

第十章: 高级文件系统管理

尚硅谷云计算 Linux 课程

版本: V1.0

讲师: 沈超

- 一、磁盘配额
 - 1、磁盘配额概念
 - 2、磁盘配额条件
 - ◆ 内核必须支持磁盘配额

 $[{\tt root@localhost} ~~] \# {\tt grep} ~{\tt CONFIG_QUOTA} ~/ {\tt boot/config-2.6.32-279.e16.i686}$

CONFIG_QUOTA=y

CONFIG QUOTA NETLINK INTERFACE=y

CONFIG QUOTA DEBUG is not set

CONFIG QUOTA TREE=y

CONFIG_QUOTACTL=y

◆ 系统中必须安装了 quota 工具,我们的 Linux 默认是安装了 quota 工具的,查看命令如下 [root@localhost ~]# rpm -qa | grep quota quota-3.17-16.el6.i686

要支持磁盘配额的分区必须开启磁盘配额功能,这个功能需要手工开启,不再是默认就开启的

- 3、概念
- 1) 用户配额和组配额
- 2) 磁盘容量限制和文件个数限制
- 3) 软限制和硬限制
- 4) 宽限时间

如果用户的空间占用数处于软限制和硬限制之间,统会在用户登陆时警告用户磁盘将满,,这个时间就是宽限时间,默认是7天。如果达到了宽限时间,用户的磁盘占用量还超过软限制,那么软限制就会升级为硬限制。

4、磁盘配额规划

我们开始磁盘配额实验,首先我们来规划下我们的实验:

- ◆ 磁盘配额是限制的普通用户在分区上使用磁盘空间和文件个数的,所以我们需要指定一个分区。那么我们手工建立一个 5GB 的/dev/sdb1 分区,把它挂载到/disk 目录当中。
- ◆ 还有我们需要建立被限制的用户和用户组。那么我们假设需要限制 user1、user2 和 user3 用户,这三个用户属于 test 用户组。
- ◆ 其中 test 组磁盘容量硬限制为 500MB, 软限制 450MB, 文件个数不做限制。user1 用户为了便于测试,磁盘容量硬限制为 50MB, 软限制为 40MB, 文件个数硬限制限制为 10 个, 软限制为 8 个。user2 和 user3 用户磁盘容量硬限制为 250MB, 软限制为 200MB, 文件个数不做限制。
- ◆ 大家发现 user1、user2 和 user3 用户加起来的磁盘容量限制为 550MB, 超过了 test 组的磁



盘容量限制 500MB。这样的话,某个用户可能达不到自己的用户限制,而达到组限制时就不能再写入数据了。也就是说,如果用户限制和组限制同时存在,那么哪个限制更小,哪个限制优先生效。

- ◆ 系统宽限时间我们改为8天。
- 5、磁盘配额步骤
- 1)分 5GB的/dev/sdb1分区,并将它挂载到/disk目录当中
- 2) 建立需要做限制的用户和用户组

[root@localhost ~]# groupadd test
[root@localhost ~]# useradd -G test user1
[root@localhost ~]# useradd -G test user2
[root@localhost ~]# useradd -G test user3
[root@localhost ~]# passwd user1
[root@localhost ~]# passwd user2
[root@localhost ~]# passwd user3

3)、在分区上开启磁盘配额功能

[root@localhost ~]# mount -o remount, usrquota, grpquota /disk #重新挂载/disk 分区,并加入用户和用户组的磁盘配额功能

我们要想永久生效,则需要修改/etc/fstab 文件,改成:

[root@localhost ~]# vi /etc/fstab
/dev/sdb1 /disk ext4 defaults, usrquota, grpquota 0 0
…省略部分输出…
[root@localhost ~]# mount - o remount /disk
#修改配置文件如果想要生效,必须重启系统,否则也需要把分区重新挂载一遍。

4)、建立磁盘配额的配置文件

[root@localhost ~]# quotacheck [选项] [分区名] 选项:

- -a: 扫描/etc/mtab 文件中所有启用磁盘配额功能的分区。如果加入此参数,命令后面 就不需要加入分区名了
- -c: 不管原有的配置文件, 重新扫描并建立新的配置文件
- -u: 建立用户配额的配置文件,也就是生成 aquota. user 文件
- -g: 建立组配额的配置文件, 会生成 aquota. group 文件
- -v: 显示扫描过程
- -m: 强制以读写的方式扫描文件系统,和-M类似。一般扫描根分区时使用。
- -f: 强制扫描文件系统,并写入新的配置文件。一般扫描新添加的硬盘分区时使用

[root@localhost ~]# quotacheck -avug

需要关闭 SELinux, 否则会报错



[root@localhost ~]# 11 /disk/

总用量 24

-rw------ 1 root root 6144 4月 17 01:08 aquota. group

-rw----- 1 root root 6144 4月 17 01:08 aquota.user

#/disk 目录中两个配额配置文件已经建立

如果需要给根分区开启配额功能,需要:

[root@localhost ~]# vi /etc/fstab

UUID=c2ca6f57-b15c-43ea-bca0-f239083d8bd2 / ext4 defaults, usrquota, grpquota

1 1

#开启/分区的配额功能

[root@localhost ~]# mount -o remount /

#重新挂载/分区

[root@localhost ~]# quotacheck -avugm

如果我们自动扫描/分区建立配额配置文件时,因为/分区已经挂载成读写系统,而 quotacheck 需要把分区先挂载成只读分区,然后建立配置文件,最后再挂载回来,所以不能直接在/分区建立配置文件。这时就需要使用-m 强制以读写方式扫描文件系统了

5) 、 设置用户和组的配额限制

[root@localhost ~]# edquota [选项] [用户名或组名] 选项:

-u 用户名: 设定用户配额

-g 组名: 设定组配额

-t: 设定宽限时间

-p: 复制配额限制。如果已经设定好某个用户的配额限制,其他用户的配额限

制如果和这个用户相同,那么可以直接复制配额限制,而不用都手工指定

我们给 user1 用户设定的配额限制是:磁盘空间软限制是 40MB, 硬限制是 50MB; 文件个数的软限制是 8 个, 硬限制是 10 个(稍微小一点,一会测试时方便测试)。命令如下:

[root@localhost ~]# edquota -u user1

#edguota 命令进入之后,就是标准的 vi 操作方法

Disk quotas for user user1 (uid 500):

#磁盘配额是设定用户 user1 (UID 是 500)

blocks Filesystem soft hard inodes soft hard /dev/sdb1 0 0 0 0 0 #分区名 己占用容量 软限制 硬限制 己占用文件数 软限制 硬限制

Disk quotas for user user1 (uid 500):

Filesystem blocks soft hard inodes soft hard /dev/sdb1 0 40000 50000 0 8 10

#不用对齐,是七列就行

再给 user2 用户配置限额, user2 用户要求是空间软限制 250MB, 硬限制 250MB, 文件个数不做限制:

[root@localhost ~]# edquota -u user2



Disk quotas for user user2 (uid 501):

Filesystem blocks soft hard inodes soft hard /dev/sdb1 0 250000 300000 0 0 0

接下来给 test 组配置限额, test 组要求是空间软限制是 450MB, 硬限制 500MB, 文件个数不做限制:

[root@localhost ~]# edquota -g test

Disk quotas for group test (gid 500):

Filesystem blocks soft hard inodes soft hard /dev/sdb1 0 450000 500000 0 0 0

6)、配额复制

user3 用户的配额值和 user2 用户完全一样,我们就可以使用 user2 用户作为模板进行复制。这样我们如果需要建立大量的配额值一致的用户时,就会非常方便,不用一个个手工建立了。复制命令如下:

[root@localhost ~]# edquota -p user2 -u user3 #命令 -p 源用户 -u 目标用户

7)、修改宽限时间

我们要求把宽限时间改为8天,修改命令如下:

[root@localhost ~]# edquota -t

Grace period before enforcing soft limits for users:

Time units may be: days, hours, minutes, or seconds

Filesystem Block grace period Inode grace period

/dev/sdb1 8days 8days #分区名 容量的宽限时间 个数的宽限时间

8) 、启动和关闭配额

配额的配置完成,接下来只需要启动配额就大功告成了,启动命令如下:

