

## I. Configuration.

### Mirror setup.

Recently I was given a brand new X2100 server made by Sun Microsystems. I installed FreeBSD on it and run a mailserver. The server has two 250GB SATA drives and I decided to use gmirror(8) to create RAID-1 on those disks.

To start with the configuration, install FreeBSD, configure it to suit your needs. After server installation, I always make the world again, configure my new kernel and install it. Don't forget to add to your kernel config file the following lines:

```
options      GEOM_GPT
options      GEOM_MIRROR
```

To find out your system drives use atacontrol(8) command.

```
# atacontrol list
ATA channel 0:
  Master: acd0 <DV-28E-N/P.6A> ATA/ATAPI revision 5
  Slave:      no device present
ATA channel 1:
  Master:      no device present
  Slave:      no device present
ATA channel 2:
  Master: ad4 <ST3250823AS/3.03> Serial ATA v1.0
  Slave:      no device present
ATA channel 3:
  Master: ad6 <ST3250823AS/3.03> Serial ATA v1.0
  Slave:      no device present
ATA channel 4:
  Master:      no device present
  Slave:      no device present
ATA channel 5:
  Master:      no device present
  Slave:      no device present
```

My system is installed on ad4 disk and I want ad6 to be a second sub-mirror.

First of all, create a mirror gm0:

```
# gmirror label -vnb round-robin gm0 /dev/ad6
Metadata value stored on /dev/ad6.
Done.
```

-b round-robin is the algorithm used for reading.

-n turns off autosynchronization of stale components.

To turn in on use gmirror configure -a gm0.

In the command above you specify the second drive - ad6!

If GEOM\_MIRROR was compiled into the kernel, then /dev/mirror/gm0 device is already present.

Otherwise, initialize GEOM\_MIRROR,

```
# gmirror load
```

this command will load the /boot/kernel/geom\_mirror.ko kernel module and will create the gm0 device - /dev/mirror/gm0.

Edit loader.conf file:

```
# echo 'geom_mirror_load="YES"' >> /boot/loader.conf
```

Disk partition.

Now, use fdisk(8) command to create slice on gm0 device and reinitialize the boot code.

```
# fdisk -vBI /dev/mirror/gm0
***** Working on device /dev/mirror/gm0 *****
parameters extracted from in-core disklabel are:
cylinders=30401 heads=255 sectors/track=63 (16065 blks/cyl)
```

Figures below won't work with BIOS for partitions not in cyl 1

parameters to be used for BIOS calculations are:

```
cylinders=30401 heads=255 sectors/track=63 (16065 blks/cyl)
```

Information from DOS bootblock is:

```
1: sysid 165 (0xa5),(FreeBSD/NetBSD/386BSD)
   start 63, size 488392002 (238472 Meg), flag 80 (active)
   beg: cyl 0/ head 1/ sector 1;
   end: cyl 704/ head 254/ sector 63
```

```
2: <UNUSED>
3: <UNUSED>
4: <UNUSED>
fdisk: Geom not found
```

bsdlabel(8) is used to create BSD partition table and to install bootstrap code.

Install the bootstrap:

```
# bsdlabel -wB /dev/mirror/gm0s1
```

Read the partition table from the current system drive /dev/ad4s1.

Save the output.

```
# bsdlabel /dev/ad4s1
# /dev/ad4s1:
8 partitions:
#      size  offset  fstype  [fsize bsize bps/cpg]
a:  409600      0  4.2BSD   2048 16384 25608
b:  2097152  409600      swap
c:  488392002      0  unused      0      0      # "raw" part, don't edit
d:  4096000  2506752  4.2BSD   2048 16384 28552
e:  1024000  6602752  4.2BSD   2048 16384 64008
f:  81920000  7626752  4.2BSD   2048 16384 28552
g:  398845250  89546752  4.2BSD   2048 16384 28552
```

Edit a partition table of /dev/mirror/gm0s1:

```
# bsdlabel -e /dev/mirror/gm0s1
```

copy the output of bsdlabel /dev/ad4s1 command, paste and save the editor.

