-L<标签>] [-o<选项>] [-t<文件系统类型>] [设备名] [挂载点]
```

在下面的例子中我们将磁盘“/dev/sdc1”挂载到“/mnt/sdc1”目录中,挂载完成后所有对“/mnt/sdc1”中内容的操作都会被保存到“/dev/sdc1”分区中。

```
[root@Server2 ~]# mkdir /mnt/sdc1      创建挂载目录
[root@Server2 ~]# mount /dev/sdc1 /mnt/sdc1  挂载磁盘分区
```

2. 挂载光驱

在 Linux 中硬盘和光驱采用相同的命名方法,假如有一个 IDE 接口的光驱,其设备编号就是/dev/hdc。在 RHEL AS5 中为光驱设备/dev/hdc 创建了两个符号链接分区是/dev/cdrom 和/dev/cdrom-hdc,所有在只有一个 IDE 接口的光驱时,可采用如下三种方式挂载。

```
[root@Server2 dev]# mount /dev/hdc /mnt/
[root@Server2 /]# mount /dev/cdrom /mnt/
[root@Server2 /]# mount /dev/cdrom-hdc /mnt/
```

3. 挂载 U 盘

U 盘和 USB 设备在 Linux 中被认为是一个非 IDE 设备,其设备名和一般硬盘的设备名相同。假如计算机上有一个 SCSI 硬盘,当插入 U 盘后,U 盘所对应的设备名就是/dev/sdb。下面例子中将 U 盘的第一分区挂载到“/mnt/sdb1”中。

```
[root@Server2 ~]# mount /dev/sdb1 /mnt/sdb1
```

4. 挂载 ISO 镜像文件

在 Linux 当中可以直接将 ISO 镜像文件挂载到一个目录中,目录内容就是 ISO 镜像文件中的内容。

```
[root@Server2 ~]# mount -o loop /cdiso/RHEL.iso /mnt/
```

5. 卸载设备

对于已经挂载的设备,如果不需要再使用,可以使用 `umount` 命令卸载设备。

```
[root@Server2 ~]# umount /dev/cdrom 卸载指定设备名
```

```
[root@Server2 ~]# umount /mnt/ 卸载指定挂载点
```

4.3.3 LVM 管理

每个 Linux 使用者在安装 Linux 时都会遇到这样的麻烦:在为系统分区时,如何精确评估和分配各个硬盘分区的容量。因为如果估计不准确,当遇到某个分区不够用时管理员可能要备份整个系统、清除硬盘、重新对硬盘分区、然后恢复数据到新分区,所以系统管理员不但要考虑到当前某个分区需要的容量,还要预见该分区以后可能需要的容量的最大值。虽然现在有很多动态调整磁盘的工具可以使用,例如 Partition Magic 等等,但是它并不能完全解决问题,因为某个分区可能会再次被耗尽;另外一个方面这需要重新引导系统才能实现,对于很多关键的服务器,停机是不可接受的,而且对于添加新硬盘,希望一个能跨越多个硬盘驱动器的文件系统时,分区调整程序就不能解决问题。我们这里所讲述的 Linux LVM 管理可以解决上述问题。

1. LVM 相关名词

LVM 是逻辑盘卷管理(Logical Volume Manager)的简称,它是 Linux 环境下对磁盘分区进行管理的一种机制,LVM 是建立在硬盘和分区之上的一个逻辑层,来提高磁盘分区管理的灵活性。通过 LVM 系统管理员可以轻松管理磁盘分区,如:将若干个磁盘分区连接为一个整块的卷组(Volume Group),形成一个存储池。管理员可以在卷组上随意创建逻辑卷组(Logical Volumes),并进一步在逻辑卷组上创建文件系统。管理员通过 LVM 可以方便的调整存储卷组的大小,并且可以对磁盘存储按照组的方式进行命名、管理和分配,而且当系统添加了新的磁盘,利用 LVM 管理员就不必通过将磁盘的文件移动到新的磁盘上来充分利用新的存储空间,而是直接扩展文件系统跨越磁盘即可。

(1)PV(物理卷):指硬盘分区或从逻辑上与磁盘分区具有同样功能的设备(如 RAID)。

(2)VG(卷组):由一个或多个物理卷组成,类似于逻辑磁盘。

(3)LV(逻辑卷):即逻辑上的分区。

(4)PE(物理范围):物理块,划分物理卷的数据块。

(5)LE(逻辑范围):逻辑块,划分逻辑卷的数据块。

LVM 结构如图 4-1 所示。

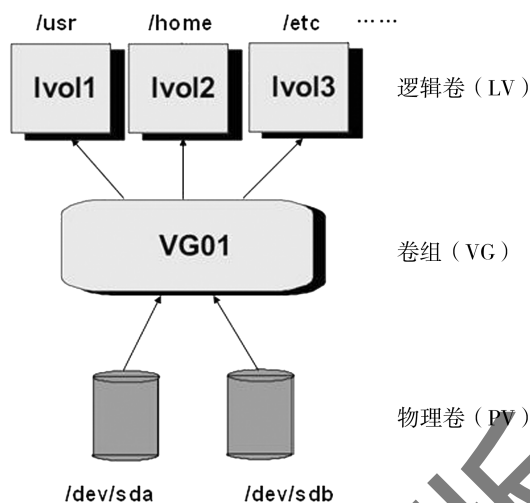


图 4-1 LVM 结构

2. LVM 相关命令

在进行 LVM 之前,需要在 fdisk 中使用“t”将分区类型转换成“Linux LVM”(“8e”)。

(1) pvcreate 命令:用于创建 PV(物理卷)。

