Table des matières

LVM : commandes de base ... 3
- ***Physical Volumes*** ... 3
  - pvscan ... 3
  - pvcreate ... 3
  - pvdisplay ... 3
- ***Volume Groups*** ... 3
  - vgcreate ... 3
  - vgextend ... 3
  - vgreduce ... 4
  - vgdisplay ... 4
  - vgscan ... 4
  - vgchange ... 4
  - vgremove ... 4
  - vgrename ... 4
- ***Logical Volumes*** ... 5
  - lvcreate ... 5
  - lvextend ... 5
  - lvreduce ... 5
  - lvdisplay ... 5
  - lvremove ... 5

Opérations courantes ... 5
- ***Créer un filesystem*** ... 6
- ***Augmenter un filesystem*** ... 6
- ***Réduire un FS*** ... 6
- ***Créer un rawdevice*** ... 6
- ***Remplacer un disque (RAID1 sur machine Compaq/HP)*** ... 7
- ***Créer un LV mirroré*** ... 7
- ***Déplacer des LVs*** ... 8
- ***Correspondances dm-* VS logical volumes*** ... 9

Metadevices (RAID 1 logiciel) ... 10

Troubleshooting ... 11
- ***EXT3-fs error (device dm-12) in start_transaction: Journal has aborted*** ... 11
- ***Retrouver un device*** ... 12
- ***VG inconsistent part I*** ... 12
- ***VG inconsistent part II*** ... 13
- ***Savoir qui écrit quoi*** ... 13
- ***VGs en double*** ... 13
- ***Déterminer le device incriminé lors de SCSI errors*** ... 15
- ***Lister hôtes SCSI*** ... 15

LVM : commandes de base

Physical Volumes

pvscan

Recherche les PVs sur tous les disques.

```
root@SpaceServer:/tmp> pvscan
pvscan -- reading all physical volumes (this may take a while...)
pvscan -- ACTIVE   PV "/dev/cciss/c0d0p2" of VG "rootvg" [67.32 GB / 49.32 GB free]
pvscan -- total: 1 [67.33 GB] / in use: 1 [67.33 GB] / in no VG: 0 [0]
```

pvcreate

Initialise le disque pour pouvoir être utilisé avec LVM. Si un PV existe déjà on peut utiliser `-ff` mais dans ce cas il faut être certain de ne pas écraser un disque utilisé ailleurs.

```
root@SpaceServer:/tmp> pvcreate /dev/emcpowerd
pvcreate -- physical volume "/dev/emcpowerd" successfully created
```

pvdisplay

Affiche les informations sur un PV (VGs attachés, etc) :

```
root@SpaceServer:/> pvdisplay /dev/emcpowerf
--- Physical volume ---
PV Name               /dev/emcpowerf
VG Name               vg_col1
PV Size               108.93 GB [228433792 secs] / NOT usable 32.19 MB [LVM: 141 KB]
PV#                   1
PV Status             available
Allocatable           yes (but full)
Cur LV                3
PE Size (KByte)       32768
Total PE              3484
Free PE               0
Allocated PE          3484
PV UUID               pmTDba-3lWV-8WEV-riwI-WkLG-odm3-1QoBpM
```

Volume Groups

vgcreate

Permet de créer un VG sur un disque initialisé avec `pvcreate`.

```
root@SpaceServer:/tmp> pvcreate /dev/emcpowerd
pvcreate -- physical volume "/dev/emcpowerd" successfully created
```

root@SpaceServer:/tmp> vgcreate vg_apps /dev/emcpowerd

```
vgcreate -- INFO: using default physical extent size 32 MB
vgcreate -- INFO: maximum logical volume size is 2 Terabyte
vgcreate -- doing automatic backup of volume group "vg_apps"
vgcreate -- volume group "vg_apps" successfully created and activated
```

vgextend

Permet d'ajouter un disque à un VG existant.

```
root@SpaceServer:/tmp> vgextend vg_apps /dev/emcpowera
vgextend -- INFO: maximum logical volume size is 2 Terabyte
vgextend -- doing automatic backup of volume group "vg_apps"
vgextend -- volume group "vg_apps" successfully extended
```

vgreduce

Permet de retirer un disque à un VG.

```
root@SpaceServer:/tmp> vgreduce vg_apps /dev/emcpowera
vgreduce -- doing automatic backup of volume group "vg_apps"
vgreduce -- volume group "vg_apps" successfully reduced by physical volume:
vgreduce -- /dev/emcpowera
```