Make sure the size in sectors of /dev/ad4s1 and /dev/mirror/gm0s1 is the same.

The chances are, /dev/mirror/gm0s1 will be shorter of 1 sector in size.

In this case partition c will have to be shorter so the last partition (in my case partition g).

To verify the sizes type:

```
# diskinfo -v /dev/ad4s1 /dev/mirror/gm0s1 | egrep '(\/dev\/lin sectors)'
/dev/ad4s1
      488392002      # mediasize in sectors
/dev/mirror/gm0s1
      488392002      # mediasize in sectors
```

My slices were OK.

List the gm0 configuration:

```
# gmirror list
Geom name: gm0
State: COMPLETE
Components: 1
Balance: round-robin
Slice: 4096
Flags: NOAUTOSYNC
GenID: 0
SyncID: 1
ID: 3740434803
Providers:
1. Name: mirror/gm0
   Mediasize: 250059349504 (233G)
   Sectorsize: 512
   Mode: r0w0e0
Consumers:
1. Name: ad6
   Mediasize: 250059350016 (233G)
   Sectorsize: 512
   Mode: r1w1e1
   State: ACTIVE
   Priority: 0
   Flags: NONE
   GenID: 0
   SyncID: 1
   ID: 339682922
```

The gm0 provider has one component - consumer ad6.

Provider gm0 is the name of the mirror, ad6 consumer is the first sub-mirror.

State of gm0 is COMPLETE. This is OK.

Organize data.

My current drive setup looks like:

```
# df -h
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/ad4s1a     193M  120M   58M   67%      /
devfs           1.0K  1.0K   0B   100%    /dev
/dev/ad4s1g     184G   28K  169G    0%    /export/home
/dev/ad4s1e     484M   12K  445M    0%    /tmp
/dev/ad4s1f      38G   1.7G   33G    5%    /usr
/dev/ad4s1d     1.9G   15M   1.7G    1%    /var
```

For every partition created on gm0s1 create a filesystem.

In my case, I did it for /, /tmp, /var, /usr and /export/home.

```
# newfs /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
# newfs -U /dev/mirror/gm0s1g
```

For every partition created on gm0s1, mount it under /mnt and copy original data from /dev/ad4.

```
# mount /dev/mirror/gm0s1a /mnt
# dump -L -0 -f- / | ( cd /mnt && restore -r -v -f- )
# umount /mnt/
# mount /dev/mirror/gm0s1d /mnt
# dump -L -0 -f- /var | ( cd /mnt && restore -r -v -f- )
# umount /mnt/
# mount /dev/mirror/gm0s1e /mnt
# dump -L -0 -f- /tmp | ( cd /mnt && restore -r -v -f- )
# umount /mnt/
# mount /dev/mirror/gm0s1f /mnt
# dump -L -0 -f- /usr | ( cd /mnt && restore -r -v -f- )
# umount /mnt/
# mount /dev/mirror/gm0s1g /mnt
# dump -L -0 -f- /export/home | ( cd /mnt && restore -r -v -f- )
```

My /etc/fstab looks like:

# Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/ad4s1b	none	swap	sw	0	0
/dev/ad4s1a	/	ufs	rw	1	1
/dev/ad4s1g	/export/home	ufs	rw	2	2
/dev/ad4s1e	/tmp	ufs	rw	2	2
/dev/ad4s1f	/usr	ufs	rw	2	2
/dev/ad4s1d	/var	ufs	rw	2	2

Mount the /dev/mirror/gm0s1a again:

```
# mount /dev/mirror/gm0s1a /mnt/
```

and change /mnt/etc/fstab to:

# Device	Mountpoint	FStype	Options	Dump	Pass#
/dev/mirror/gm0s1b	none	swap	sw	0	0
/dev/mirror/gm0s1a	/	ufs	rw	1	1
/dev/mirror/gm0s1g	/export/home	ufs	rw	2	2
/dev/mirror/gm0s1e	/tmp	ufs	rw	2	2
/dev/mirror/gm0s1f	/usr	ufs	rw	2	2
/dev/mirror/gm0s1d	/var	ufs	rw	2	2

Make the same change to /etc/fstab file - the one on ad4 drive.