```
[root@Server2 ~]# pvcreate /dev/sdb1 /dev/sdb2
```

将两个物理分区创建为物理卷

```
Physical volume "/dev/sdb1" successfully created
```

```
Physical volume "/dev/sdb2" successfully created
```

(2) vgcreate 命令:用于创建 VG(卷组)。

```
[root@Server2 ~]# vgcreate vg1 /dev/sdb1 /dev/sdb2
```

将两个物理卷创建成卷组,卷组命名为 vg1

```
Volume group "vg1" successfully created
```

(3) vgchange 命令:用于激活卷组。

```
[root@Server2 ~]# vgchange -a y vg1
```

```
0 logical volume(s) in volume group "vg1" now active
```

(4) vgextend 命令:用于添加新的物理卷到卷组中。

```
[root@Server2 ~]# vgextend vg1 /dev/sdc1
```

```
Volume group "vg1" successfully extended
```

将物理卷 sdc1 加入到卷组 vg1 中

(5) vgreduce 命令:用于从卷组中删除一个物理卷。

```
[root@Server2 ~]# vgreduce vg1 /dev/sdc1
```

```
Removed "/dev/sdc1" from volume group "vg1"
```

将物理卷 sdc1 从卷组 vg1 中删除

(6) lvcreate 命令:用于创建逻辑卷。

```
[root@Server2 ~]# lvcreate -L5G -n lvm1 vg1
```

```
Logical volume "lvm1" created
```

该命令就在卷组 vg1 上创建名字为 lvm1,大小为 5G 的逻辑卷(vg1 为卷组名,lvm1 为逻辑卷名)。

(7)lvremove 命令:用于删除一个逻辑卷。

```
[root@Server2 ~]# lvremove /dev/vg1/lvm1
```

此命令将逻辑卷 lvm1 删除。

(8)lvextend 命令:用于扩展逻辑卷大小。

```
[root@Server2 ~]# lvextend -L+3G /dev/vg1/lvm1
```

此命令将逻辑卷 lvm1 增加了 3G 空间。

(9)lvreduce 命令:用于减少逻辑卷大小。

