FreeBSD gvinum RAID5 on Sparc64

Not long ago a friend of mine generously donated me a Sun E450 server for use with my current networking projects.

The machine came along with four IBM DDRS34560 hard drives. Since a capacity of 4 GB per drive is not that overwhelming by today's standards I was looking forward to incorporate them into a RAID5 array.

The operating system of choice to achieve this was FreeBSD/sparc64 6.1-RELEASE (custom-built kernel, Sun E450 SMP Kernel Config).

This howto will not go into too much details. If you want to learn more about this stuff check out the man pages of gvinum, geom and have a look at the FreeBSD handbook, Chapter #19.

#1 Get Some Good Hard Drives

First of all you need to have at least three hard drives to build up a RAID5 plex (this is the term used for arrays in gvinum). It does however not make very much sense to build a RAID5 from three disks, making it a RAID1 with hotspare is a better approach then. I would use at least four disks (better five) but not more than seven. RAID5 makes heavy parity calculations, so the more disks you have, the longer it will take to spread the parity (don't do a RAID5 over 14 disks, it is much slower than a RAID0 spanned accross two RAID5's of seven disks).
Although it's possible to use hard drives of any size and even make them member of multiple plexes of any kind, I would not recommend to do so. It makes things a bit more complicated.
This is why I choose to always dedicate a single disk to a single plex only, making it (at least to look) similar to what ordinary hardware RAID controllers do.

To get a balanced weight of I/O performance I would also recommend that you always span a plesk accross identical disks. Using multiple disks of different kind could lead to a very unbalanced behaviour.
Imagine what happens if you span a RAID5 accross one 5400 rpm SCSI-1 and two 15k rpm U360 hard drives...

In my case I had four IBM DDRS34560 hard drives to get along.

#2 Hard Drive Partitions

When you have your hard drives ready you should setup the partitions (or disk labels to be correct).

Due to platform specific differences FreeBSD/sparc64 neither has an fdisk nor a bsdlabel command you could work with. The utility of choice to manage the disk label is called 'sunlabel'.

First of all initialize all drives with a new disk label (replace DEVICE by your appropriate device name):

sunlabel -w DEVICE auto

Then you should edit the disk label (again, replace DEVICE by your appropriate device name):

sunlabel -e DEVICE

You will certainly notice a difference compared with a disklabel on the i386 platform. Create a new a: partition and start it at offset 1 (the offset is required to allow gvinum meta data to be store on disk). Since sunlabel will only allow to use sector offsets, this will waste more space than what's actually needed for the meta data though this should not be a concern. Don't make the mistake to start at offset 0 though, it won't work out properly.

To set partition a: size take the partition c: size and reduce it by the amount you see in the 'sectors/cylinder' header.

```
# /dev/da2:
text: SUN4.2G cyl 3880 alt 2 hd 16 sec 135
bytes/sector: 512
sectors/cylinder: 2160
sectors/unit: 8380800

8 partitions:
#
# Size is in sectors.
# Offset is in cylinders.
#    size       offset     tag        flag
#    ---------- ----------  ----------  ----
a:    8378640          1  unassigned    wm
c:    8380800          0      backup    wm
```

Repeat these steps for all future member disks of the RAID5 plex.

If you have all identical disks, you could safely dump the disk label of your first device to a prototype file like this:

```
sunlabel DEVICE > sunlabel.DEVICE
```

Then restore the label to your other devices like this:

```
sunlabel -R NEW_DEVICE sunlabel.DEVICE
```

#3 Create gvinum RAID5 volume

Next you should create a sample configuration file (eg. /tmp/raid5.conf) for initialization. Consider that the chunk size (261k in the example) should not be a power of 2, otherwise you filesystem super blocks might end up on the same physical disk.

```
drive vol1_disk1 device /dev/da2a
drive vol1_disk2 device /dev/da3a
drive vol1_disk3 device /dev/da4a
drive vol1_disk4 device /dev/da5a

volume raid5_vol1
plex org raid5 261k
sd drive vol1_disk1
sd drive vol1_disk2
sd drive vol1_disk3
sd drive vol1_disk4
```

If you may also choose to build your array with a designated hotspare drive, which might then look like this:

```
drive vol1_disk1 device /dev/da2a
drive vol1_disk2 device /dev/da3a
drive vol1_disk3 device /dev/da4a
```

```
drive vol1_disk4 device /dev/da5a hotspare
volume raid5_vol1
plex org raid5 261k
sd drive vol1_disk1
sd drive vol1_disk2
sd drive vol1_disk3
```

Now invoke gvinum to create the RAID5 volume:

```
gvinum create /tmp/raid5.conf
```

This should print a status listing after initialization (the sample shows an array without hotspare drive):

```
4 drives:
D vol1_disk4          State: up       /dev/da5a       A: 0/4091 MB (0%)
D vol1_disk3          State: up       /dev/da4a       A: 0/4091 MB (0%)
D vol1_disk2          State: up       /dev/da3a       A: 0/4091 MB (0%)
D vol1_disk1          State: up       /dev/da2a       A: 0/4091 MB (0%)

1 volume:
V raid5_vol1          State: up       Plexes:       1 Size:         11 GB

1 plex:
P raid5_vol1.p0       R5 State: up       Subdisks:     4 Size:         11 GB

4 subdisks:
S raid5_vol1.p0.s3      State: up       D: vol1_disk4   Size:       4091 MB
S raid5_vol1.p0.s2      State: up       D: vol1_disk3   Size:       4091 MB
S raid5_vol1.p0.s1      State: up       D: vol1_disk2   Size:       4091 MB
S raid5_vol1.p0.s0      State: up       D: vol1_disk1   Size:       4091 MB
```

Additional status information should be visible in your dmesg output.

The status can be reviewed by invoking 'gvinum list' at any time.

Make sure that the configuration gets saved by running:

```
gvinum saveconfig
```

#4 Format And Mount gvinum RAID5 volume

Now you are ready to format and mount the RAID5 volume.

```
newfs /dev/gvinum/raid5_vol1
mount /dev/gvinum_raid5_vol1 /mnt
```

Add the device to your fstab to automatically mount it during startup. For this to work you should also instruct the boot loader to enable gvinum. Add this line to /boot/loader.conf:

```
geom_vinum_load="YES"
```

This step can be omitted if you have included geom_vinum with your kernel. This is however not recommended according to the FreeBSD manual.

#5 What Else Must Be Done?

Your RAID5 volume should be up an running by now.

The man pages of gvinum and geom will cover advanced topics, amongst them mirroring, concatenation and combinations thereof.

Special attention must be given to optimization, eg. how the chunk or stripe size and the filesystem block size affect read/write performance.