gmirror With Disk Partitions

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME
	2014-02-04		WB

Contents

1 Introduction 1

2 Making The Mirror More Specific 1

3 Setup 1

4 Done! 3

Last updated 2014-02-04

Available in HTML or PDF. Links to all my articles here. Created with AsciiDoc.

Warning
Due to head contention on mirror rebuilding, I do not recommend mirroring multiple partitions like this. At present, use MBR partitioning and mirror whole disks if the disks are 2T or smaller, as shown in the Handbook, or use ZFS for mirrors of larger disks.

`gmirror` is usually used to mirror entire disks. But sometimes mirroring only individual partitions on a disk has advantages.

This document describes setting up mirrored partitions, which has worked for me but should be tested and verified in your environment. Please test and back up before committing valuable data to this setup, as no warranty is expressed or implied.

1 Introduction

FreeBSD's *gmirror(8)* provides RAID1 mirroring of data between two or more disks. Duplicating data helps insure that at least one valid copy survives in the event of a hardware failure. The Handbook RAID1 chapter describes typical *gmirror* setup.

Tip
Mirroring protects against drive failure, but failure of a power supply or other shared component could still cause an outage. For even better protection, consider HAST.

A FreeBSD system with two 750G drives in a mirror was shut down and the secondary drive disconnected. With just the primary drive connected, */usr/src* was updated. At most, a few hundred megabytes changed. When the second drive was reconnected, the mirror took several hours to synchronize. Likely it copied the entire primary drive; without a journal that would keep track of changes, the mirror has no other way to make sure the secondary drive matches the first.

2 Making The Mirror More Specific

Is a whole-drive mirror really necessary? The test system described above really only needed 9G of disk space. Duplicating the rest of the drive didn't protect data, it was just wasting time copying unused space.

Rather than mirroring the whole disk, individual partitions can be mirrored. That has several advantages. Mirrored space can be limited, making for faster synchronization. If a drive fails, replacement partitions of the exact size needed can be created on a larger drive. Easier, or at least more convenient, than finding an entire drive to dedicate to the mirror.

The extra space on the drives can be used for other data that doesn't need to be mirrored. And there's twice as much of that space when it isn't mirrored.

Some additional setup in creating identical partitions on both disks is really about it.

3 Setup

Create a partition layout on the primary disk. This is exactly as if the drive were to be used alone, and in fact it will work that way. GPT labels are used so drive names and numbers are irrelevant. Done correctly, drives and filesystems should work regardless of the device names assigned by the controller, or the type of controller, or order or number of drives. It should be possible to set up a mirror of two drives on a single SATA controller, turn the system off, reconnect one to a different SATA controller and the other through a SATA to USB adapter to a USB port. . . and have the system come up and work without complaint.

```
# gpart create -s gpt ada0
# gpart add -t freebsd-boot -l boot-primary -s 64 ada0
# gpart bootcode -b /boot/pmbr -p /boot/gptboot -i 1 ada0
# gpart add -t freebsd-ufs -l rootfs-primary -b 2048 -s 4G ada0
# gpart add -t freebsd-swap -l swap-primary -s 4G ada0
# gpart add -t freebsd-ufs -l varfs-primary -s 8G ada0
# gpart add -t freebsd-ufs -l tmpfs-primary -s 8G ada0
# gpart add -t freebsd-ufs -l usrfs-primary -s 80G ada0
# gpart add -t freebsd-ufs -l otherfs-primary ada0
```

Create an identical layout on the secondary disk.

```
# gpart create -s gpt ada1
# gpart add -t freebsd-boot -l boot-secondary -s 64 ada1
# gpart bootcode -b /boot/pmbr -p /boot/gptboot -i 1 ada1
# gpart add -t freebsd-ufs -l rootfs-secondary -b 2048 -s 4G ada1
# gpart add -t freebsd-swap -l swap-secondary -s 4G ada1
# gpart add -t freebsd-ufs -l varfs-secondary -s 8G ada1
# gpart add -t freebsd-ufs -l tmpfs-secondary -s 8G ada1
# gpart add -t freebsd-ufs -l usrfs-secondary -s 80G ada1
# gpart add -t freebsd-ufs -l otherfs-secondary ada1
```

Load the *geom_mirror* kernel module before creating the mirror.

```
# kldload geom_mirror
```

Create the mirrors, one for each of the */*, *var*, *tmp*, and *usr* filesystems.

```
# gmirror label -v -b round-robin gmrootfs /dev/gpt/rootfs-primary /dev/gpt/rootfs- ↩
    secondary
# gmirror label -v -b round-robin gmvarfs /dev/gpt/varfs-primary /dev/gpt/varfs-secondary
# gmirror label -v -b round-robin gmtmpfs /dev/gpt/tmpfs-primary /dev/gpt/tmpfs-secondary
# gmirror label -v -b round-robin gmusrfs /dev/gpt/usrfs-primary /dev/gpt/usrfs-secondary
```

Create the filesystems on the mirror devices.

```
# newfs /dev/mirror/gmrootfs
# newfs -U /dev/mirror/gmvarfs
# newfs -U /dev/mirror/gmtmpfs
# newfs -U /dev/mirror/gmusrfs
```

Wait a second... what happened to *otherfs-primary* and *otherfs-secondary*? Left unmirrored, those large areas of the disk can be used for data that doesn't need to be mirrored. Backups, temporary files, whatever. Create filesystems on them, too.

```
# newfs -U /dev/gpt/otherfs-primary
# newfs -U /dev/gpt/otherfs-secondary
```

This example copies live filesystems onto newly-created mirrors that are on the same computer. The */* filesystem is copied last, so it will have the most current snapshot.

```
# dump -C16 -0uanL -f - /tmp | (mount /dev/mirror/gmtmpfs  /mnt && cd /mnt && restore -rf - ↩
     && cd && umount /mnt)
# dump -C16 -0uanL -f - /usr | (mount /dev/mirror/gmusrfs  /mnt && cd /mnt && restore -rf - ↩
     && cd && umount /mnt)
# dump -C16 -0uanL -f - /var | (mount /dev/mirror/gmvarfs  /mnt && cd /mnt && restore -rf - ↩
     && cd && umount /mnt)
# dump -C16 -0uanL -f - /    | (mount /dev/mirror/gmrootfs /mnt && cd /mnt && restore -rf - ↩
     && cd && umount /mnt)
```

Finally, do a few last configuration corrections on the new mirrored root filesystem.

```
# mount /dev/mirror/gmrootfs /mnt
```

/mnt/boot/loader.conf

```
geom_mirror_load="YES"
```

/mnt/etc/fstab

```
/dev/gpt/swap-primary   none            swap    sw              0       0
/dev/mirror/gmrootfs    /               ufs     rw              1       1
/dev/mirror/gmtmpfs     /tmp            ufs     rw              2       2
/dev/mirror/gmusrfs     /usr            ufs     rw              2       2
/dev/mirror/gmvarfs     /var            ufs     rw              2       2
/dev/gpt/otherfs-pri    /other-pri      ufs     rw,noauto       2       2
/dev/gpt/otherfs-sec    /other-sec      ufs     rw,noauto       2       2
```

4 Done!

That’s it. Boot from the new drives. In theory, the system should continue to work if a drive partially or completely fails, or if a controller fails, or if the drives are connected to a different controller. Even if a different type of controller is used, like an external USB to SATA adapter, it should still work.

"In theory" and "should" are important words here. Test this configuration before using it. If you discover a flaw or refinement, please let me know.