MessageLabs

Be certain

Introduction to
Linux Software RAID1

Andrew Oakley, Anti Spam Technical Architect

18 April 2007

RAID Basics

- RAID = Redundant Array of Inexpensive Disks
- Many drives to create a single volume
- Different levels, eg:
  - RAID0 = "Striping" - Large volume data is chunked ("striped") across drives (faster read/write but no redundancy)
  - RAID1 = "Mirroring" - Small volume copied identically two several drives (slower read/write but can survive drive loss)
  - RAID5 = "Block-Level Striping" - Large volume made up of many drives, data is arranged so that any one drive can fail and data can be rebuilt from blocks on surviving drives
- There are many other variations, I'm only going to talk about RAID1

Support

- Used to require dedicated hardware card
- Now also on some motherboards
- Hardware RAID is set up via extension to BIOS - you get an extra "Press F3 to set up RAID" screen or similar upon boot
- Linux has supported RAID in OS ("software RAID") since 2.4
- No need for hardware card or fancy motherboard in Linux!
- You can use any IDE, SCSI or SATA drives
- With RAID1 "Mirroring", you will see a volume equal only to the smallest drive size in the array (eg. 60gb & 40gb = 40gb)

Easy choices

MessageLabs

- I'm only going to talk about RAID1 "Mirroring"
- Create a single volume, identical copy on each of two disks
- Either disk can fail and we will keep our data
- Ideal for:
  - Surviving drive hardware failure
  - Confidence when using second-hand hardware
  - Keeping a moment-by-moment "backup" of data
  - Typical Samba fileserver, home mail server, low/medium use webserver
- Not ideal for:
  - Gaming and other intensive disk IO applications
  - Both reads and writes are marginally slower
  - NOT A SUBSTITUTE FOR OFFLINE BACKUPS
    If you delete the data, it is just as gone as normal!

Avoid Legacy Tools

- Software RAID HOWTO is woefully out of date
- raidtools are deprecated as of kernel 2.4
- fd RAID volume type (eg. in fdisk) is also deprecated
- mdadm is what you want to use, certainly on distros <5 years old
- mdadm rarely uses config file - don't bother with config file,
  it will almost always autodetect and ignore it anyway

Assumptions

- You have a single root partition on /dev/hda1
- You have a spare drive /dev/hdc that is the same size or bigger
- You want to create a RAID1 mirror of your root filesystem
- You want to boot from root partition using GRUB
  (If not, this demo will still be interesting, but ignore GRUB bits)
- You have kernel >2.4 and mdadm installed
- You have recent verified backups of all important files

Get Started

- Create a partition on /dev/hdc1 that is the same size or larger than /dev/hda1
- Create a new RAID device made of /dev/hdc1 plus a special virtual device called "missing" - we will fill this with hda1 later

```
# mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/hdc1 missing
```

- Format /dev/md0

```
# mkfs.ext3 /dev/md0
```

- Mount new /dev/md0

```
# mkdir /newroot
```

```
# mount /dev/md0 /newroot
```

- Copy existing root filesystem onto /dev/md0

```
# cp -axv / /newroot
```

Reboot into RAID

- Change grub (or lilo) to mount root as /dev/md0

```
title           Ubuntu, kernel 2.6.15-28-k7
root            (hd0,0)
kernel          /boot/vmlinuz-2.6.15-28-k7 root=/dev/md0 ro quiet splash
initrd          /boot/initrd.img-2.6.15-28-k7
savedefault
boot
```

- Don't forget to copy /boot/grub/menu.lst to /newroot

```
# cp /boot/grub/menu.lst /newroot/boot/grub/menu.lst
```

- Setup GRUB

```
grub> root (hd2,0)
grub> setup (hd2,0)
grub> quit
```

Check everything

- Reboot. If it fails to boot, use GRUB menu to go back to previous config.
- Otherwise you should now be on /dev/md0

```
# df
# cat /proc/mdstat
```

- Should show root filesystem is /dev/md0
- Should show /dev/md0 has one device and one missing [U_]
- MAKE SURE YOUR SYSTEM WORKS NOW. We are about to add /dev/hda1 to the array, overwriting it - point of no return!

```
# mdadm --manage /dev/md0 --add /dev/hda1
# cat /proc/mdstat
```

That's All Folks

- You're done! Reboot and check everything is okay.
- If something's gone wrong, you can use rescue disk to boot from /dev/hda1 or /dev/hdc1 - you can boot from either without needing RAID, they are normal ext3 filesystems!
  (Note: This only applies to RAID1! Very useful feature!)
- You can observe RAID status with:

```
# cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 hdc1[0] hda1[1]
      38074560 blocks [2/2] [UU]
unused devices: <none>
```

- mdmonitor can email you when a drive fails or other event action

Gotchas

- mdadm will ignore it's own config file when booting from RAID (well, duh, how could it read the config file from something it's only just about to boot from?)
- If you have multiple RAID arrays, mdadm boot autodetect will arrange arrays in order of drive letter:
  - eg. two RAID volumes hda1 & hdc1=md0 , hdb1 & hdd1=md1
  - You can't change it so that hda1 & hdc1=md1 and vice versa
  - Unless you don't boot from RAID
- Personally I don't see much point having RAID that isn't bootable, if the boot partition fails then the system is dead
- You can do really cool stuff by taking one drive out of a RAID1 array and putting it in another system! Instant system copy!