Table des matières

Opérations de bases 3
Créer un FS 3
Augmenter un FS 3
Supprimer un FS 3
Réduire un FS 3
Ext3 4
Tips 4
Nombre max de fichiers ouvrables 4
Nombre de fichiers ouverts 5
Backuper et restaurer le MBR 5
Migrer des données d'un vg à un autre 5
Déterminer le type d'un fs (sans fstab) 5
Débloquer le boot si fstab foireux 5
Monter un partage samba au boot 6
Troubleshooting 6
Restaurer des fichiers 6
Impossible d'augmenter un FS (ext3) 6
Erreurs : spurious interrupt 7
parted 8
Monter automatiquement un device USB 8
Eviter les FSCK au boot 8
FS busy 8
Monter un FS freebsd (ufs) 8
Resizer une partition sans LVM 8
Supprimer un répertoire si un FS a été monté dessus 8
Supprimer fichier ouvert 9

Opérations de bases

On travaille en *ext3*.

Créer un FS

On créé le LV :

```
lvcreate -L 2G -n lv_ben rootvg
```

Puis le FS :

```
mkfs.ext3 /dev/rootvg/lv_ben
```

Les paramètres par défaut sont généralement suffisants.

Augmenter un FS

On augmente le LV :

```
lvextend -L +4G /dev/rootvg/lv_ben
```

ou

```
lvextend -L 6G /dev/rootvg/lv_ben
```

Puis on augmente le FS (RHEL 4.x) :

```
ext2online /fs
```

ou (selon la distrib Debian/RHEL 5.x) :

```
resize2fs /dev/rootvg/lv_ben
```

Supprimer un FS

On démonte le FS :

```
umount /tmp/test_ben
```

Si KO :

```
fuser -km /tmp/test_ben
```

On supprime le LV :

```
lvremove /dev/rootvg/lv_ben
```

Réduire un FS

On est obligé de démonter le FS pour cette opération :

```
umount /tmp/test_ben
```

On réduit le FS :

```
root@server9000157:/tmp# umount /tmp/ben

root@server9000157:/tmp# e2fsck -f /dev/rootvg/lv_test
e2fsck 1.35 (28-Feb-2004)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
```

```
/dev/rootvg/lv_test: 11/131072 files (9.1% non-contiguous), 21802/524288 blocks

root@server9000157:/tmp# resize2fs /dev/rootvg/lv_test 128M
resize2fs 1.35 (28-Feb-2004)
Resizing the filesystem on /dev/rootvg/lv_test to 131072 (1k) blocks.
The filesystem on /dev/rootvg/lv_test is now 131072 blocks long.

root@server9000157:/tmp# lvreduce -L 128M /dev/rootvg/lv_test
  WARNING: Reducing active and open logical volume to 128.00 MB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lv_test? [y/n]: y
  Reducing logical volume lv_test to 128.00 MB
  Logical volume lv_test successfully resized

root@server9000157:/tmp# mount /dev/rootvg/lv_test /tmp/ben
```

Ext3

- Afficher les infos sur un lv spécifique :

```
dumpe2fs -h /dev/vg_vrk1/lv_testben
```

- Supprimer le journal du filesystem :

```
tune2fs -O ^has_journal /dev/vg_vrk1/lv_testben
```

- Créer un journal pour un filesystem :

```
tune2fs -O has_journal -J size=128 /dev/vg_vrk1/lv_testben
```

- Bypasser le cache FS

```
mount -o sync -o dirsync /dev/rootvg/lv_test /tmp/test
```

Ensuite on peut estimer le débit d'un disque :

```
dd if=/dev/zero of=/tmp/test/fichier.img bs=1024k count=1000
```

- Modifier les paramètres de fsck automatique

Parfois on veut pouvoir éviter le fsck automatique et le lancer à la main. Ces paramètres peuvent être visualisés grâce à la commande *dumpe2fs* :

```
root@server9000157:/tmp# dumpe2fs /dev/rootvg/lv_test 2>/dev/null|egrep "Max|interval"
Maximum mount count:      22
Check interval:           15552000 (6 months)
```

```
root@server9000157:/tmp# tune2fs -i 14d -c 50 /dev/rootvg/lv_test
tune2fs 1.35 (28-Feb-2004)
Setting maximal mount count to 50
Setting interval between check 1209600 seconds
```

- Déterminer la taille du journal

```
root@server9000157:~# debugfs -R "stat <`tune2fs -l /dev/rootvg/tmp|grep "Journal inode"  \
  |awk '{print $NF}'`>" /dev/rootvg/tmp 2>&1| awk '/Size: / {print $6}'|head -1
33554432

root@server9000157:~# echo $((33554432/1024))
32768
```

Soit 32 Mo.