```
[root@Server2 ~]# lvreduce -L 3G /dev/vg1/lvm1
```

此命令将逻辑卷 lvm1 的大小减少到 3G。

4.3.4 RAID 管理

RAID 是 Redundent Array of Inexpensive Disks 的缩写,直译为"廉价冗余磁盘阵列",简称为"磁盘阵列"。RAID 通常可以理解成一种使用磁盘驱动器的方法,它将一组磁盘驱动器用某种逻辑方式联系起来,作为逻辑上的一个磁盘驱动器来使用。一般情况下,组成的逻辑磁盘驱动器的容量要小于各个磁盘驱动器容量的总和。RAID 的具体实现可以靠硬件也可以靠软件。

1. 磁盘阵列基础知识

RAID 的优点主要表现在以下方面:

- (1)成本低,功耗小,传输速率高;
- (2)提供容错功能;
- (3)具备数据校验功能;
- (4)性价比高。

2. 常见的 RAID 级别

(1)RAID 0。

无冗余无校验的磁盘阵列。数据同时分布在各个磁盘驱动器上,没有容错能力,读写速度在 RAID 中最快,但因为任何一个磁盘驱动器损坏都会使整个 RAID 系统失效,所以安全系数反倒比单个的磁盘驱动器还要低。一般用在对数据安全要求不高,但对速度要求很高的场合。RAID 0 至少需要两块硬盘,如图 4-2 所示。

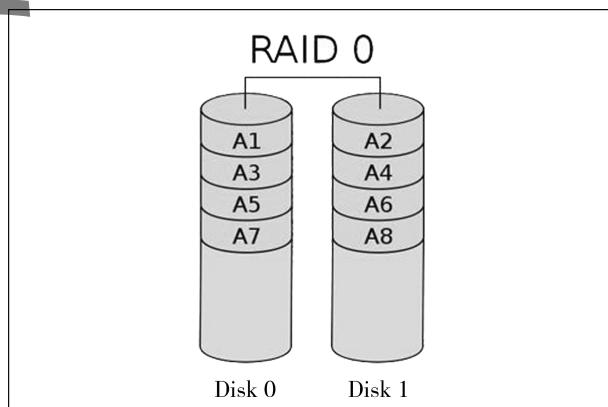


图 4-2 RAID 0 阵列

(2) RAID 1。

也叫镜像磁盘阵列。每一个磁盘驱动器都有一个镜像磁盘驱动器,镜像磁盘驱动器随时保持与原磁盘驱动器的内容一致。RAID1 具有最高的安全性,但只有一半的磁盘空间被用来存储数据。主要用在对数据安全性要求很高,而且要求能够快速恢复被损坏的数据的场合。RAID 1 至少需要两块硬盘,如图 4-3 所示。

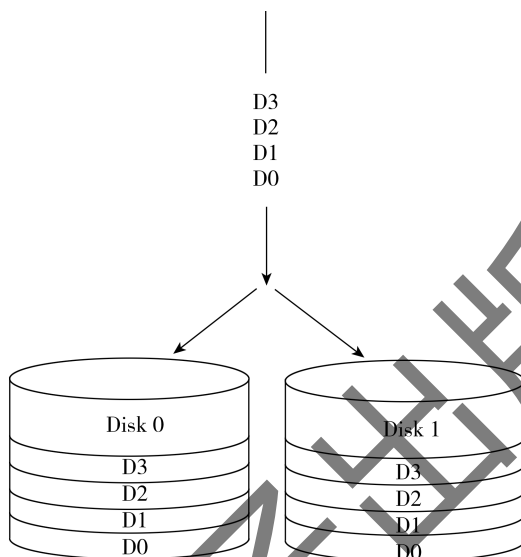


图 4-3 RAID 1 阵列

(3) RAID 5。

RAID 5 是在 RAID 3 的基础上进行了一些改进,当向阵列中的磁盘写数据,奇偶校验数据均匀存放在阵列中的各个盘上,允许单个磁盘出错。RAID 5 也是以数据的校验位来保证数据的安全,但它不是以单独硬盘来存放数据的校验位,而是将数据段的校验位交互存放于各个硬盘上。这样,任何一个硬盘损坏,都可以根据其他硬盘上的校验位来重建损坏的数据。硬盘的利用率也是 $n-1$,如图 4-4 所示。

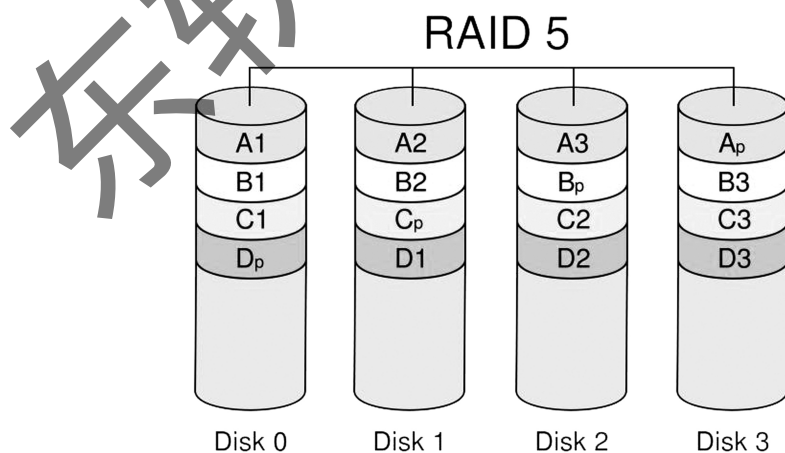


图 4-4 RAID 5 阵列

3. 磁盘阵列常用命令

使用 mdadm 命令将磁盘分区或磁盘加入磁盘阵列的命令如下所述:

基本语法: mdadm [mode] <raid-device> [options] <component-devices>

mdadm [模式] <raid 设备> [参数] <需要加入的 RAID 磁盘>

常用的可用参数包括:

- (1)-A, --assemble: 加入一个以前定义的阵列;
- (2)-C, --create: 创建一个新的阵列;
- (3)-Q, --query: 查看一个 device, 判断它为一个 md device 或是一个 md 阵列的一部分;
- (4)-D, --detail: 打印一个或多个 md device 的详细信息;
- (5)-E, --examine: 打印 device 上的 md superblock 的内容;
- (6)-F, --follow, --monitor: 选择 Monitor 模式;
- (7)-G, --grow: 改变在用阵列的大小或形态;
- (8)-h, --help: 帮助信息, 用在以上选项后, 则显示该选项信息;
- (9)-V, --version: 版本;
- (10)-v, --verbose: 显示细节;
- (11)-b, --brief: 较少的细节。用于 --detail 和 --examine 选项
- (12)-f, --force: 强制;
- (13)-c, --config= : 指定配置文件, 缺省为“/etc/mdadm.conf”。

4.3.5 磁盘配额

如果任何人都可以随意占用 Linux 服务器的硬盘空间, 服务器硬盘还能支撑多久? 所以, 限制和管理用户使用的硬盘空间是非常重要的, 无论是文件服务、FTP 服务还是 Email 服务, 都要求对用户使用的磁盘容量进行有限地控制, 以避免对资源的滥用。Linux 的磁盘配额 (Disk Quotas) 能够简单高效地实现这个功能。

1. 磁盘配额基础知识

Linux 系统中, 由于是多用户多任务的环境, 所以会有多人共同使用一个硬盘空间的情况发生, 如果其中有少数几个使用者使用大量的硬盘空间, 那么其他用户必将受到影响。因此管理员应该适当开放硬盘的权限给使用者, 以便妥善分配系统资源。

Linux 系统的磁盘配额功能用于限制用户所使用的磁盘空间, 并且在用户使用了过多的磁盘空间或分区的空闲过少时, 系统管理员会接到警告。磁盘配额可以针对单独用户进行配置, 也可以针对用户组进行配置。配置的策略也比较灵活, 既可以限制占用磁盘空间, 也可以限制文件的数量。配额必须由 root 用户或者有 root 权限的用户启用和管理。

实现磁盘配额的条件有:

- (1) 确保内核支持 (目前市面上所有常见 Linux 系统都支持)。
- (2) 确保做配额的分区格式是 ext2/ext3/ext4 格式。

只有采用 Linux 的 ext2/ext3/ext4 的文件系统的磁盘分区才能进行磁盘配额。一台文件服务器, 经常会建立单独的分区来存储用户数据, 比如建立一个独立的分区, 格式化成 ext2/ext3/ext4 文件系统, 然后挂载到主系统的目录上进行管理。

(3) 确保系统安装了 quota 软件包。

通过 `rpm -qa | grep quota` 命令来进行查询是否安装了 quota 软件包, 如果显示版本号, 表明系统中安装了该软件包, 否则未安装。

2. 实现磁盘配额步骤

要实现磁盘配额, 请使用以下步骤:

- (1) 检查 Linux 内核是否打开磁盘配额支持;
- (2) 修改 `/etc/fstab`, 对所选文件系统激活配额选项;
- (3) 更新装载文件系统使改变生效;
- (4) 扫描相应文件系统, 用 `quotacheck` 命令生成基本配额文件;
- (5) 用 `edquota` 命令对特定用户或用户组更改磁盘配额;
- (7) 最后用命令激活配额。

3. 管理磁盘配额

如果配额被实现, 它们就需要被维护, 主要维护方式是观察, 即查看配额是否被超出并确保配额的正确性。当然, 如果用户屡次超出他们的配额, 系统管理员就可以根据用户类型和磁盘空间对他们工作的影响来做出几种决策, 管理员可以帮助用户来检索对磁盘空间的使用, 也可以按需要增加用户的配额。

(1) `quotacheck` 命令扫描支持磁盘配额的分区, 并建立磁盘配额文件。