vgdisplay

Affiche les infos sur un VG donné, notamment les disques sur lesquels il se trouve.

```
root@SpaceServer:/etc/postfix> vgdisplay -v vg_dex4
--- Volume group ---
VG Name               vg_dex4
VG Access             read/write
VG Status             available/resizable
VG #                  4
MAX LV                256
Cur LV                9
Open LV               8
MAX LV Size           2 TB
Max PV                256
Cur PV                1
Act PV                1
VG Size               54.41 GB
PE Size               32 MB
Total PE              1741
Alloc PE / Size       1193 / 37.28 GB
Free  PE / Size       548 / 17.12 GB
VG UUID               MQ2P2L-S4Qi-Ig8q-LYW1-0vJN-V4LJ-b2CXcK

 --- Logical volume ---
 truncated

--- Physical volumes ---
PV Name (#)           /dev/emcpowerc (1)
PV Status             available / allocatable
Total PE / Free PE    1741 / 548
```

vgscan

Recherche les VGs sur tous les disques et met à jour '`/etc/lvmtab`'.

```
root@SpaceServer:/tmp> vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- found active volume group "rootvg"
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of your volume group
```

vgchange

Permet de changer les attributs d'un VG. En général on utilise `vgchange -an rootvg` pour désactiver un VG (on doit dans ce cas avoir démonté les FS) et `vgchange -ay rootvg` pour activer.

vgremove

Permet de supprimer un VG (on doit d'abord le désactiver).

```
vgremove vg_apps
```

vgrename

Permet de renommer un VG sans le désactiver. Attention à bien modifer */etc/fstab*

```
vgrename vg_apps vg_appli
```

Logical Volumes

lvcreate

Permet de créer un LV sur un VG donné.

```
root@SpaceServer:/tmp> lvcreate -L 2G -n lv_test vg_apps
lvcreate -- doing automatic backup of "vg_apps"
lvcreate -- logical volume "/dev/vg_apps/lv_test" successfully created
```

lvextend

Permet d'augmenter la taille du LV. Umount d'abord du FS

```
root@SpaceServer:/tmp> lvextend -L +1G /dev/vg_apps/lv_test
lvextend -- extending logical volume "/dev/vg_apps/lv_test" to 3 GB
lvextend -- doing automatic backup of volume group "vg_apps"
lvextend -- logical volume "/dev/vg_apps/lv_test" successfully extended
```

lvreduce

Permet de réduire la taille du LV. Attention on réduit d'abord le FS et seulement ensuite on peut réduire le LV.

```
root@SpaceServer:/tmp> lvreduce -L -1G /dev/vg_apps/lv_test
lvreduce -- WARNING: reducing active logical volume to 2 GB
lvreduce -- THIS MAY DESTROY YOUR DATA (filesystem etc.)
lvreduce -- do you really want to reduce "/dev/vg_apps/lv_test"? [y/n]: y
lvreduce -- doing automatic backup of volume group "vg_apps"
lvreduce -- logical volume "/dev/vg_apps/lv_test" successfully reduced
```

lvdisplay

Affiche les différentes infos sur le LV. Utilisez -v pour avoir plus d'infos.

```
root@SpaceServer:/tmp> lvdisplay /dev/vg_apps/lv_test
--- Logical volume ---
LV Name                /dev/vg_apps/lv_test
VG Name                vg_apps
LV Write Access        read/write
LV Status              available
LV #                   1
# open                 0
LV Size                2 GB
Current LE             64
Allocated LE           64
Allocation             next free
Read ahead sectors     1024
Block device           58:9
```

lvremove

Permet de supprimer un LV.

```
root@SpaceServer:/tmp> lvremove /dev/vg_apps/lv_test
lvremove -- do you really want to remove "/dev/vg_apps/lv_test"? [y/n]: y
lvremove -- doing automatic backup of volume group "vg_apps"
lvremove -- logical volume "/dev/vg_apps/lv_test" successfully removed
```

Opérations courantes

note : on travaille ici en *reiserfs*. Pour l'*ext3* utiliser **mkfs.ext3** au lieu de **mkreiserfs**.

Créer un filesystem

On créé d'abord le volume logique (LV), puis le filesystem (FS) à proprement parler. On finit par modifier le */etc/fstab*

Création du LV :

```
lvcreate -L taille[M|G] -n nom_du_lv nom_du_vg
lvcreate -L 500M -n lv_apache vgdata
```

Création du FS (ici au format *reiserfs*):

```
mkreiserfs /dev/nom_du_vg/nom_du_lv
mkreiserfs /dev/vgdata/lv_apache
```

Augmenter un filesystem

Il est possible d'augmenter à chaud sans avoir à démonter le FS. Dans ce cas on augmente d'abord le LV `puis` ensuite on augmente le FS.

Augmentation du LV :

```
lvextend -L +taille[M|G] /dev/nom_du_vg/nom_du_lv
lvextend -L +500M /dev/vgdata/lv_apache
```

Augmentation du FS :

```
resize_reiserfs -s+taille[M|G] /dev/nom_du_vg/nom_du_lv
reiserfs -s+500M /dev/vgdata/lv_apache
```

Réduire un FS

Il est obligatoire de démonter le FS pour la réduction. On réduit d'abord le FS `puis` le LV.

