

## Práctica RAID 0 stripped

### 1. Una partició a cada disc físic (fdisk, gpart). Mínim dos discos.

Primerament, creem dos discos físics desde VMware per dur a terme la pràctica, en total tindrem 3 discos funcionant en la màquina virtual. Utilitzant la comanda `lvmdiskscan`, hem pogut obtenir els dispositius de bloc (`sdb` i `sdc`) i fent ús de `fdisk /dev/sdb` i `/dev/sdc` hem creat una partició de la següent manera: (a cada disc físic)

```
Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-41943039, default 2048):
Last sector, +sectors or +size[K,M,G,T,P] (2048-41943039, default 41943039):
Created a new partition 1 of type 'Linux' and of size 20 GiB.
```

Si tornem a executar `lvmdiskscan` o mirem el directori `/dev`, podem veure que s'han creat unes noves particions `sdb1` i `sdc1`:

```
alumne@pcrecanvib5:~$ sudo lvmdiskscan
/dev/sda1 [      18,00 GiB]
/dev/sda2 [       <2,00 GiB]
/dev/sdb1 [      <20,00 GiB]
/dev/sdc1 [      <20,00 GiB]
0 disks
4 partitions
0 LVM physical volume whole disks
0 LVM physical volumes
```

### 2. Un volum físic (Physical Volume) a cada partició (pvcreate).

La forma de lliurar un dispositiu al LVM és marcar-lo com un Physical Volum (PV). Per dur a terme aquesta part, simplement cal usar la comanda `pvcreate` a les dues particions creades al pas anterior:

```
alumne@pcrecanvib5:~$ sudo pvcreate /dev/sdb1
[sudo] password for alumne:
Physical volume "/dev/sdb1" successfully created.
alumne@pcrecanvib5:~$ sudo pvcreate /dev/sdc1
Physical volume "/dev/sdc1" successfully created.
```

### 3. Un grup de volums (Volume Group) que agruparà als volums físics de varis discs en un sol disc.

VG és una espècie de disc dur virtual el qual la seva grandària ve donada per la suma de l'espai dels PVs que el componen. Per crear el VG i agrupar en aquest els dos PVs creats anteriorment, sols ha fet falta utilitzar la comanda `vgcreate`. El grup l'hem anomenat `vgcaso` com diu l'enunciat de la pràctica.

```
alumne@pcrecanvib5:~$ sudo vgcreate vgcaso /dev/sdb1 /dev/sdc1
Volume group "vgcaso" successfully created
```

Si utilitzem la comanda `pvs` o `vgs`, es pot veure com s'han agrupat els PVs al VG correctament:

```
alumne@pcrecanvib5:~$ sudo pvs
PV          VG      Fmt  Attr  PSize   PFree
/dev/sdb1   vgcaso  lvm2  a--   <20,00g <20,00g
/dev/sdc1   vgcaso  lvm2  a--   <20,00g <20,00g
```

### 4. Un volum lògic (Logical Volume). És un dispositiu de blocs virtual, creat a sobre d'un VG (`lvcreate`).

Un cop tenim un VG ja podem crear els dispositius que realment farem servir. Els volums lògics (LV) pertanyen a un VG, de què prenen el seu espai. Es poden crear, esborrar-se i créixer sense necessitat de reiniciar la màquina o aturar serveis.

Per crear el LV, s'ha utilitzat la comanda `lvcreate` indicant-li el VG a què pertany, la mida i i optionalment, el nom que volem donar-li al LV. S'han creat dos LVs tots dos de tamany 2GB, un anomenat `lvol_banda` (stripped) i l'altre anomenat `lvol_seq` (lineal) com indica l'enunciat.

`lvol_banda`:

```
alumne@pcrecanvib5:~$ sudo lvcreate -L 2G -i 2 -I 128k -n lvol_banda vgcaso
Logical volume "lvol_banda" created.
```

`lvol_seq`:

```
alumne@pcrecanvib5:~$ sudo lvcreate -L 2G -n lvol_seq vgcaso
Logical volume "lvol_seq" created.
```

Per comprovar que tot s'ha creat correctament, hem utilitzat la comanda `lvdisplay -m`:

```
alumni@pcrecanvib5:~$ sudo lvdisplay -m
--- Logical volume ---
LV Path                /dev/vgcaso/lvol_banda
LV Name                lvol_banda
VG Name                vgcaso
LV UUID                ehyslV-WdmS-RjWV-cNio-wL4I-fzz5-bBk0DC
LV Write Access        read/write
LV Creation host, time pcrecanvib5, 2020-04-25 12:23:25 +0200
LV Status              available
# open                 0
LV Size                2,00 GiB
Current LE             512
Segments              1
Allocation             inherit
Read ahead sectors    auto
 - currently set to   1024
Block device           253:0

--- Segments ---
Logical extents 0 to 511:
Type                striped
Stripes             2
Stripe size         128,00 KiB
Stripe 0:
  Physical volume    /dev/sdb1
  Physical extents   0 to 255
Stripe 1:
  Physical volume    /dev/sdc1
  Physical extents   0 to 255
```

```
--- Logical volume ---
LV Path                /dev/vgcaso/lvol_seq
LV Name                lvol_seq
VG Name                vgcaso
LV UUID                nkviEb-G0q0-KbRc-gZW6-wD5K-iCuC-VivODP
LV Write Access        read/write
LV Creation host, time pcrecanvib5, 2020-04-25 12:24:56 +0200
LV Status              available
# open                 0
LV Size                2,00 GiB
Current LE             512
Segments              1
Allocation             inherit
Read ahead sectors    auto
 - currently set to   256
Block device           253:1

--- Segments ---
Logical extents 0 to 511:
Type                linear
Physical volume      /dev/sdb1
Physical extents     256 to 767
```

