

Using NetBSD's RaidFrame On Alpha

Yet another software RAID howto, I seem to be growing these in a hurry after all ;-)

This time I will cover NetBSD's RaidFrame and how to setup RAID1 disk mirroring on a Digital Alpha PWS433au.

While the basic outline retains pretty much the concept given in the [NetBSD Guide](#), I thought to write down my personal experience on this matter.

The main reason for this is actually the fact that there is so much discrepancy in the information available. Some of the (partially misleading) facts I came along:

- RaidFrame does not support RootFS booting on Alpha
- RaidFrame requires fiddling around with 'sector offsets' on Alpha
- RaidFrame does not work at all on Alpha
- etc.

It is incredible how much information exists which does not go along with the official manual, the manpages or simply is not accurate enough for whatever reason. Most of it is actually *very* outdated, does not even reflect the newest developments whatsoever.

After reading the docs, scanning the newsgroups and trying things out on my own, I uncovered these myths:

- RaidFrame does support booting RootFS on Alpha
- RaidFrame does *not* require any sector offset magic on Alpha
- RaidFrame works in general on Alpha

Although the NetBSD Guide covers only RaidFrame on x86 and sparc64, it will work out on Alpha for the most relevant parts. The differences will be outlined below using the original headlines as used in the NetBSD Guide.

15.3.4 Preparing Disk1

Wipe out the disk first (note: on Alpha partition 'c' covers the whole disk, not partition 'd' as on x86):

```
#dd if=/dev/zero of=/dev/rsd1c bs=8k count=1
```

Then recreate the partition table:

```
#disklabel -r -e -I sd1
type: unknown
disk: Disk1
label:
flags:
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 19386
total sectors: 19541088
[...snip...]
8 partitions:
```

```
# size offset fstype [fsize bsize cpg/sgs]
a: 19541088 0 RAID # (Cyl. 0 - 19385)
c: 19541088 0 unused 0 0 # (Cyl. 0 - 19385)
```

Just make the RAID partition 'a' start at offset '0' and everything will be fine. You do not (and should not) add any additional sector offset to the partition as SRM requires all bootable partitions to be located at sector 0.

15.3.7. Setting up kernel dumps

This is basically the same as outlined in the manual though you may omit the first offset. The numbers are taken from the original manual, so change them according to your own disklabel output.

```
#dc

64 # RF_PROTECTED_SECTORS
+
19015680 # offset of raid0b
+p
19015744 # offset of swap within sd1
q
```

15.3.8. Migrating System to RAID

Most of this part is identical. The difference comes in only when installing the boot loader. This command will do the trick:

```
#installboot -v /dev/rsd1c /usr/mdec/bootxx_ffs
```

The boot loader will need to be installed onto the second disk later on step 15.3.10. Use the same command as required.

15.3.11. Testing Boot Blocks

If the boot loader is properly installed, your RaidFrame array should boot without any problems from both disks. This can be easily tested by issuing 'boot dkc100' or 'boot dkc200'. The actual device names of your hard drives can be seen by running 'show device' on the SRM console.

To let SRM automatically try multiple devices for system boot you will need to alter your SRM environment.

```
#set bootdef_dev dkc100,dkc200
```

Now SRM should always boot from dkc100 and use dkc200 in case of a failure of the first disk.