Tips

Nombre max de fichiers ouvrables

```
root@machine:~# cat /proc/sys/fs/file-max
1593454
```

Nombre de fichiers ouverts

```
root@machine:~# cat /proc/sys/fs/file-nr
3391    969     52427
|    |       |
|    |       |
|        |       maximum open file descriptors
|        total free allocated file descriptors
total allocated file descriptors
(the number of file descriptors allocated since boot)
```

Backuper et restaurer le MBR

- Backup de la table de partition (MBR) :

```
dd if=/dev/hdx of=/chemin/sauvegarde.dd bs=512 count=1
```

- Restauration de la table de partition (MBR) :

```
dd if=/chemin/sauvegarde.dd of=/dev/hdx
```

- Backup de la table de partition étendue uniquement :

```
sfdisk -d /dev/hdx > /chemin/save_etendue.sf
```

- Restauration de la table de partition étendue uniquement :

```
sfdisk /dev/hdx < /chemin/save_etendue.sf
```

Migrer des données d'un vg à un autre

- Créer le lv sur le vg de destination

```
lvcreate -L 4G -n lv_dest vg_dest
```

- On migre les données avec dd

```
dd if=/dev/rootvg/lv_depart of=/dev/vg_dest/lv_dest
```

- Un petit fsck s'impose avant de monter le FS

```
fsck -y /dev/vg_dest/lv_dest
```

Il y a surement plus propre mais c'est rapide pour migrer un grand nombre de LVs (avec une boucle). On peut aussi utiliser *pvmove*, *tar*, *cp*, etc.

Déterminer le type d'un fs (sans fstab)

```
root@server:~# lvcreate -L 128M -n lv_testben rootvg
root@server:~# mkfs.ext3 /dev/rootvg/lv_testben
```

```
root@server:~# lvdisplay /dev/rootvg/lv_testben|grep "Block device"
  Block device           253:12
```

```
root@server:~# ls -l /dev/dm* |fgrep -w "253"|egrep "\-12"
brw-r-----  1 root root 253, 12 Mar  7 10:49 /dev/dm-12
```

```
root@server:~# blkid |grep dm-12
/dev/dm-12: UUID="d03dc2de-f7c1-4a88-8ca6-89108b22570c" SEC_TYPE="ext3" TYPE="ext2"
```

Débloquer le boot si fstab foireux

- A positionner au boot après que **/**, **/proc**, **/dev/pts/** et **/sys** soient positionnés.

```
#/bin/bash

### On verifie que les LVs sont bien presents
### On corrige le fstab en consequence

# Backup du fichier fstab
```

```
cp /etc/fstab /etc/fstab.orig.`date "+%d%m%Y_%Hh%Mm%Ss"`

awk '/vg/ {print $1" "$2}' /etc/fstab|grep -v swap |while read LV FS
do
if ! [[ -b $LV ]] ;then
# On ne logge pas dans /var/log vu qu'il est peut etre pas monte
echo "$LV inexistant, on corrige le fstab" >> /sbin/check_fstab.log
cat /etc/fstab |sed "s%$LV%#$LV%g" > /etc/fstab.temp
mv /etc/fstab.temp /etc/fstab
fi
done
```

Monter un partage samba au boot

Rajouter, dans le fstab :

```
//xp2400/tmp /mnt/xp2400/tmp cifs username=ben,password=PASSWORD 0 0
```

Ou avec *credentials=/home/madirectory/.credentials*

Ou faire un script de montage auto.

Troubleshooting

Restaurer des fichiers

Avec extundelete.