També, hem avaluat el rendiment de lectura dels dos LVs creats; striped i lineal, amb la comanda `hdparm`. Els resultats han estat els següents:

```
alumni@pcrecanvib5:~$ sudo hdparm -tT /dev/vgcaso/lvol_banda
/dev/vgcaso/lvol_banda:
Timing cached reads:   21248 MB in  1.99 seconds = 10651.60 MB/sec
Timing buffered disk reads: 2048 MB in  1.47 seconds = 1396.03 MB/sec
alumni@pcrecanvib5:~$ sudo hdparm -tT /dev/vgcaso/lvol_seq
/dev/vgcaso/lvol_seq:
Timing cached reads:   18870 MB in  2.00 seconds = 9455.97 MB/sec
Timing buffered disk reads: 2048 MB in  1.18 seconds = 1742.05 MB/sec
```

## 5. Un sistema de fitxers pel volum lògic (mkfs)

Per acabar, hem creat per cadascun dels LV un sistema de fitxer diferent; ext3 per `lvol_banda` i xfs per `lvol_seq`, i veure si el tipus de sistema de fitxer afecta en el rendiment dels volums lògics. Per fer-ho, hem executat les següents comandes amb `mkfs`:

```
alumni@pcrecanvib5:~$ sudo mkfs -t ext3 /dev/vgcaso/lvol_banda
mke2fs 1.44.1 (24-Mar-2018)
Creating filesystem with 524288 4k blocks and 131072 inodes
Filesystem UUID: d9bb8d5d-55d9-4f84-b6c5-5ff0b4ba9394
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912

Allocating group tables: done
Writing inode tables: done
Creating journal (16384 blocks): done
Writing superblocks and filesystem accounting information: done
```



```

alumne@pcrecanvib5:~$ sudo mkfs -t xfs /dev/vgcaso/lvol_seq
meta-data=/dev/vgcaso/lvol_seq isize=512    agcount=4, agsize=131072 blks
       =                               sectsz=512   attr=2, projid32bit=1
       =                               crc=1        finobt=1, sparse=0, rmapbt=0, refl
inok=0
data      =                               bsize=4096   blocks=524288, imaxpct=25
       =                               sunit=0        swidth=0 blks
naming    =version 2                     bsize=4096   ascii-ci=0 ftype=1
log       =internal log                  bsize=4096   blocks=2560, version=2
       =                               sectsz=512   sunit=0 blks, lazy-count=1
realtime  =none                           extsz=4096   blocks=0, rtextents=0

```

Finalment, sols queda avaluar el rendiment d'escriptura i de lectura amb la comanda dd fent dos experiments.

Primer experiment:

```

alumne@pcrecanvib5:~$ sudo dd if=/dev/vgcaso/lvol_banda of=/tmp/test.dat bs=64k
count=16k conv=fdatasync
16384+0 records in
16384+0 records out
1073741824 bytes (1,1 GB, 1,0 GiB) copied, 1,35924 s, 790 MB/s
alumne@pcrecanvib5:~$ sudo dd if=/dev/vgcaso/lvol_seq of=/tmp/test.dat bs=64k co
unt=16k conv=fdatasync
16384+0 records in
16384+0 records out
1073741824 bytes (1,1 GB, 1,0 GiB) copied, 1,49382 s, 719 MB/s

```

Segon experiment després d'executar “sudo /sbin/sysctl -w vm.drop\_caches=3”:

```

alumne@pcrecanvib5:~$ sudo /sbin/sysctl -w vm.drop_caches=3
vm.drop_caches = 3
alumne@pcrecanvib5:~$ sudo dd if=/dev/vgcaso/lvol_seq of=/tmp/test.dat bs=64k co
unt=16k conv=fdatasync
16384+0 records in
16384+0 records out
1073741824 bytes (1,1 GB, 1,0 GiB) copied, 1,29448 s, 829 MB/s
alumne@pcrecanvib5:~$ sudo dd if=/dev/vgcaso/lvol_banda of=/tmp/test.dat bs=64k
count=16k conv=fdatasync
16384+0 records in
16384+0 records out
1073741824 bytes (1,1 GB, 1,0 GiB) copied, 1,15178 s, 932 MB/s

```

Podem comprovar amb els dos experiments que la velocitat d'escriptura en el lvol\_banda és una mica superior al de lvol\_seq. No sabem si es degut al sistema de fitxers o simplement perquè utilitza un RAID stripped. Pel que fa el segon experiment, la velocitat augmenta degut a que la comanda anterior, llibera la memòria caché de la màquina.