[root@localhost ~]# quotaon [选项] [分区名] 选项:

- -a: 依据/etc/mtab 文件启动所有的配额分区。如果不加-a, 后面就一定要指定分区名
- -u: 启动用户配额
- -g: 启动组配额
- -v: 显示启动过程的信息

[root@localhost ~]# quotaon -vug /disk//dev/sdb1 [/disk]: group quotas turned on /dev/sdb1 [/disk]: user quotas turned on

#启动/disk 分区的配额

[root@localhost ~]# quotaon - avug #这条命令也可以

关闭配额的命令如下:

更多云计算-Java -大数据 -前端 -python 人工智能资料下载,可百度访问: 尚硅谷官网



[root@localhost ~]# quotaoff [选项] [分区名] 选项

- -a: 依据/etc/mtab 文件关闭所有的配额分区。如果不加-a, 后面就一定要指定分区名
- -u: 关闭用户配额
- -g: 关闭组配额

[root@localhost ~] # quotaoff - a #依据/etc/mtab 文件关闭配额分区

6、磁盘配额查询

◆ quota 查询用户或用户组配额:

[root@localhost ~]# quota [选项] [用户名或组名] 选项:

-u 用户名: 查询用户配额

-g 组名: 查询组配额

-v: 显示详细信息

-s: 以习惯单位显示容量大小,如 M, G

[root@localhost ~]# quota -uvs user1

◆ repquota 查询文件系统配额

[root@localhost ~]# repquota [选项] [分区名] 选项:

-a: 依据/etc/mtab 文件查询配额。如果不加-a 选项,就一定要加分区名

-u: 查询用户配额

-g: 查询组配额

-v: 显示详细信息

-s: 以习惯单位显示容量大小

[root@localhost ~]# repquota - augvs

7、测试

[userl@localhost disk]\$ dd if=/dev/zero of=/disk/testfile bs=1M count=60 #建立 testfile 文件,指定大小 60MB

8、非交互设定用户磁盘配额

 $[root@localhost ^]$ # setquota -u 用户名 容量软限制 容量硬限制 个数软限制 \个数硬限制 分区名

[root@localhost ~]# useradd user4

[root@localhost ~]# passwd user4

#建立用户

[root@localhost ~]# setquota -u user4 10000 20000 5 8 /disk

#设定用户在/disk 分区的容量软限制为 10MB, 硬限制 20MB。文件个数软限制 5 个, 硬限制#8 个。



这个命令在写脚本批量设置时更加方便。当然写脚本时也可以先建立一个模板的用户,设定好磁 盘配额,再进行配额复制,也是可以的。

二、LVM 逻辑卷管理

1、简介

LVM 是 Logical Volume Manager 的简称,中文就是逻辑卷管理。

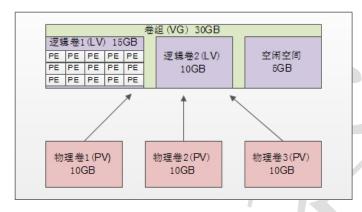


图 10-1 LVM 示意图

- ◆ 物理卷 (PV, Physical Volume): 就是真正的物理硬盘或分区。
- ◆ 卷组(VG, Volume Group): 将多个物理卷合起来就组成了卷组,组成同一个卷组的物理卷可以是同一个硬盘的不同分区,也可以是不同硬盘上的不同分区。我们可以把卷组想象为一个逻辑硬盘。
- ◆ 逻辑卷(LV, Logical Volume): 卷组是一个逻辑硬盘, 硬盘必须分区之后才能使用, 这个分区我们称作逻辑卷。逻辑卷可以格式化和写入数据。我们可以把逻辑卷想象成为分区。
- ◆ 物理扩展(PE, Physical Extend): PE 是用来保存数据的最小单元,我们的数据实际上都是写入PE 当中,PE 的大小是可以配置的,默认是 4MB。
- 2、建立 LVM 的步骤:
- ◆ 首先需要把物理硬盘分成分区,当然也可以是整块物理硬盘。
- ◆ 然后把物理分区建立成为物理卷(PV),也可以直接把整块硬盘都建立为物理卷。
- ◆ 接下来把物理卷整合成为卷组(VG)。卷组就已经可以动态的调整大小了,可以把物理分区 加入卷组,也可以把物理分区从卷组中删除。
- ◆ 最后就是把卷组再划分成为逻辑卷(LV),当然逻辑卷也是可以直接调整大小的。我们说逻辑卷可以想象成为分区,所以也需要格式化和挂载。