```
[root@Server2 ~]# quotacheck [-afuv] [挂载点]
```

- ① -a: 扫描所有支持磁盘配额的分区, 使用此选项后不必再使用挂载点。
- ② -u: 针对用户扫描文件与目录的使用情况, 会建立 `aquota.user` 文件。
- ③ -g: 针对用户组扫描文件与目录的使用情况, 会建立 `aquota.group` 文件。
- ④ -v: 显示扫描过程的信息。
- ⑤ -f: 强制扫描。

(2) `edquota` 命令更改指定用户或用户组的磁盘配额。

```
[root@Server2 ~]# edquota [-u 用户][ -g 用户组]
```

(3) `quotaon` 命令启用用户和用户组配额功能。

```
[root@Server2 ~]# quotaon [-avug] [挂载点]
```

- ① -a: 扫描所有支持磁盘配额的分区, 使用此选项后不必再使用挂载点。
- ② -u: 针对用户开启磁盘配额。
- ③ -g: 针对用户组开启磁盘配额。
- ④ -v: 显示扫描过程的信息。

(4) `quotaoff` 命令关闭用户和用户组配额功能, 该命令选项与“`quotaon`”命令相同。

```
[root@Server2 ~]# quotaoff [-avug] [挂载点]
```

(5) 用 `repquota` 命令查看磁盘配额。

```
[root@Server2 ~]# repquota -a [挂载点] [-vug]
```

① -a: 显示所有支持磁盘配额分区的使用情况, 使用此选项后不必再使用“挂载点”。挂载点: 查看磁盘配额使用情况的分区。

② -u: 显示用户的磁盘配额。

③ -g: 显示用户组的磁盘配额。

④-v:输出所有的磁盘配额使用情况。

4.4 项目实施

4.4.1 RAID 5 创建

网络工程师小张在服务器上安装了多块 SCSI 类型硬盘,为了保证服务器上重要数据的安全性,将存放数据的磁盘分区设置成 RAID 5。

1. 磁盘分区

(1)小张在 Linux 系统的命令行模式中使用“fdisk -l”命令来查看系统中的硬盘分区情况。

```
[root@Server2 ~]# fdisk -l
```

```
Disk /dev/sda: 21.4 GB, 21474836480 bytes
```

```
255 heads, 63 sectors/track, 2610 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	1	13	104391	83	Linux
/dev/sda2		14	1288	10241437+	83	Linux
/dev/sda3		1289	1925	5116702+	83	Linux
/dev/sda4		1926	2610	5502262+	5	Extended
/dev/sda5		1926	2056	1052226	82	Linux swap / Solaris

```
Disk /dev/sdb: 536.8 GB, 536870912000 bytes
```

```
255 heads, 63 sectors/track, 65270 cylinders
```

```
Units = cylinders of 16065 * 512 = 8225280 bytes
```

```
Disk /dev/sdb doesn't contain a valid partition table
```

小张发现此服务器有两块磁盘,其中“/dev/sda”设备已经有分区,“/dev/sdb”设备没有进行分区,“/dev/sdb”设备将作为服务器的数据盘,小张利用 FDISK 命令对此硬盘进行分区。

(2)小张将此硬盘划分四个分区,其中三个分区配置成 RAID 5,一个分区做成热备份分区,这样当 RAID 5 阵列中有一个分区损坏时,热备份分区会自动补上接替工作。步骤如下:

① 首先进行磁盘分区操作,在命令行中输入“fdisk /dev/sdb”。

```
[root@Server2 ~]# fdisk /dev/sdb
```

```
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklabel
```

```
Building a new DOS disklabel. Changes will remain in memory only,
```

```
until you decide to write them. After that, of course, the previous
```

```
content won't be recoverable.
```

The number of cylinders for this disk is set to 65270.