Démontage du FS :

```
umount /nom_du_fs
umount /apache
```

Réduction du FS :

```
resize_reiserfs -s-taille[M|G] /dev/nom_du_vg/nom_du_lv
resize_reiserfs -s-500M /dev/vgdata/lv_apache
```

Réduction du LV :

```
lvreduce -L -taille[M|G] /dev/nom_du_vg/nom_du_lv
lvreduce -L 500M /dev/vgdata/lv_apache
```

Remontage du FS :

```
mount /nom_du_fs
mount /apache
```

Créer un rawdevice

On créé d'abord un LV sur lequel on positionnera un rawdevice :

```
lvcreate -L 16G -n lv_raw_01 vg_data
```

Lancer la commande *raw* pour binder les rawdevices :

```
raw /dev/raw/rawX /dev/vg_data/lv_raw_01
```

Renseigner le fichier */etc/sysconfig/rawdevices*

Configurer le démarrage des raws au boot

```
/etc/init.d/rawdevices start
```

Remplacer un disque (RAID1 sur machine Compaq/HP)

- Vérifier l'existant:

```
/usr/sbin/hpacucli
```

→ *controller all show*

```
Smart Array 6i in Slot 0      ()
```

→ *controller slot=0 physicaldrive all show*

```
Smart Array 6i in Slot 0

  array A
   physicaldrive 2:0   (port 2:id 0 , Parallel SCSI, 72.8 GB, OK)
   physicaldrive 2:1   (port 2:id 1 , Parallel SCSI, 72.8 GB, OK)

  array B
   physicaldrive 2:2   (port 2:id 2 , Parallel SCSI, 72.8 GB, OK)
   physicaldrive 2:3   (port 2:id 3 , Parallel SCSI, 72.8 GB, OK)
```

→ *controller slot=0 logicaldrive all show*

```
Smart Array 6i in Slot 0

  array A
   logicaldrive 1 (67.8 GB, 1+0, OK)

  array B
   logicaldrive 2 (67.8 GB, 1+0, OK)
```

- Créer un nouveau RAID:

```
/usr/sbin/hpacucli
```

→ *controller slot=0 create type=logicaldrive drives=allunassigned raid=1+0*

Puis *pvcreate* sur le nouveau disque.

- Etendre un RAID existant:

```
/usr/sbin/hpacucli
```

→ *controller slot=0 array B modify size=max*

Reboot du serveur pour prise en compte au niveau système puis *pvresize* pour étendre le LVM.

Créer un LV mirroré

```
lvcreate --type mirror -L 128MB -m 1 --mirrorlog mirrored -n vol1 testvg
```

Soit le vg suivant **vg_mirror** :

```
Volume groupe : vg_mirror

Volume(s) physique(s) : 2
PE : totaux = 223072 Mo, alloues : 0 Mo, libres : 223072 Mo
PV : /dev/emcpowerm         ,tot_sz = 111536 Mo   ,lib_sz  = 111536 Mo
PV : /dev/emcpowern         ,tot_sz = 111536 Mo   ,lib_sz  = 111536 Mo
```

Par défaut le lvm a besoin de 3 disques : 2 pour mirrorer les datas et un 3ème pour la log mais du coup si on perd ce disque on perd les datas ... Autant le mettre en RAM (il sera recréé à chaque reboot par exemple).

```
root@serverl1101561:/# lvcreate -m 1 --corelog -L 20G -n lv_one vg_mirror
  Logical volume "lv_one" created

Volume groupe : vg_mirror

Volume(s) physique(s) : 2
PE : totaux = 223072 Mo, alloues : 40960 Mo, libres : 182112 Mo
PV : /dev/emcpowerm         ,tot_sz = 111536 Mo   ,lib_sz  = 91056 Mo
PV : /dev/emcpowern         ,tot_sz = 111536 Mo   ,lib_sz  = 91056 Mo
```

```
Volume(s) logique(s)  : 3
LV : lv_one                ,log_sz = 20480 Mo,   sur   ne_mimage_0 ne_mimage_1
LV : lv_one_mimage_0       ,log_sz = 20480 Mo,   sur
LV : lv_one_mimage_1       ,log_sz = 20480 Mo,   sur

root@server1101561:/# mkfs.ext3  /dev/vg_mirror/lv_one
root@server1101561:/# mount /dev/vg_mirror/lv_one /mnt
```

Malheureusement on ne peut pas étendre le LV à chaud, il faut démonter le FS puis faire un *lvchange -an* ce qui n'est pas très pratique.

```
root@server1101561:/# lvextend -L +10G /dev/vg_mirror/lv_one
  Extending 2 mirror images.
  Mirrors cannot be resized while active yet.
```