3、物理卷管理

1)、硬盘分区

创建方式就是使用 fdisk 交互命令,不过需要注意的是分区的系统 ID 不再是 Linux 默认的分区 ID 号 83 了,而要改成 LVM 的 ID 号 8e。

2)、建立物理卷

[root@localhost ~]# pvcreate [设备文件名]

建立物理卷时,我们说即可以把整块硬盘都建立成物理卷,也可以把某个分区建立成物理卷。如果要把整块硬盘都建立成物理卷,命令如下

更多云计算-Java -大数据 -前端 -python 人工智能资料下载,可百度访问: 尚硅谷官网



[root@localhost ~]# pvcreate /dev/sdb

在我们的使用中,是要把分区建立成为物理卷,所以执行以下命令:

[root@localhost ~]# pvcreate /dev/sdb5

3) 查看物理卷

我们可以看到在我的系统中,/dev/sdb5-7 这三个分区是物理卷。最后一行的意思是: 总共 3 个物理卷[大小] / 使用了 0 个卷[大小] / 空闲 3 个卷[大小]。

第二个查询命令是 pvdi splay, 它可以查看到更详细的物理卷状态, 命令如下:

[root@localhost ~]# pvdisplay

"/dev/sdb5" is a new physical volume of "1.01 GiB"

NEW Physical volume		
PV Name	/dev/sdb5	←PV 名
VG Name		←属于的 VG 名,还没有分配,所以空白
PV Size	1.01 GiB	←PV 的大小
Allocatable	NO	← 是否已经分配
PE Size	0	←PE 大小,因为还没有分配,所以 PE 大小也没有指定
Total PE	0	←PE 总数
Free PE	0	←空闲 PE 数
Allocated PE	0	←可分配的 PE 数
PV UUID	CEsVz3-f0sD-e1w0-wkHZ-iaLq-06aV-xtQNTB ←PV 的 UUID	

4) 删除物理卷

[root@localhost ~]# pvremove /dev/sdb7

- 4、卷组管理
- 1) 建立卷组

[root@localhost ~]# vgcreate [选项] 卷组名 物理卷名 选项:

-s PE 大小: 指定 PE 的大小,单位可以是 MB, GB, TB 等。如果不写默认 PE 大小事 4MB

我们又三个物理卷/dev/sdb5-7,我们先把/dev/sdb5 和/dev/sdb6 加入卷组,留着/dev/sdb7 一会实验调整卷组大小,命令如下:

[root@localhost ~]# vgcreate -s 8MB scvg /dev/sdb5 /dev/sdb6
Volume group "scvg" successfully created

2)、查看卷组

查看卷组的命令同样是两个, vgscan 主要是查看系统中是否有卷组, 而 vgdisplay 则是查看卷组的详细状态的。命令如下:

[root@localhost ~]# vgscan





Reading all physical volumes. This may take a while... Found volume group "scvg" using metadata type 1vm2 #scvg 的卷组确实存在 [root@localhost ~]# vgdisplay --- Volume group ---VG Name ←卷组名 scvg System ID Format 1vm2 Metadata Areas Metadata Sequence No 1 VG Access read/write ←卷组访问状态 VG Status resizable ←卷组状态 MAX LV ←最大逻辑卷数 0 Cur LV 0 Open LV Max PV 0 ←最大物理卷数 Cur PV 2 ←当前物理卷数 Act PV 2 VG Size 2.02 GiB ←卷组大小 PE Size 8.00 MiB ←PE 大小 Total PE 258 ←PE 总数 0 / 0 Alloc PE / Size ←已用 PE 数量/大小 Free PE / Size 258 / 2.02 GiB ←空闲 PE 数量/大小 VG UUID FsOdPf-LV7H-0Ir3-rthA-3UxC-LX5c-FLFriJ