There is nothing wrong with that, but this is larger than 1024,
and could in certain setups cause problems with:

- 1) software that runs at boot time (e.g., old versions of LILO)
- 2) booting and partitioning software from other OSs

(e.g., DOS FDISK, OS/2 FDISK)

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

②在 Command 后输入“n”，新建分区，首先是建立一个主分区，输入“p”。

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

③接着，输入分区编号。

Partition number (1-4): 1

④输入新建分区的起始柱面，一般不输入直接按回车键。

First cylinder (1-65270, default 1):

Using default value 1

⑤规划主分区大小为 40GB，直接输入大小时使用“+”接分区大小及单位。

Last cylinder or +size or +sizeM or +sizeK (1-65270, default 65270): +40G

⑥在创建完主分区之后，开始建立扩展分区，输入“e”，并指定分区编号为 2。

Command (m for help): n

Command action

e extended

p primary partition (1-4)

e

Partition number (1-4): 2

⑦在建立分区大小时直接按回车键，把磁盘所有的剩余空间都给了扩展分区。

First cylinder (4865-65270, default 4865):

Using default value 4865

Last cylinder or +size or +sizeM or +sizeK (4865-65270, default 65270):

Using default value 65270

⑧在接下来的分区操作中创建了 4 个逻辑分区，输入“l”表示创建逻辑分区，逻辑分区的大小为 40GB。

Command (m for help): n

Command action

l logical (5 or over)

p primary partition (1-4)

l

First cylinder (4865-65270, default 4865):

Using default value 4865

Last cylinder or +size or +sizeM or +sizeK (4865-65270, default 65270): +40G

⑨在完成 4 个逻辑分区的创建之后,在 Command 后输入“p”,查看分区情况,以确保分区创建是否正确,其中第一个逻辑分区编号总是从 5 开始。

Command (m for help): p

```
Disk /dev/sdb: 536.8 GB, 536870912000 bytes
255 heads, 63 sectors/track, 65270 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	4864	39070048+	83	Linux
/dev/sdb2		4865	65270	485211195	5	Extended
/dev/sdb5		4865	9728	39070048+	83	Linux
/dev/sdb6		9729	14592	39070048+	83	Linux
/dev/sdb7		14593	19456	39070048+	83	Linux
/dev/sdb8		19457	24320	39070048+	83	Linux

2. 磁盘分区类型转换

小张将新划分的四个分区的分区类型转换为“Linux raid auto”(“fd”)类型。步骤如下:

(1)在 Command 后面输入“t”,分别将四个分区类型进行转换。

Command (m for help): t

Partition number (1-8): 5

Hex code (type L to list codes): fd

Changed system type of partition 5 to fd (Linux raid autodetect)

Command (m for help): t

Partition number (1-8): 6

Hex code (type L to list codes): fd

Changed system type of partition 6 to fd (Linux raid autodetect)

Command (m for help): t

Partition number (1-8): 7

Hex code (type L to list codes): fd

Changed system type of partition 7 to fd (Linux raid autodetect)

Command (m for help): t

Partition number (1-8): 8

Hex code (type L to list codes): fd

Changed system type of partition 8 to fd (Linux raid autodetect)

(2)转换完之后,在 Command 后面输入“p”,再次查看分区情况。

Command (m for help): p

```
Disk /dev/sdb: 536.8 GB, 536870912000 bytes
```

```
255 heads, 63 sectors/track, 65270 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	4864	39070048+	83	Linux
/dev/sdb2		4865	65270	485211195	5	Extended
/dev/sdb5		4865	9728	39070048+	fd	Linux raid autodetect
/dev/sdb6		9729	14592	39070048+	fd	Linux raid autodetect
/dev/sdb7		14593	19456	39070048+	fd	Linux raid autodetect
/dev/sdb8		19457	24320	39070048+	fd	Linux raid autodetect

(3)最后在 Command 后面输入“w”，保存 FDISK 修改并退出，然后使用“partprobe”命令使新建的分区信息写入磁盘分区表。

```
Command (m for help): w
```

```
The partition table has been altered!
```

```
Calling ioctl() to re-read partition table.
```

```
Syncing disks.
```

```
[root@Server2 ~]# partprobe
```

```
Warning: Unable to open /dev/hdc read-write (Read-only file system). /dev/hdc has been opened read-only.
```

3. RAID 阵列创建

(1)小张使用“mdadm”来创建 RAID 阵列。其中“-C /dev/md0”表示创建的 RAID 设备名为“/dev/md0”，如果要创建多个阵列可使用的 RAID 设备名为“md1、md2...”；“-l 5”表示创建的 RAID 等级是 RAID 5；“-n 3”表示使用三个分区来创建 RAID 阵列；“-x 1”表示使用一个分区作为热备份分区；“/dev/sdb{5,6,7,8}”表示 RAID 阵列中的分区，其中三个分区用于创建 RAID 5，一个分区作为热备份分区。

```
[root@Server2 ~]# mdadm -C /dev/md0 -l 5 -n 3 -x 1 /dev/sdb{5,6,7,8}
mdadm: array /dev/md0 started.
```