Une solution existe néanmoins. On supprime une patte du mirroir, on étend et on réintègre la patte en question :

```
root@serverl1101561:/# lvconvert -m 0 /dev/vg_mirror/lv_one
  Logical volume lv_one converted.

root@server1101561:/# lvextend -L +10G /dev/vg_mirror/lv_one
  Extending logical volume lv_one to 30.00 GB
  Logical volume lv_one successfully resized

root@server1101561:/# lvconvert -m 1 --corelog /dev/vg_mirror/lv_one
  Logical volume lv_one converted.
```

On peut voir le statut de la resynchro :

```
root@server1101561:/var/log# lvs -a /dev/vg_mirror/lv_one
  LV     VG        Attr   LSize  Origin Snap%  Move Log Copy%
  lv_one vg_mirror mwi-ao 50.00G                         9.03
```

On vérifie le résultat :

```
Volume groupe : vg_mirror

Volume(s) physique(s) : 2
PE : totaux = 223072 Mo, alloues : 61440 Mo, libres : 161632 Mo
PV : /dev/emcpowerm         ,tot_sz = 111536 Mo   ,lib_sz  = 80816 Mo
PV : /dev/emcpowern         ,tot_sz = 111536 Mo   ,lib_sz  = 80816 Mo

Volume(s) logique(s)  : 3
LV : lv_one                ,log_sz = 30720 Mo,   sur   ne_mimage_0 ne_mimage_1
LV : lv_one_mimage_0       ,log_sz = 30720 Mo,   sur
LV : lv_one_mimage_1       ,log_sz = 30720 Mo,   sur
```

On étend le FS :

```
root@server1101561:/# ext2online /mnt/
ext2online v1.1.18 - 2001/03/18 for EXT2FS 0.5b
```

Déplacer des LVs

Cas 1

⇒ On veut déplacer tous les LVs du disque **emcpowerm** vers **emcpowern**, à chaud.

- Soit le *vg_appli* constitué du disque **emcpowerm** :

```
root@server1101561:~# vgdisplay -v vg_appli 2>/dev/null|grep "PV Name"
  PV Name               /dev/emcpowerm
```

- On rajoute le disque **emcpowern** au *vg_appli* :

```
root@server1101561:~# pvcreate /dev/emcpowern
  Physical volume "/dev/emcpowern" successfully created

root@server1101561:~# vgextend vg_appli /dev/emcpowern
  Volume group "vg_appli" successfully extended
```

- Soient les LVs suivants :

```
root@server1101561:~# ls -1 /dev/vg_appli
```

```
lv_data1
lv_data2
lv_data3
lv_data4
lv_data5
```

- On utilise *pvmove* pour déplacer ces LVs, à chaud :

```
root@server1101561:~# ls -1 /dev/vg_appli|while read i
> do
> pvmove -n $i /dev/emcpowerm /dev/emcpowern
> echo "$i moved \!"
> done
  /dev/emcpowerm: Moved: 2.0%
  /dev/emcpowerm: Moved: 4.2%
  /dev/emcpowerm: Moved: 6.5%
  /dev/emcpowerm: Moved: 8.8%
  /dev/emcpowerm: Moved: 11.0%
...
```

- On supprime le disque **emcpowerm** du *vg_appli* :

```
vgreduce vg_appli /dev/emcpowerm
vgscan
vgcfgbackup vg_appli
```

Cas 2

⇒ On veut déplacer certains LVs du disque **emcpowerm** vers **emcpowern** avec le moins d'indispo possible et faire un nouveau VG *vg_appli2* avec les LVs déplacés.

On reprend la même procédure que précédemment. Une fois tous les LVs déplacés intégralement sur le nouveau disque il faut désactiver le VG (et donc avoir démonter tous les FS au préalable et désactiver les éventuels raws) :

```
root@server1101561:~# vgchange -an vg_appli
  0 logical volume(s) in volume group "vg_appli" now active

root@server1101561:~# vgsplit vg_appli vg_appli2 /dev/emcpowern
  Volume group "vg_appli2" successfully split from "vg_appli"
```

Et voila :

```
root@server1101561:~# vgscan
  Reading all physical volumes.  This may take a while...
  Found volume group "vg_appli2" using metadata type lvm2
  Found volume group "vg_appli" using metadata type lvm2
  Found volume group "rootvg" using metadata type lvm2
```

Ensuite on active les VGs avec *vgchange -ay* et on remonte les LVs.