3)、增加卷组容量

[root@localhost ~]# vgextend scvg /dev/sdb7 Volume group "scvg" successfully extended #把/dev/sdb7物理卷也加入 scvg 卷组 [root@localhost ~]# vgdisplay --- Volume group ---VG Name scvg System ID Format 1vm2 Metadata Areas Metadata Sequence No 2 VG Access read/write VG Status resizable MAX LV Cur LV 0 Open LV 0 Max PV



Cur PV 3 Act PV 3 VG Size 3.02 GiB ←卷组容量增加 PE Size 8.00 MiB Total PE 387 ←PE 总数增加 Alloc PE / Size 0 / 0 Free PE / Size 387 / 3.02 GiB VG UUID FsOdPf-LV7H-0Ir3-rthA-3UxC-LX5c-FLFriJ

4)、减小卷组容量

[root@localhost ~]# vgreduce scvg /dev/sdb7

Removed "/dev/sdb7" from volume group "scvg"

#在卷组中删除/dev/sdb7物理卷

[root@localhost ~]# vgreduce -a

#删除所有的未使用物理卷

5) 、删除卷组

[root@localhost ~]# vgremove scvg

Volume group "scvg" successfully removed

卷组删除之后,才能删除删除物理卷。还要注意的是 scvg 卷组还没有添加任何的逻辑卷,那如果拥有了逻辑卷,记得先删除逻辑卷再删除卷组。还记得我刚说的吗?删除就是安装的反过程,每一步都不能跳过。

- 5、逻辑卷管理
- 1)、建立逻辑卷

[root@localhost ~]# lvcreate [选项] [-n 逻辑卷名] 卷组名 选项:

- -L 容量: 指定逻辑卷大小,单位 MB, GB, TB等
- -1 个数:按照 PE 个数指定逻辑卷大小,这个参数需要换算容量,太麻烦
- -n 逻辑卷名: 指定逻辑卷名

那我们就建立一个 1.5GB 的 userlv 逻辑卷吧,建立命令如下:

[root@localhost ~]# lvcreate -L 1.5G -n userlv scvg

Logical volume "userly" created

#在 scvg 卷组中建立 1.5GB 的 userlv 逻辑卷

建立完逻辑卷之后,还要格式化和挂载之后逻辑卷才能正常使用。格式化和挂载命令和操作普通分区时是一样的,不过需要注意的是逻辑卷的设备文件名是/dev/卷组名/逻辑卷名,如我们的 userlv 的设备文件名就是"/dev/scvg/userlv",具体命令如下:

[root@localhost ~]# mkfs -t ext4 /dev/scvg/userlv

#格式化

[root@localhost ~]# mkdir /disklvm

[root@localhost ~]# mount /dev/scvg/userlv /disklvm/

#建立挂载点,并挂载





[root@localhost ~]# mount

…省略部分输出…

/dev/mapper/scvg-userlv on /disklvm type ext4 (rw)

#已经挂载了

当然如果需要开机自动挂载,也要修改/etc/fstab 文件。

2)、查看逻辑卷

同样的查看命令是两个,第一个命令 lvscan 只能看到系统中是否拥有逻辑卷,命令如下:

[root@localhost ~]# lvscan

ACTIVE '/dev/scvg/userlv' [1.50 GiB] inherit

#能够看到激活的逻辑卷, 大小事 1.5GB

第二个命令是 lvdisplay 可以看到逻辑卷的详细信息,命令如下:

[root@localhost ~]# lvdisplay

--- Logical volume ---

LV Path /dev/scvg/userlv ←逻辑卷设备文件名

LV Name userlv ←逻辑卷名
VG Name scvg ←所属的卷组名
LV UUID 2kyKmn-Nupd-CldB-8ngY-NsI3-b8hV-QeUuna

LV Write Access read/write

LV Creation host, time localhost, 2013-04-18 03:36:39 +0800

LV Status available

open 1

LV Size 1.50 GiB ←逻辑卷大小

Current LE 192
Segments 2
Allocation inherit
Read ahead sectors auto

- currently set to 256
Block device 253:0

3) 调整逻辑卷大小

[root@localhost ~]# lvresize [选项] 逻辑卷设备文件名 选项:

- -L 容量: 安装容量调整大小,单位 KB, GB, TB等。使用+代表增加空间,-号代表减少空间。如果直接写容量,代表设定逻辑卷大小为指定大小。
- -1 个数:按照 PE 个数调整逻辑卷大小

我们先在/disklvm中建立点文件,一会调整完大小,我们看看数据是否会丢失:

 $[{\tt root@localhost} \ ^{\tt ~}] \# \ {\tt cd} \ / {\tt disklvm} /$

[root@localhost disklvm]# touch testf

[root@localhost disklvm]# mkdir testd

[root@localhost disklvm]# 1s
lost+found testd testf

我们刚刚的 userlv 的大小事 1.5GB, 我们的 scvg 中还有 1.5GB 的空闲空间, 那么增加我们的

更多云计算-Java -大数据 -前端 -python 人工智能资料下载,可百度访问: 尚硅谷官网





userlv 逻辑卷的大小到 2.5GB 吧:

[root@localhost disklvm]# lvresize -L 2.5G /dev/scvg/userlv

Extending logical volume userly to 2.50 GiB

Logical volume userly successfully resized

#增加 user1v 逻辑卷的大小到 2.5GB

#当然命令也可以这样写 [root@localhost disklvm]# lvresize -L +1G /dev/scvg/userlv

[root@localhost disklvm]# lvdisplay

--- Logical volume ---

LV Path /dev/scvg/userlv

LV Name userlv VG Name scvg

LV UUID 2kyKmn-Nupd-CldB-8ngY-NsI3-b8hV-QeUuna

LV Write Access read/write

LV Creation host, time localhost, 2013-04-18 03:36:39 +0800

LV Status available

open

LV Size 2.50 GiB ←大小改变了

Current LE 320 Segments 3

Allocation inherit
Read ahead sectors auto
- currently set to 256
Block device 253:0

逻辑卷的大小已经改变了,但是好像有些问题啊:

[root@localhost disklvm]# df -h /disklvm/

文件系统 容量 己用 可用 己用‰ 挂载点

/dev/mapper/scvg-userlv 1.5G 35M 1.4G 3%/disklvm

怎么/disklvm 分区的大小还是 1.5GB 啊?刚刚只是逻辑卷的大小改变了,如果需要让分区使用这个新逻辑卷,我们还要使用 resize2fs 命令来调整分区的大小。不过这里就体现了 LVM 的优势,我们不需要卸载分区,直接就能调整分区的大小。resize2fs 命令如下:

[root@localhost ~]# resize2fs [选项] [设备文件名] [调整的大小] 选项:

-f: 强制调整

设备文件名: 指定调整哪个分区的大小

调整的大小: 指定把分区调整到多大,要加 M, G 等单位。如果不加大小,会使用整个

分区

那么我们已经把逻辑卷调整到了 2.5GB, 这时我们就需要把整个逻辑卷都加入/disklvm 分区, 命令如下:

[root@localhost ~]# resize2fs /dev/scvg/userlv

resize2fs 1.41.12 (17-May-2010)

Filesystem at /dev/scvg/userlv is mounted on /disklvm; on-line resizing required old desc_blocks = 1, new_desc_blocks = 1





Performing an on-line resize of /dev/scvg/userlv to 655360 (4k) blocks. The filesystem on /dev/scvg/userlv is now 655360 blocks long. #已经调整了分区大小

[root@localhost ~]# df -h /disklvm/ 文件系统 容量 已用 可用 已用% 挂载点 /dev/mapper/scvg-userlv 2.5G 35M 2.4G 2% /disklvm #分区大小已经是 2.5GB 了 [root@localhost ~]# ls /disklvm/ lost+found testd testf #而且数据并没有丢失

4) 删除逻辑卷

[root@localhost ~]# lvremove 逻辑卷设备文件名

我们删除 userlv 这个逻辑卷,记得删除时要先卸载。命令如下:

[root@localhost ~]# umount /dev/scvg/userlv
[root@localhost ~]# lvremove /dev/scvg/userlv