Impossible d'augmenter un FS (ext3)

Lors d'une extension à chaud avec *ext2online* on peut tomber sur ces messages d'erreur (notamment en RHEL 4) :

```
ext2online: ext2_ioctl: No space left on device
ext2online: unable to resize /dev/mapper/vg_vrk1-lv_testben
```

Et on trouve dans */var/log/messages* :

```
JBD: ext2online wants too many credits (2053 > 2048)
```

Ou le supprimer puis le récréer :

```
tune2fs -O ^has_journal /dev/vg_vrk1/lv_testben
tune2fs -O has_journal -J size=128 /dev/vg_vrk1/lv_testben
```

Un petit coup de *resize2fs* :

```
resize2fs /dev/vg_vrk1/lv_testben
```

On sinon remonter le FS et l'augmenter avec *ext2online*.

Conseils sur taille du journal

```
ext3 tuning
The ext3 system is a great workhorse filesystem. Lots of tools, lots of distros that know how to read it, and it's pretty much the "safe" choice for almost all workloads. Still, there are things that the default ext3 doesn't do as well as it should
so most installations need a little TLC.

For the most part, you should plan on shutting down a system before tuning it (after making backups!). Tuning doesn't take too long and is a lot simpler to do if the system is offline.

First off, you should check out your existing filesystem settings with:

# tune2fs -l /dev/hdXY

1) Directory indexing - Which helps ext3 deal with any directories that have lots of files. (See the Gentoo Forum link for explanations of why.)

# tune2fs -O dir_index /dev/hdXY
# e2fsck -D /dev/hdXY
```

```
The first command changes the ext3 system to use directory indexing for all new directories, the second command updates all existing directories.

2) Journal mode

# tune2fs -O has_journal -o journal_data /dev/hdXY

I prefer full journaled mode. The "-O has_journal" should be unnecessary (all ext3 file systems have journals after all) but probably ensures that things work if you accidently run it on a ext2 filesystem.

3) Journal size

This requires poking around a bit to find out what your current journal size is. First, you need to find the inode of the journal.

# tune2fs -l /dev/hdXY | grep -i "journal inode"
Journal inode: 8
# /sbin/debugfs /dev/md2
debugfs 1.39 (29-May-2006)
debugfs: stat <8>
Inode: 8 Type: regular Mode: 0600 Flags: 0x0 Generation: 0
User: 0 Group: 0 Size: 134217728
File ACL: 0 Directory ACL: 0
Links: 1 Blockcount: 262416
Fragment: Address: 0 Number: 0 Size: 0
ctime: 0x4658e77b -- Sat May 26 22:05:47 2007
atime: 0x00000000 -- Wed Dec 31 19:00:00 1969
mtime: 0x4658e77b -- Sat May 26 22:05:47 2007

In this particular case, for a 12GB partition, the journal size is 128MB (262416 blocks at 4096 bytes each, or look at the "Size:" field which is in bytes). On my 64GB partition, the journal size is also only 128MB.

So, do we want to muck with the journal size? Well, maybe... Doubling the size is probably okay, maybe even making it 4x larger. But beyond that and I think you'd want to tread carefully.

# tune2fs -J size=$SIZE /dev/hdXY

$SIZE is defined in megabytes, so for me to double the 128MB journal, I'd use a value of "size=256".

Source links:
Whitepaper: Red Hat's New Journaling File System: ext3 (RedHat, 2001)
EXT3 Filesystem tuning (Christoph C. Cemper, 2005)
Tuning ext3 for large disk arrays (LKML, Peter Chubb, 2005)
Some ext3 Filesystem Tips (Gentoo Forums, Peter Gordon, 2005)
Performance Tuning Guidelines for Large Deployments (Zimbra, 2007)
Linux Magazine: Tuning Journaling File Systems (2007, registration required)

Labels: ext3
posted by Thomas at 00:26
Comments:

Thanks for the useful tip. One possible correction:

I think "Blockcount: 262416" is disk blocks not filesystem blocks (i.e. 4096 is wrong and should read 512).

262416 * 512 -> 128 MiB

262416 * 4096 -> 1 GiB

I think it's there to facilitate partitioning and what-not.
posted by Anonymous Anonymous at November 23, 2007 2:40 AM

ext3 journal can't be greater than 102.400 FS-blocks.
If your FS-block is 4Kb, journal max size is 400Mb.
An ext3 journal of 128Mb is a good sized journal. On large (2T) partitions with a lot of active processes 400Mb journals make sense.

Sources : http://www.tgharold.com/techblog/2007/05/ext3-tuning.shtml
```

