

XFS

Practical

Exercises

05 - Quotas

Overview

Goals

Define the goals for this lab.

To learn basic quota commands and implementation

To examine the internals of the xfs_quota implementation.

Prerequisites

Basic unix command knowledge.

Filesystems operations (mount, mkfs)

Setup

- A test block device containing an xfs filesystem
- The xfsprogs and quota packages installed

Exercises

Exercise 1 – Configuring Quotas and Quota Reporting

1. Create an entry for this filesystem in fstab, enable user quotas by adding quota to the list of mount options and mount the filesystem.

```
# sudo mkdir /mnt/xfstest
# sudo vi /etc/fstab
/dev/hdb1 /mnt/xfstest xfs defaults,quota 0 0

# sudo mount /mnt/xfstest
```

2. Investigate the following xfs_quota administrative commands

```
# sudo /usr/sbin/xfs_quota -x
xfs_quota> help
xfs_quota> report
xfs_quota> state
xfs_quota> path
```

Create some files on the filesystem and rerun the report command.

Exercise 2 – Quota Controls [user/group]

- run “xfs_quota -x” as root and set quota controls on an ordinary user.

```
# sudo /usr/sbin/xfs_quota -x /mnt/xfstest
```

```
xfs_quota> limit bsoft=10m bhard=20m youruser
```

Investigate how the limit command has affected your user.

```
# /usr/sbin/xfs_quota -c 'quota -v'
```
- Exceeding soft Quota

```
# cd /mnt/xfstest
```

```
# dd if=/dev/urandom of=./testfile1 bs=1k count=15000
```

```
# /usr/sbin/xfs_quota -c 'quota' /mnt/xfstest
```

Note: while your (soft) quota has been exceeded you can still write files.

```
# ls > testfile2
```
- Exceeding hard Quota

```
# rm testfile1 testfile2
```

```
# dd if=/dev/urandom of=./testfile1 bs=1k count=30000
```

```
dd: writing './testfile1': Disk quota exceeded
```

```
20417+0 records in
```

```
20416+0 records out
```

```
20905984 bytes (21 MB) copied, 4.20713 seconds, 5.0 MB/s
```

Run the quota command and examine the output.

```
# /usr/sbin/xfs_quota -c quota
```

```
Disk quotas for User youruser (500)
```

Filesystem	Blocks	Quota	Limit	Warn/Time	Mounted on
/dev/hdb1	20416	10240	20480	00 [6 days]	/mnt/xfstest
- Run the generic repquota command to compare the behavior.

```
# /usr/sbin/repquota /mnt/xfstest
```

```
*** Report for user quotas on device /dev/hdb1
```

```
Block grace time: 7days; Inode grace time: 7days
```

User		Block limits			File limits				
		used	soft	hard	grace	used	soft	hard	grace
youruser	+ -	20416	10240	20480	6days	1	0	0	
- Remove the test files created above and investigate the affects of quotas on holy files.
Holy files can be created with dd

```
eg # dd if=/dev/urandom of=./testfile bs=1k count=1 seek=2000000
```

Compare the outputs of

- ls -hl
 - du -h *
 - /usr/sbin/xfs_quota -c quota
6. Experiment with inode quotas.
- xfs_quota> limit isoft=5 ihard=10 youruser
 - Create files to exceed your soft and hard limits.
- Note: xfs_quota> quota takes a -i option to report on inodes.*
7. Experiment with group quotas
- You will need to remount the filesystem after adding the gquota option to the fstab.
 - xfs_quota commands use -g to indicate they are working with groups.

Exercise 3 – Quota Controls [project]

1. Add pquota to the mount options for your test file system. You will have to remove group quotas if set as they are not compatible with group project quotas.
2. Create /etc/projects and /etc/projid files

```
eg    /etc/projects
      33:/mnt/xfstest/a
      33:/mnt/xfstest/b
      /etc/projid
      testproject:33
      # mkdir /mnt/xfstest/a /mnt/xfstest/b
```

3. Update the projects state and set project limits.

```
# sudo /usr/sbin/xfs_quota -x /mnt/xfstest
xfs_quota> project -s testproject
xfs_quota> print
Filesystem      Pathname
/mnt/xfstest    /dev/hdb1 (pquota)
/mnt/xfstest/a  /dev/hdb1 (project 33, testproject)
/mnt/xfstest/b  /dev/hdb1 (project 33, testproject)
xfs_quota> limit -p bsoft=10m bhard=20m testproject
xfs_quota> quota -vp testproject
Disk quotas for Project #33 (testproject)
Filesystem      Blocks    Quota   Limit Warn/Time   Mounted on
/dev/hdb1       0        10240   20480  00 [-----] /mnt/xfstest
```

Exercise 4 – Examining Quota Internals

1. Examine quota inodes and quota entries.

Examine the quota inodes.

```
sudo xfs_db -xr /dev/hdb1
xfs_db: sb 0
xfs_db: p
...
uquotino = null
pquotino = 132

xfs_db: inode 132
xfs_db: p
...
```

Examine the quota entries.

```
xfs_db: dquot -p testproject
xfs_db: p
diskdq.magic = 0x4451
diskdq.version = 0x1
diskdq.flags = 0x2
diskdq.id = 33
diskdq.blk_hardlimit = 2560
diskdq.blk_softlimit = 1280
diskdq.ino_hardlimit = 0
diskdq.ino_softlimit = 0
diskdq.bcount = 1
diskdq.icount = 2
diskdq.itimer = 0
diskdq.btimer = 0
diskdq.iwarns = 0
diskdq.bwarns = 0
diskdq.rtb_hardlimit = 0
diskdq.rtb_softlimit = 0
diskdq.rtbcount = 0
diskdq.rtbtimer = 0
diskdq.rtbwarns = 0
```

2. Examine inodes of quota controlled files/directories

```
# cd /mnt/xfstest/a
# ls > testfile
# ls -la
 133 .      128 ..    135 testfile

# sudo xfs_db -xr /dev/hdb1
xfs_db: inode 133
xfs_db: p
...
core.projid = 33
core.uid = 0
core.gid = 0
...
```

Questions

1. How would you inform users of their quota violations?

Answers

1. How would you inform users of their quota violations?
 - Email users who are over quota. The generic quota package provides warnquota which is usually executed daily using cron.
 - For interactive users quota commands may be added to shell startup scripts (ie /etc/bash.bashrc.local).
 - Generate a quota report on the user or departments homepage.