Correspondances dm-* VS logical volumes

```
root@server2311827:/# lvmdiskscan  |grep dm- |tail
  /dev/dm-23     [        1.00 GB]
  /dev/dm-24     [        3.44 GB]
  /dev/dm-25     [        1.47 GB]
  /dev/dm-26     [        1.47 GB]
  /dev/dm-27     [      160.00 MB]
  /dev/dm-28     [      224.00 MB]
  /dev/dm-29     [        5.00 GB]
  /dev/dm-30     [        5.00 GB]
  /dev/dm-31     [      288.00 MB]
  /dev/dm-32     [       39.06 GB]
```

On peut les confondre avec des devices multipath, pour être sur on peut lancer un *multipath -ll* et effectuer la correspondance. Si la commande n'existe pas ou ne rend rien c'est tout simple que linux créé un device dm-* pour chaque LV créé. Pour checker :

- On récupère les minor/major :

```
root@server2311827:/#  ls -l /dev/dm-27
brw-r-----  1 root root 253, 27 Apr 17 09:39 /dev/dm-27
```

- On check dans les VGs :

```
root@server2311827:/# vgdisplay -v 2>/dev/null |grep 253:27 -B 11|grep "LV Name"
```

```
  LV Name                /dev/rootvg/lv_wls103_d1
```

Metadevices (RAID 1 logiciel)

- Créer les metadevices :

```
mdadm --create /dev/md0 -l 1 --raid-devices=2 /dev/emcpowera2 /dev/emcpowerc2
mdadm --create /dev/md1 -l 1 --raid-devices=2 /dev/emcpowerb /dev/emcpowerd
```

- Créer un RAID 1 avec un seul disque au début :

```
mdadm --create /dev/md0 -l 1 --raid-devices=2 /dev/sda2 missing
```

- Démarrer les metadevices :

```
mdadm --assemble /dev/md0 /dev/emcpowera2 /dev/emcpowerc2
mdadm --assemble /dev/md1 /dev/emcpowerb /dev/emcpowerd
```

- Arrêter les metadevices :

```
mdadm --stop /dev/md0
mdadm --stop /dev/md1
```

- Dans le fichier */etc/mdadm.conf* rajouter :

```
DEVICE /dev/emcpowerb /dev/emcpowerd
DEVICE /dev/emcpowera2 /dev/emcpowerc2
```

- Puis pour vérifier :

```
mdadm --examine --scan -c /etc/mdadm.conf

root@SpaceServer:/root> mdadm --examine --scan -c /etc/mdadm.conf
ARRAY /dev/md0 level=raid1 num-devices=2 UUID=cdeabe60:356153ce:16bed617:874f97db
   devices=/dev/emcpowera2,/dev/emcpowerc2
ARRAY /dev/md1 level=raid1 num-devices=2 UUID=66458bd1:5bb9acd4:18503815:b36812b6
   devices=/dev/emcpowerb,/dev/emcpowerd
```

```
root@spaceServer:/mnt> mdadm --detail /dev/md0
/dev/md0:
        Version : 00.90.00
  Creation Time : Tue Apr 17 16:18:46 2007
     Raid Level : raid1
     Array Size : 57108416 (54.46 GiB 58.48 GB)
    Device Size : 57108416 (54.46 GiB 58.48 GB)
   Raid Devices : 2
  Total Devices : 2
Preferred Minor : 0
    Persistence : Superblock is persistent

    Update Time : Thu Apr 19 12:20:23 2007
          State : dirty, no-errors ----------------------------> le dirty c'est "normal" :-]
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0
```

- Ensuite si tout est ok on peut écrire dans le fichier :

```
mdadm --examine --scan -c /etc/mdadm.conf >> /etc/mdadm.conf
```

- Supprimer un device d'un RAID :

⇒ on le passe en *failed*

```
root@SpaceServer:/mnt>  mdadm /dev/md0 --fail /dev/emcpowerb
mdadm: set /dev/emcpowerb faulty in /dev/md0
```

⇒ on le supprime

```
root@SpaceServer:/mnt>  mdadm /dev/md0 --remove /dev/emcpowerb
mdadm: hot removed /dev/emcpowerb

root@SpaceServer:/mnt> mdadm --detail /dev/md0
/dev/md0:
```

```
    Number   Major   Minor   RaidDevice State
       0       0       0        0      faulty removed
       1     232       48       1      active sync   /dev/emcpowerd
```

- Ensuite on peut rajouter un nouveau device

```
root@SpaceServer:/mnt/ben> mdadm /dev/md0 --add /dev/emcpowerb
mdadm: hot added /dev/emcpowerb
```

Hop ! Synchro en cours :

```
root@SpaceServer:/mnt/ben> cat /proc/mdstat
Personalities : [raid1]
read_ahead 1024 sectors
Event: 11
md1 : active raid1 emcpowera2[0] emcpowerc2[1]
      56902144 blocks [2/2] [UU]

md0 : active raid1 emcpowerb[2] emcpowerd[1]
      57108416 blocks [2/1] [_U]
      [>....................]  recovery =  0.2% (150040/57108416) finish=88.5min speed=10717K/sec
unused devices: <none>
```