(2)使用“mkfs.ext3”命令来格式化 RAID 阵列，并使用“mdadm -D -s”命令让 RAID 阵列永久生效。

```
[root@Server2 ~]# mkfs.ext3 /dev/md0 # 格式化 RAID 阵列
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
9781248 inodes, 19534976 blocks
976748 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=0
597 block groups
```

```
32768 blocks per group, 32768 fragments per group
```

```
16384 inodes per group
```

```
Superblock backups stored on blocks:
```

```
32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,  
4096000, 7962624, 11239424
```

```
Writing inode tables: done
```

```
Creating journal (32768 blocks): done
```

```
Writing superblocks and filesystem accounting information: done
```

```
This filesystem will be automatically checked every 31 mounts or  
180 days, whichever comes first. Use tune2fs -c or -i to override.
```

```
[root@Server2 ~]# mdadm -D -s > /etc/mdadm.conf # RAID 阵列永久生效
```

(3)完成上述配置之后,将这个 RAID 阵列挂载到目录中去使用。

```
[root@Server2 ~]# mount /dev/md0 /raid5
```

(4)使用“mdadm --detail”命令来查看 RAID 阵列。

```
[root@Server2 ~]# mdadm --detail /dev/md0
```

```
/dev/md0:
```

```
Version : 00.90.03
```

```
Creation Time : Wed Dec 26 21:07:50 2012
```

```
Raid Level : raid5
```

```
Array Size : 78139904 (74.52 GiB 80.02 GB)
```

```
Device Size : 39069952 (37.26 GiB 40.01 GB)
```

```
Raid Devices : 3
```

```
Total Devices : 4
```

```
Preferred Minor : 0
```

```
Persistence : Superblock is persistent
```

```
Update Time : Wed Dec 26 21:13:48 2012
```

```
State : clean
```

```
Active Devices : 3
```

```
Working Devices : 4
```

```
Failed Devices : 0
```

```
Spare Devices : 1
```

```
Layout : left-symmetric
```

```
Chunk Size : 64K
```

```
UUID : 74102221:7b480fab:a594b3ff:718a16e0
```

```
Events : 0.106
```

```

Number Major Minor RAID Device State
0 8 21 0 active sync /dev/sdb5
1 8 22 1 active sync /dev/sdb6
2 8 23 2 active sync /dev/sdb7

3 8 24 - spare /dev/sdb8

```

(5) 通过查阅资源得知, 可以使用“mdam /dev/md0 -a /dev/sdc9”将“sdc9”分区加入“/dev/md0”阵列中, 也可以使用“mdam /dev/md0 -r /dev/sdc9”将“sdc9”分区从“/dev/md0”阵列中移除。

4.4.2 LVM 管理

网络工程师小张为了保证数据盘某些分区可以动态扩充和删减容量, 决定将这些分区进行 LVM 管理。

1. 磁盘分区类型转换

在 FDISK 中使用“t”, 将“/dev/sdc7”和“/dev/sdc8”分区类型转换为“Linux LVM” (“8e”)。

```
Command (m for help): p
```

```

Disk /dev/sdc: 536.8 GB, 536870912000 bytes
255 heads, 63 sectors/track, 65270 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		1	2433	19543041	83	Linux
/dev/sdc2		2434	38907	292977405	5	Extended
/dev/sdc5		2434	7297	39070048+	83	Linux
/dev/sdc6		7298	12161	39070048+	83	Linux
/dev/sdc7		12162	17025	39070048+	83	Linux
/dev/sdc8		17026	21889	39070048+	83	Linux

```
Command (m for help): t # 转换分区类型
```

```
Partition number (1-8): 7
```

```
Hex code (type L to list codes): 8e # 将分区类型转换成 8e
```

```
Changed system type of partition 7 to 8e (Linux LVM)
```

```
Command (m for help): t # 转换分区类型
```

```
Partition number (1-8): 8
```

```
Hex code (type L to list codes): 8e # 将分区类型转换成 8e
```

```
Changed system type of partition 8 to 8e (Linux LVM)
```

```
Command (m for help): p # 查看分区表
```



```
Disk /dev/sdc: 536.8 GB, 536870912000 bytes
255 heads, 63 sectors/track, 65270 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdc1		1	2433	19543041	83	Linux
/dev/sdc2		2434	38907	292977405	5	Extended
/dev/sdc5		2434	7297	39070048+	83	Linux
/dev/sdc6		7298	12161	39070048+	83	Linux
/dev/sdc7		12162	17025	39070048+	8e	Linux LVM
/dev/sdc8		17026	21889	39070048+	8e	Linux LVM

2. LVM 操作

(1) 使用“pvcreate”命令将“/dev/sdc7”、“/dev/sdc8”两个物理分区转换成物理卷,可以使用“pvdisplay”来进行查看。

```
[root@Server2 ~]# pvcreate /dev/sdc7 /dev/sdc8
Physical volume "/dev/sdc7" successfully created
Physical volume "/dev/sdc8" successfully created

[root@Server2 ~]# pvdisplay
--- NEW Physical volume ---
PV Name                /dev/sdc7
VG Name
PV Size                37.26 GB
Allocatable           NO
PE Size (KByte)       0
Total PE              0
Free PE               0
Allocated PE         0
PV UUID               UDpGDZ-JQFB-glep-ZXhS-U3eB-yifi-Mejd0c

--- NEW Physical volume ---
PV Name                /dev/sdc8
VG Name
PV Size                37.26 GB
Allocatable           NO
PE Size (KByte)       0
Total PE              0
Free PE               0
Allocated PE         0
PV UUID               43Jonb-ijcF-JXxY-T4cA-8afq-X4R0-o683v0
```