Erreurs : spurious interrupt

```
ata1: spurious interrupt (irq_stat 0x8 active_tag -84148995 sactive 0xffffe00)
ata1: spurious interrupt (irq_stat 0x8 active_tag -84148995 sactive 0x7fffff07)
ata1: spurious interrupt (irq_stat 0x8 active_tag -84148995 sactive 0x1e00000)
ata1: spurious interrupt (irq_stat 0x8 active_tag -84148995 sactive 0x0)
ata1: spurious interrupt (irq_stat 0x8 active_tag -84148995 sactive 0x0)
```

Parfois l'implémentation du NCQ pour certains disques SATA peut être foireuse selon sa conf (kernel, modules, etc). On peut avoir des corruptions de données. Pour désactiver le NCQ :

```
echo 1 > /sys/block/sdX/device/queue_depth
```

cf. http://linux-ata.org/faq.html#ncq

parted

On peut créer des partitions en mode non intéractif :

```
parted --script /dev/emcpowera mkpartfs primary ext2 0 20
parted --script /dev/emcpowera mkpartfs primary ext2 20 40
parted --script /dev/emcpowera mkpartfs primary ext2 40 60
parted --script /dev/emcpowera mkpart extended 60 111540
parted --script /dev/emcpowera mkpartfs logical ext2 60 80
parted --script /dev/emcpowera mkpartfs logical ext2 80 100
```

Monter automatiquement un device USB

```
apt-get install usbmount
```

Éditer le fichier */etc/usbmount/usbmount.conf*

Eviter les FSCK au boot

Soit on créé un fichier un fichier *fastboot* à la racine :

```
touch /fastboot
```

Soit on rajoute *fastboot* au niveau de grub :

```
kernel /vmlinuz-2.6.18-128.el5 ro root=/dev/rootvg/root crashkernel=128M@16M console=tty1 fastboot
```

FS busy

Si *lsof* et *fuser* ne rendent rien on peut checker avec *dmsetup info*, on récupère le nom et on lance un *dmsetup info NOM_DU_LV* et on checke le paramètre “Open count”. Si c'est différent de zéro vérifier le */etc/mtab* et faire un *mount |grep NOM_DU_LV* pour voir si le kernel n'a pas gardé un entrée. Ensuite refaire un *umount* ou *umount -f*

Monter un FS freebsd (ufs)

```
mount -t ufs -o ufstype=ufs2,ro /dev/sdb2 /mnt
```

Resizer une partition sans LVM

⇒ augmenter la taille du device sur l'ESX
⇒ rebooter si besoin pour la prise en compte

```
/dev/sdc1                1020M  943M   78M  93% /test
```

```
umount /test
fdisk /dev/sdc : p (print), d (deleter la partition sdc1), n (recréer une partoche plus grande)
xfs_repair /dev/sdc1 ou e2fsck -fy /dev/sdc1
mount /dev/sdc1 /test
xfs_growfs /dev/sdc1 ou resize2fs /dev/sdc1
```

```
/dev/sdc1                 2.0G  943M  1.1G  47% /test
```

Supprimer un répertoire si un FS a été monté dessus

```
mount --bind / /mnt
```

Supprimer fichier ouvert

```
$ /usr/sbin/lsof | grep deleted
ora    25575 data   33u   REG      65,65  4294983680   31014933 /oradata/DATAPRE/UNDOTBS009.dbf (deleted)
```

```
    $ file /proc/25575/fd/33
    /proc/25575/fd/33: broken symbolic link to `/oradata/DATAPRE/UNDOTBS009.dbf (deleted)'
    $ echo > /proc/25575/fd/33
```

From:
https://unix-bck.ndlp.info/ - **Where there is a shell, there is a way**

Permanent link:
https://unix-bck.ndlp.info/doku.php/informatique:nix:linux:linux_fs

Last update: **2017/07/20 07:59**