In case the systems does not boot from gm0 you can create those two files:

```
# echo "1:ad(4,a)/boot/loader" > /boot.config
# echo "1:ad(4,a)/boot/loader" > /mnt/boot.config
```

You can remove them later on.

Shutdown the system and keep fingers crossed.

```
# shutdown -r
```

After reboot, log into the system and check the mounting:

```
# df -h
Filesystem      Size  Used Avail Capacity  Mounted on
/dev/mirror/gm0s1a 193M 120M   58M   67% /
devfs            1.0K  1.0K    0B  100% /dev
/dev/mirror/gm0s1g 184G   19M 169G    0% /export/home
/dev/mirror/gm0s1e 484M   64K 445M    0% /tmp
/dev/mirror/gm0s1f  38G  1.6G  33G    5% /usr
/dev/mirror/gm0s1d 1.9G   16M  1.7G    1% /var
```

Mirror and synchronize.

Everything went OK, so it's time to add a second drive - the sub-mirror:

```
# gmirror insert -p 1 gm0 ad4
-p priority, specifies priority of the given component.
In this case ad4 has the priority 1 whereas ad6 0.
```

```
# gmirror list
Geom name: gm0
State: DEGRADED
Components: 2
Balance: round-robin
Slice: 4096
Flags: NOAUTOSYNC
GenID: 0
SyncID: 1
ID: 3740434803
Providers:
1. Name: mirror/gm0
   Mediasize: 250059349504 (233G)
   Sectorsize: 512
   Mode: r6w6e7
Consumers:
1. Name: ad6
   Mediasize: 250059350016 (233G)
   Sectorsize: 512
   Mode: r1w1e1
   State: ACTIVE
   Priority: 0
   Flags: DIRTY
   GenID: 0
   SyncID: 1
   ID: 339682922
2. Name: ad4
   Mediasize: 250059350016 (233G)
   Sectorsize: 512
   Mode: r1w1e1
   State: STALE
   Priority: 1
   Flags: SYNCHRONIZING
   GenID: 0
   SyncID: 1
   ID: 76271603
```

```
# gmirror rebuild gm0 ad4
```

```
# gmirror list
Geom name: gm0
State: DEGRADED
Components: 2
Balance: round-robin
Slice: 4096
Flags: NOAUTOSYNC
GenID: 0
SyncID: 1
ID: 3740434803
Providers:
1. Name: mirror/gm0
   Mediasize: 250059349504 (233G)
   Sectorsize: 512
   Mode: r7w6e7
Consumers:
1. Name: ad6
   Mediasize: 250059350016 (233G)
   Sectorsize: 512
   Mode: r1w1e1
   State: ACTIVE
   Priority: 0
```

```
Flags: DIRTY
GenID: 0
SyncID: 1
ID: 339682922
2. Name: ad4
Mediasize: 250059350016 (233G)
Sectorsize: 512
Mode: r1w1e1
State: SYNCHRONIZING
Priority: 1
Flags: DIRTY, SYNCHRONIZING, FORCE_SYNC
GenID: 0
SyncID: 1
Synchronized: 0%
ID: 76271603
```

Dont forget to turn on autosynchronization:

```
# gmirror configure -a gm0
```

or easier method:

```
# gmirror configure -a gm0
# gmirror insert -p 1 gm0 /dev/da0
```

To check the status of resynchronization:

```
# gmirror status
  Name      Status  Components
mirror/gm0  DEGRADED  ad6
              ad4 (26%)
# gmirror status
  Name      Status  Components
mirror/gm0  DEGRADED  ad6
              ad4 (99%)
# gmirror status
  Name      Status  Components
mirror/gm0  COMPLETE  ad6
              ad4
# gmirror list
Geom name: gm0
State: COMPLETE
Components: 2
Balance: round-robin
Slice: 4096
Flags: NONE
GenID: 0
SyncID: 1
ID: 3740434803
[.....]
```

The state is COMPLETE so the resynchronization went without any problems and components are ACTIVE now.

Kernel dumps.