(2) 使用“vgcreate”命令将“/dev/sdc7”、“/dev/sdc8”两个物理卷加入一个新建的名为“vg1”的逻辑卷组中,可以使用“vgdisplay”来查看。

```
[root@Server2 ~]# vgcreate vg1 /dev/sdc7 /dev/sdc8
Volume group "vg1" successfully created
[root@Server2 ~]# vgdisplay
--- Volume group ---
VG Name                vg1
System ID
Format                 lvm2
Metadata Areas        2
Metadata Sequence No  1
VG Access              read/write
VG Status              resizable
MAXLV                 0
CurLV                0
OpenLV                0
Max PV                0
Cur PV               2
Act PV                2
VG Size                74.52 GB
PE Size                4.00 MB
Total PE              19076
Alloc PE / Size       0 / 0
Free PE / Size        19076 / 74.52 GB
VG UUID                3iTwhg-01UE-EgpA-P2o2-pDq4-chzc-zYbc5i
```

(3)使用“lvcreate”命令在“vg1”逻辑卷组中创建名为“lvml”的逻辑卷,使用“-L”指定逻辑卷大小,可以使用“lvdisplay”来查看。

```
[root@Server2 ~]# lvcreate -L40G -n lvml vg1 # 逻辑卷大小 40G
Logical volume "lvml" created
[root@Server2 ~]# lvdisplay
--- Logical volume ---
LV Name                /dev/vg1/lvml
VG Name                vg1
LV UUID                7b4p5o-4P6H-okM9-Qahx-wJyj-JS4D-e1D5R5
LV Write Access        read/write
LV Status              available
# open                 0
LV Size                40.00 GB
Current LE             10240
Segments               2
Allocation              inherit
Read ahead sectors     0
Block device           253:0
```

(4)将逻辑卷 lvml 进行格式化,格式化完成之后进行挂载分区操作。

```
[root@Server2 ~]# mkfs.ext3 /dev/vg1/lvml
mke2fs 1.39 (29-May-2006)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
5242880 inodes, 10485760 blocks
524288 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=0
320 block groups
32768 blocks per group, 32768 fragments per group
16384 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632, 2654208,
    4096000, 7962624

Writing inode tables: done
Creating journal (32768 blocks): done
Writing superblocks and filesystem accounting information:

done

This filesystem will be automatically checked every 20 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
```

```
[root@Server2 ~]# mkdir /lvml #新建目录
```

```
[root@Server2 ~]# mount /dev/vg1/lvml /lvml/ #挂载分区
```

(5) 当发现逻辑卷组不够用时,使用“vgextend”命令进行扩容,小张将物理卷“/dev/sdc9”加入到逻辑卷组“vg1”中。

```
[root@Server2 ~]# vgextend vg1 /dev/sdc9
Volume group "vg1" successfully extended
```

(6) 当发现此逻辑卷容量不够用时,使用“lvextend”命令进行扩容,将此逻辑卷容量增加了10GB空间。

```
[root@Server2 ~]# lvextend -L +10G /dev/vg1/lvml
Extending logical volume lvml to 50.00 GB
Logical volume lvml successfully resized
```

4.4.3 磁盘配额管理

网络工程师小张最近苦恼的是某些用户使用服务器磁盘空间容量太多,导致空间比较紧张,于是决定使用 Linux 磁盘配额管理来限制用户使用磁盘空间,即对“/home”分区进行配额

管理。

1. 更改/etc/fstab 文件

(1) 编辑“/etc/fstab”文件,将包含“LABEL=/home”这一行的“defaults”后面加上usrquota,使分区支持用户、用户组磁盘配额。

```
LABEL=/home /home ext3 defaults,usrquota,grpquota 0 0
```

(2) 使用“mount”命令重新挂载“/home”分区。

```
[root@Server2 ~]# mount -o remount /home/
```

2. 磁盘配额设置

(1) 使用“quotacheck”命令扫描磁盘,并建立磁盘配额所需要的文件。这里使用了“-avug”参数,目的是扫描所有支持磁盘配额的分区,并建立相关的用户配额和用户组配额文件。

```
[root@Server2 ~]# quotacheck -avug
quotacheck: Scanning /dev/sda3 [/home] done
quotacheck: Checked 7 directories and 20 files
```

(2) 使用“edquota”命令来制定用户或用户组的磁盘配额,这里小张使用“edquota -u 用户名”对“mary”和“tom”用户进行文件容量大小存储限制,将他们的软配额“soft”设置成 100MB,硬配额“hard”设置成 200MB,这两个单位默认是 KB,即当两个用户使用空间超过 100MB 时会出现警告,但是仍然可以存储数据,当超过 200MB 时就不允许再存储数据。

```
[root@Server2 ~]# edquota -u tom
Disk quotas for user tom (uid 504):
  Filesystem  blocks   soft   hard   inodes   soft   hard
  /dev/sda3     0  100000 200000     0         0     0
```