Par défaut la vitesse de reconstruction est limitée pour des soucis de perfs, on peut le voir dans la log :

```
md: syncing RAID array md0
md: minimum _guaranteed_ reconstruction speed: 1000 KB/sec/disc.
md: using maximum available idle IO bandwith (but not more than 200000 KB/sec) for reconstruction.
md: using 128k window, over a total of 14277056 blocks.
```

On peut augmenter la vitesse via */proc/sys/dev/raid/speed_limit_min* :

```
echo 25000 >> /proc/sys/dev/raid/speed_limit_min
```

On voit tout de suite la différence :

Avant :

```
root@server9002737:/mnt/ben# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 emcpowerd[1] emcpowere[0]
      14277056 blocks [2/2] [UU]
      [======>.............]  resync = 33.7% (4821888/14277056) finish=151.5min speed=1036K/sec
```

Après:

```
root@server9002737:/mnt/ben# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 emcpowerd[1] emcpowere[0]
      14277056 blocks [2/2] [UU]
      [================>....]  resync = 82.6% (11803776/14277056) finish=1.6min speed=25004K/sec
```

- On peut aussi retailler un métadevice (augmenter ou réduire) :

```
mdadm --grow size=50G /dev/md0
pvresize /dev/md0
```

Troubleshooting

EXT3-fs error (device dm-12) in start_transaction: Journal has aborted

On récupère les infos *minor* / *major* du device :

```
[root@SomeMachine]# ls -l /dev/dm-12
brw-r-----  1 root root 253, 12 Jun 24 11:27 /dev/dm-12
```

On peut également trouver les infos sous */dev/mapper* et utiliser */proc/partitions*.

On cherche le LV correspondant :

```
[root@SomeMachine]# lvdisplay -v |grep -B 13 "253:12"
File descriptor 3 left open
```

```
   Finding all logical volumes

  --- Logical volume ---
  LV Name                /dev/vg_oraORACLESID/lv_oracle
  VG Name                vg_oraORACLESID
  LV UUID                6Re4id-CaYP-OcML-z3J7-g4lG-AEtu-PFXLtO
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                512.00 MB
  Current LE             128
  Segments               1
  Allocation             inherit
  Read ahead sectors     0
  Block device           253:12
```

On en déduit le FS grâce au */etc/fstab* ou au fichier de démarrage MC Service Guard (comme ici) :

```
[root@SomeMachine]# grep "vg_oraORACLE_SID/lv_oracle" cl_*/*.sh
LV[0]="/dev/vg_oraORACLE_SID/lv_oracle"; FS[0]="/apps/oracle"; FS_TYPE[0]="ext3"; FS_MOUNT_OPT[0]=""
```

Retrouver un device

Soit l'erreur suivante dans */var/log/messages* :

```
Feb 19 09:28:44 SomeMachine kernel: 3a:0c: rw=0, want=652360104, limit=167772160
```

Pas forcément très clair ... Ce qui nous intéresse ici c'est **3a:0c**, on convertit de l'hexa vers le décimal :

Hexa	Décimal
3a	58
0c	12

Il s'agit du device LVM **(58,12)** :

```
root@SomeMachine:/tmp> lvscan |grep ACTIVE|awk '{print $4}'|xargs lvdisplay|grep 58:12 -B 13

--- Logical volume ---
LV Name                /dev/vg_col1/lv_sybasedata1
VG Name                vg_col1
LV Write Access        read/write
LV Status              available
LV #                   3
# open                 1
LV Size                160 GB
Current LE             5120
Allocated LE           5120
Allocation             next free
Read ahead sectors     1024
Block device           58:12
```

Ensuite on retrouve le FS associé via le fichier */etc/fstab* ou le fichier de démarrage cluster (MC Service Guard par exemple).

VG inconsistent part I

On peut parfois avoir cette erreur si un disque a été retiré à l'arrache :

```
Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
Couldn't find all physical volumes for volume group vg_oap.
```

Pour y remédier :

```
vgreduce --removemissing --test vg_oap

vgreduce --removemissing vg_oap
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find all physical volumes for volume group vg_oap.
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find all physical volumes for volume group vg_oap.
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find all physical volumes for volume group vg_oap.
```

```
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find all physical volumes for volume group vg_oap.
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Couldn't find device with uuid 'sAK55E-35qf-Ffs5-ju6g-ZCnN-ojwi-BPb1CJ'.
  Wrote out consistent volume group vg_oap
```

Puis un petit *vgscan* pour vérifer.