If you expect kernel dumps, configure the dumpdev device.  
Edit /etc/rc.conf and add:  
dumpdev="/dev/mirror/gm0s1b"

To make savecore(8) life easier also edit /etc/rc.early  
and /etc/rc.local:

```
# echo "gmirror configure -b prefer gm0" >> /etc/rc.early
# echo "gmirror configure -b round-robin gm0" >> /etc/rc.local
```

Now the system is safe with RAID-1 over two system disks.

II. Breaking the mirror.

```
# gmirror remove gm0 ad6
# gmirror list
Geom name: gm0
State: COMPLETE
Components: 1
Balance: round-robin
Slice: 4096
Flags: NONE
```

```
GenID: 0
SyncID: 1
ID: 3740434803
Providers:
1. Name: mirror/gm0
  Mediasize: 250059349504 (233G)
  Sectorsize: 512
  Mode: r6w6e7
Consumers:
1. Name: ad4
  Mediasize: 250059350016 (233G)
  Sectorsize: 512
  Mode: r1w1e1
  State: ACTIVE
  Priority: 0
  Flags: DIRTY
  GenID: 0
  SyncID: 1
  ID: 76271603
```

The state is COMPLETE as there is only one component left - consumer ad4.

### III. Disk replacement

Assume drive ad4 is broken and it's not visible to the system.

```
# atacontrol list
ATA channel 0:
  Master: acd0 <DV-28E-N/P.6A> ATA/ATAPI revision 5
  Slave:   no device present
ATA channel 1:
  Master:   no device present
  Slave:   no device present
ATA channel 2:
  Master:   no device present
  Slave:   no device present
ATA channel 3:
  Master:  ad6 <ST3250823AS/3.03> Serial ATA v1.0
  Slave:   no device present
ATA channel 4:
  Master:   no device present
  Slave:   no device present
ATA channel 5:
  Master:   no device present
  Slave:   no device present
```

If the drives are hotswap, remove the broken ad4 drive and try to reinitialize it.

```
# atacontrol reinit ata2
Master:   no device present
Slave:    no device present
```

No luck.

```
# atacontrol attach ata4
atacontrol: ioctl(IOCATAATTACH): File exists
```

```
# atacontrol detach ata4
# atacontrol attach ata4
Master:   no device present
Slave:    no device present
```

```
# atacontrol reinit ata2
Master:   no device present
Slave:    no device present
```

To make a new drive visible to the system I had to reboot the server.

```
# gmirror insert -p 1 gm0 ad4
Not all disks connected. Try 'forget' command first.
```

```
To forget about components which are not connected use command:
# gmirror forget gm0
```

```
Add a component to the mirror.
# gmirror insert -p 1 gm0 ad4
```

```
Check if the mirror is synchronizing.
# gmirror list
```

```
Geom name: gm0
State: DEGRADED
Components: 2
Balance: round-robin
Slice: 4096
Flags: NONE
GenID: 0
SyncID: 6
ID: 1189193877
Providers:
1. Name: mirror/gm0
   Mediasize: 250059349504 (233G)
   Sectorsize: 512
   Mode: r7w6e7
Consumers:
1. Name: ad6
   Mediasize: 250059350016 (233G)
   Sectorsize: 512
   Mode: r1w1e1
   State: ACTIVE
   Priority: 0
   Flags: NONE
   GenID: 0
   SyncID: 6
   ID: 1830980100
2. Name: ad4
   Mediasize: 250059350016 (233G)
   Sectorsize: 512
   Mode: r1w1e1
   State: SYNCHRONIZING
   Priority: 1
   Flags: DIRTY, SYNCHRONIZING
   GenID: 0
   SyncID: 6
   Synchronized: 0%
   ID: 1487713881
```

As you can see, the drive replacement went OK and the synchronization process started. When it's finished, the state of gm0 should change from DEGRADED to COMPLETED.

After reading this article, I hope you will have a good insight on gmirror(8). I tried to write it in an easy to read and understand manner. Although I know Pawel Jakub Dawidek very well I must admit gmirror(8) is a very nice and easy tool for RAID-1. I really encourage you to use it for mirroring.