(3) 使用“quotaon -avu”来启用针对用户的磁盘配额。

```
[root@Server2 ~]# quotaon -avu
/dev/sda3 [/home]: user quotas turned on
```

(4) 按照同样的方法对 mary 用户设置了磁盘配额。

4.5 技术拓展

在 Linux 基础篇第一章 Linux 系统优化安装策略时我们提出了以下策略:

(1) 建立 2 个 10GB 的分区,在安装时创建软件 RAID 1,选择挂载点是“/”,文件系统类型为 ext3;

(2) 分两个大小是 1G 的分区,在安装时创建 LVM,指定卷组的名字是 VGNAME,指定逻辑卷的名字是 LVNAME,大小为 500M,选择挂载点是“/home”,文件系统类型为 ext3。

在这里,网络工程师小张将根分区(/)做成了软件 RAID 1 磁盘阵列,充分地保护了根分区的数据安全。/home 分区的数据会随着用户使用的情况而不断增加,将此分区利用 LVM(逻辑卷管理)做成逻辑卷,可以动态地增加磁盘容量,这样可以解决/home 分区的容量问题。

1. RAID 1 设置

小张在对公司的另一台 Linux 服务器安装时就采用了系统优化安装策略,在系统初始安装

时将根分区(/)设置成 RAID 1 磁盘阵列,步骤如下:

(1)在系统安装界面中单击“新建”按钮,如图 4-5 所示。



图 4-5 新建分区

(2)在“添加分区”的界面中,将“文件系统类型”的选项设置成“software RAID”,大小设置为 10000MB,如图 4-6 所示。还有一个分区按同样的方法来设置。



图 4-6 创建 RAID 类型分区

(3)当两个“software RAID”分区建立好之后,单击“RAID”按钮进行后续设置,如图 4-7 所示。



图 4-7 RAID 阵列设置

(4) 在“RAID 选项”界面中，选择“创建 RAID 设备”，然后单击“确定”按钮，如图 4-8 所示。



图 4-8 创建 RAID 设备

(5) 在“创建 RAID 设备”界面中，将挂载点选择“/”，RAID 级别设置成“RAID1”，其他设置保持默认，单击“确定”按钮，即完成了 RAID1 的设置，如图 4-9 所示。



图 4-9 创建 RAID1

2. LVM 管理

(1) 小张将“/home”分区做成 LVM, 同样在系统安装界面中单击“新建”按钮, 如图 4-10 所示。



图 4-10 新建分区

(2) 在“添加分区”的界面中, 将“文件系统类型”的选项设置成“physical volume(LVM)”, 大小设置为 1000MB, 如图 4-11 所示。还有一个分区按同样的方法来设置。



图 4-11 创建物理卷

(3) 当两个“physical volume”分区建立好之后,单击“LVM”按钮进行后续设置,如图 4-12 所示。



图 4-12 LVM 设置

(4) 在“制作 LVM 卷组”界面中,将卷组名称设置成“VGNAME”,单击“添加”按钮,将挂载点设置成“/home”,逻辑卷名称设置成“LVNAME”,大小设置成“500MB”,如图 4-13 所示。



图 4-13 制作 LVM 卷组

4.6 本章小结

Red Hat Enterprise Linux AS5 的磁盘管理在企业中应用最多的是磁盘分区、磁盘配额、磁盘 LVM 管理和磁盘阵列管理。

本章开篇就介绍了 Linux 的磁盘分区,读者在弄清楚这些知识之后就可以开始后续的磁盘操作,如 RAID、LVM 和磁盘配额。同时本章最后的技术扩展内容中介绍了 Linux 在安装中的 RAID、LVM 设置,较全面地扩充了 Linux 的磁盘管理知识。

4.7 强化练习

通过 VMware Workstation 建立一台虚拟机,使用 Red Hat Enterprise Linux AS5 并满足以下要求:

- 在 VMware 中添加一块虚拟硬盘,执行以下操作。
 - 使用 fdisk 命令进行磁盘分区,然后使用 fdisk-l 查看分区情况;
 - 使用 mkfs 命令创建文件系统;
 - 使用 mount 和 umount 命令实施挂载和卸载文件系统的操作;
 - 修改配置文件/etc/fstab,在系统启动时自动挂载文件系统。
- 磁盘配额练习。
 - 启动 vim 来编辑/etc/fstab 文件;

- (2)把/etc/fstab 文件中的 home 分区添加用户和组的磁盘配额;
- (3)用 quotacheck 命令创建 aquota.user 和 aquota.group 文件;
- (4)给用户 user01 设置磁盘限额功能;
- (5)将其 blocks 的 soft 设置为 102400、hard 设置为 409600、inodes 设置为 12800、hard 设置为 51200,编辑完成后保存并退出;
- (6)重新启动系统;
- (7)用 quotaon 命令启用 quota 功能;
- (8)切换到用户 user1,查看自己的磁盘限额及使用情况;
- (9)尝试复制大小分别超过磁盘限额软限制和硬限制的文件到用户的主目录下,检验一下磁盘限额功能是否起作用。

东软电子出版社