VG inconsistent part II

Parfois on peut avoir le message d'erreur ci-dessous :

```
vgscan -- ERROR "vg_read_with_pv_and_lv(): allocated LE of LV" can't get data of volume group from physical volume(s)
```

Pour y remédier on peut tenter la commande *vgcfgrestore* et utiliser un backup précédent. Un backup est généré :

- dans */etc/lvmconf* sous RHEL3
- dans */etc/lvm/backup* sous RHEL4 / Debian and co

On vérifie le VG :

```
root@server1106215:/etc/lvmconf> vgscan
vgscan -- reading all physical volumes (this may take a while...)
vgscan -- found active volume group "rootvg"
vgscan -- found inactive volume group "vg_toto"
vgscan -- only found 0 of 160 LEs for LV /dev/vg_titi/lv_srec (0)
vgscan -- ERROR "vg_read_with_pv_and_lv(): allocated LE of LV" can't get data of volume group "vg_titi" from physical volume(s)
vgscan -- "/etc/lvmtab" and "/etc/lvmtab.d" successfully created
vgscan -- WARNING: This program does not do a VGDA backup of your volume groups
```

On restaure le conf du VG :

```
root@server1106215:/etc/lvmconf> vgcfgrestore -f vg_titi.conf -n vg_titi "/dev/emcpowerd"
vgcfgrestore -- size of physical volume /dev/emcpowerd differs from backup
```

Pour ignorer les contraintes de taille (de toute façon au point où on en est) :

```
root@server1106215:/etc/lvmconf> vgcfgrestore -i -f vg_titi.conf -n vg_titi "/dev/emcpowerd"
vgcfgrestore -- forcing write of VGDA of "vg_titi" to physical volume "/dev/emcpowerd"
vgcfgrestore -- ignoring size mismatches
vgcfgrestore -- VGDA for "vg_titi" successfully restored to physical volume "/dev/emcpowerd"
vgcfgrestore -- you may not have an actual backup of restored volume group "vg_titi"
```

Juste pour se proteger, on fait un vgcfgbackup sur les 2 noeuds.

```
root@server1106216:/etc/lvmconf> ls -l /etc/lvmconf/vg_titi.conf
-rw-r-----    1 root     root       166924 Feb  1 11:15 /etc/lvmconf/vg_titi.conf
root@server1106215:/etc/lvmconf> ls -l *conf
-rw-r-----    1 root     root       166924 Feb  1 11:14 vg_titi.conf
```


Parfois il faut lancer le *vgcfgrestore* avec un autre disque du VG jusqu'à ce que ça passe (dans le cas où le VG est sur plusieurs disques), la commande *pvscan* permet de lister les PVs.

Savoir qui écrit quoi

- Désactiver temporairement l'écriture du kernel logger dans kern.log (*/etc/syslogd.conf*).

```
echo 1 > /proc/sys/vm/block_dump
while true; do dmesg -c; sleep 1; done
echo 0 > /proc/sys/vm/block_dump
```

VGs en double

Lorsque des disques SAN sont rajoutés sur une machine Linux alors qu'ils n'ont pas été formatés on peut avoir des erreurs de Duplicate VG name notamment avec le vg_apps. Cela provient du fait que les disques ajoutés non pas été formatés et qu'ils contenaient dejà un vg_apps. Pour que ce soit encore plus rock n' roll le vg_apps importé n'a pas tous ses disques. Par exemple :

→ le vg_apps déjà existant et complet :

```
  --- Volume group ---
  VG Name               vg_apps
  VG UUID               sz0iLr-t1jV-iaJf-8xLM-1clh-FK2L-21Zeyk

  --- Physical volumes ---
  PV Name               /dev/dm-21
  PV UUID               llalr9-8I9F-y3st-mcCa-QkZ7-QDX0-ZAHI3O
```

→ le vg_apps incomplet et importé par la rajout du disque SAN :

```
  --- Volume group ---
  VG Name               vg_apps
  VG UUID               Ca5e4x-xbqq-6j5V-x9vG-Dvdb-xixF-Lq2XUa

  --- Physical volumes ---
  PV Name               unknown device
  PV UUID               jAnT2n-BoUf-Cazf-1SrY-sVx4-3dzn-J83scd
  PV Status             allocatable
  Total PE / Free PE    13942 / 0

  PV Name               /dev/dm-18
  PV UUID               14VTx7-IZYM-yejh-zCf5-lkHM-qDg7-DGLl0c
  PV Status             allocatable
  Total PE / Free PE    27884 / 577

  PV Name               unknown device
  PV UUID               JUOqyk-t6Ob-ciL6-IB4X-k23y-fQNv-KQLd2v
  PV Status             allocatable
  Total PE / Free PE    13942 / 3702
```

On remarque les unknown device (chaque disque du VG contient toutes les infos du VG dans son entête). Par ailleurs chaque fois qu'un commande LVM est lancé la machine vomit des erreurs et on ne peut plus bosser sur le vg_apps car il est vu en double par l'OS.

Pour vérifier que c'est bien le /dev/-dm18 qui pose problème on peut visualiser les LVs présents :

```
pvdisplay -m /dev/dm-18
```

On vérifie qu'aucun FS rétourné par la commande n'est monté.

Pour résoudre le problème :

```
vgrename Ca5e4x-xbqq-6j5V-x9vG-Dvdb-xixF-Lq2XUa vg_apps_KO => on renomme le VG avec son VG UUID
vgchange -a n vg_apps_KO                  => on désactive le VG
vgremove vg_apps_KO                    => on bute le VG
pvremove -ff /dev/dm-18                   => on bute l'entête LVM du device
```

Sauf que dans le cas que j'ai rencontré je n'ai pas pu renommer le VG car il était en cours d'utilisation … En fait lors du vgscan un LV du VG foireux devait avoir les mêmes minor/major et a été importé dans le bon VG :

```
root@parsl2414967:/tmp# ls -l /dev/vg_apps/lv_dba
lrwxrwxrwx 1 root root 26 Nov 17 18:08 /dev/vg_apps/lv_dba -> /dev/mapper/vg_apps-lv_dba
```

L'OS le voyait actif car le bon vg_apps était actif et donc impossible de faire quoique ce soit sur le VG corrompu. La commande lvremove ne passait donc pas (une sorte de LV fantôme). Pour le supprimer réellement il faut utiliser la commande de bas niveau dmsetup qui permet d'accéder aux devices multipath et au LVM. On le supprime :

```
dmsetup info -c |grep lv_dba    => on récupère le nom du LV au format dmsetup
dmsetup remove vg_apps-lv_dba   => on le supprime réellemment
```

Ensuite on peut reprendre la manip ci-dessus en renommant le VG. Si jamais ça ne passe pas on peut aussi supprimer le device /dev/dm-18 :

```
mpath8 (360060480000290103021533030353143) dm-18 EMC,SYMMETRIX
[size=109G][features=0][hwhandler=0][rw]
\_ round-robin 0 [prio=2][active]
 \_ 3:0:0:45 sdj 8:144 [active][ready]
 \_ 4:0:0:45 sds 65:32 [active][ready]

dmsetup remove mpath8           => on supprime le device
pvremove /dev/sdj /dev/sds      => on vire les infos LVM sur les 2 chemins
```

Un vgscan permet de remettre tout d'aplomb. Le device /dev/dm-18 est maintenant vierge de toutes infos LVM et on peut enfin bosser.

Cette méthode permet de ne pas rebooter le serveur ni d'arrêter les applis qui tournent.

Déterminer le device incriminé lors de SCSI errors

```
May 31 01:00:04 server3006361 kernel: scsi3 (0:0): rejecting I/O to offline device
May 31 01:00:04 server3006361 kernel: SCSI error: host 3 id 0 lun 0 return code = 4000000
May 31 01:00:04 server3006361 kernel:    Sense class 0, sense error 0, extended sense 0
May 31 01:00:05 server3006361 su(pam_unix)[2593]: session opened for user root by (uid=0)
May 31 01:00:06 server3006361 kernel: scsi3 (0:0): rejecting I/O to offline device
May 31 01:00:06 server3006361 kernel: SCSI error: host 3 id 0 lun 0 return code = 4000000
May 31 01:00:06 server3006361 kernel:    Sense class 0, sense error 0, extended sense 0
May 31 01:00:06 server3006361 kernel: scsi3 (0:0): rejecting I/O to offline device
May 31 01:00:06 server3006361 kernel: SCSI error: host 3 id 0 lun 0 return code = 4000000
May 31 01:00:06 server3006361 kernel:    Sense class 0, sense error 0, extended sense 0
May 31 01:00:06 server3006361 kernel: scsi3 (0:0): rejecting I/O to offline device
May 31 01:00:06 server3006361 kernel: SCSI error: host 3 id 0 lun 0 return code = 4000000
May 31 01:00:06 server3006361 kernel:    Sense class 0, sense error 0, extended sense 0
```

```
root@server3006361:PRODUCTION:/var/log# cat /proc/scsi/scsi |egrep -A 2 "scsi3"
Host: scsi3 Channel: 00 Id: 00 Lun: 00
  Vendor: Dell     Model: Virtual  CDROM   Rev: 123
  Type:   CD-ROM                           ANSI SCSI revision: 02
```

Lister hôtes SCSI

```
[root@localhost ~]# lsscsi --hosts
[0]    ata_piix
[1]    ata_piix
[2]    ahci
```

```
[root@localhost ~]# lsblk -S
NAME HCTL       TYPE VENDOR   MODEL             REV TRAN
sda  2:0:0:0    disk ATA      VBOX HARDDISK    1.0  sata
sr0  1:0:0:0    rom  VBOX     CD-ROM           1.0  ata
```

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

Permanent link:
https://unix.ndlp.info/doku.php/informatique:nix:linux:linux_lvm

Last update: **2018/01/11 07:38**