

Chapter 24 RAID - Notes

24.2 Introduction

Use of RAID spreads I/O activity over multiple physical disks, rather than just one. Purpose to enhance data integrity and recoverability in case of failure, as well as to boost performance when used with modern storage devices. Number of different RAID levels which vary in their relative strengths in safety, performance, complexity, cost.

24.3 Learning Objectives:

- Explain the concept of RAID.
- Summarize RAID levels.
- Configure a RAID device using the essential steps provided.
- Monitor RAID devices in multiple ways.
- Use **hot spares**.

24.4 RAID

RAID (Redundant Array of Independent Disks) spreads I/O over multiple disks. Can really increase performance in modern disk controller interfaces, such as SCSI, which can perform the work in parallel efficiently.

RAID can be implemented either in software (it is a mature part of Linux kernel) or in hardware. If your hardware RAID is known to be of good quality, should be more efficient than using software RAID. With hardware implementation, operating system is actually not directly aware of using RAID -> it is transparent. Eg. three 512GB hard drives (two for data, one for parity) configured with RAID-5 will just look like a single 1TB disk.

However, one disadvantage of using hardware RAID: if disk controller fails, must be replaced by compatible controller, which may not be easy to obtain. When using software RAID, same disks can be attached to + work with any disk controller. Such considerations more likely to be relevant for low/mid-range hardware.

Three essential features of RAID:

- **mirroring**: writing the same data to more than one disk
- **striping**: splitting of data to more than one disk
- **parity**: extra data is stored to allow problem detection and repair, yielding fault tolerance

Thus, use of RAID can improve both performance + reliability.

One of main purposes of a RAID device: to create filesystem which spans more than one disk. Allows us to create filesystems which are larger than any one drive. RAID devices typically created by combining partitions from several disks together.

Another advantage of RAID devices: ability to provide better performance, redundancy, or both. Striping provides better performance by spreading information over multiple devices so simultaneous writes possible. Mirroring writes same information to multiple drives, giving better redundancy.

mdadm: used to create/manage RAID devices.

Once created, the array name, `/dev/mdX`, can be used just like any other device, such as `/dev/sda1`.

24.5 RAID Levels

Number of RAID specifications of increasing complexity and use. Most commonly used levels: 0, 1, 5.

- **RAID 0** uses only striping. Data spread across multiple disks. However, in spite of name, no redundancy, no stability or recovery capabilities. In fact, if any disk fails, data will be lost. But performance can be improved significantly because of parallelization of I/O tasks.
- **RAID 1** uses only mirroring. Each disk has duplicate. Good for recovery. At least two disks required.
- **RAID 5** uses rotating parity stripe. A single drive failure will cause no loss of data, only performance drop. Must be at least 3 disks.
- **RAID 6** has striped disks with dual parity. Can handle loss of two disks, requires at least 4 disks. Because RAID 5 can impose significant stress on disks, which can lead to failures during recovery procedures, RAID 6 has become more important.
- **RAID 10**: mirrored and striped data set. At least 4 drives needed.

As general rule, adding more disks improves performance.

24.6 Software RAID Configuration

Essential steps on configuring a software RAID device:

- Create partitions on each disk (type `fd` in **fdisk**)
- Create RAID devices with **mdadm**
- Format RAID device
- Add device to `/etc/fstab`
- Mount RAID device
- Capture RAID details to ensure persistence

The command:

```
$ sudo mdadm -S
```

can be used to stop RAID.

For example, create two partitions of type `fd` on disks `sdb` and `sdX` (say `/dev/sdbX` and `/dev/sdX`) by running **fdisk** on each:

```
$ sudo fdisk /dev/sdb
$ sudo fdisk /dev/sdX
```

Then, set up the array, format it, add to configuration, and mount it:

```
$ sudo mdadm --create /dev/md0 --level=1 --raid-disks=2 /dev/sdbX /dev/sdX
$ sudo mkfs.ext4 /dev/md0
$ sudo bash -c "mdadm --detail --scan >> /etc/mdadm.conf"
$ sudo mkdir /myraid
$ sudo mount /dev/md0 /myraid
```

Be sure to add a line in `/etc/fstab` for the mount point:

```
/dev/md0 /myraid ext4 defaults 0 2
```

Can examine `/proc/mdstat` to see RAID status:

```
$ cat /proc/mdstat
Personalities : [raid1]
md0 : active raid1 sdb8[1] sdc7[0]
----- 521984 blocks [2/2]
unused devices: <none>
```

Can use:

```
$ sudo mdadm -S /dev/md0
```

to stop the RAID device.

24.7 Monitoring RAIDs

Can monitor RAID devices multiple ways:

```
$ sudo mdadm --detail /dev/md0
$ sudo /proc/mdstat
```

One can also use **mdmonitor**, which requires configuring `/etc/mdadm.conf`. Command:

```
$ sudo mdadm --detail /dev/mdX
```

will show current status of RAID device `/dev/mdX`. Another way to do this -> to examine `/proc` filesystem:

```
$ cat /proc/mdstat
```

Will show status of all RAID devices on system.

Can also use **mdmonitor** service by editing `/etc/mdadm.conf` and adding a line like:

```
MAILADDR email@somewhere.com
```

so that it notifies you with email sent to `email@somewhere.com` when a problem occurs with a RAID device, eg. when any of the arrays fail to start or fall into degraded state. Can turn it on with:

```
$ sudo systemctl start mdmonitor
```

and make sure it starts at boot with:

```
$ sudo systemctl enable mdmonitor
```

(Note: on Ubuntu systems, the service is called **mdadm** rather than **mdmonitor**)

24.8 RAID Hot Spares

Redundancy: one of the important things that RAID provides. To help ensure any reduction in that redundancy is fixed as quickly as possible, a **hot spare** can be used.

To create hot space when creating RAID array:

```
$ sudo mdadm --create /dev/md0 -l 5 -n3 -x 1 /dev/sda8 /dev/sda9 /dev/sda10 /dev/sda11
```

`-x 1` switch tells **mdadm** to use one spare device. Note: hot spare can also be added at later time.

The command:

```
$ sudo mdadm --fail /dev/md0 /dev/sdb2
```

will test redundancy and hot spare of array.

To restore the tested drive, or new drive in legitimate failure situation, first remove "faulty" member, then add "new" member:

```
$ sudo mdadm --remove /dev/md0 /dev/sdb2  
$ sudo mdadm --add /dev/md0 /dev/sde2
```

##

[Back to top](#)

[Previous Chapter](#) - [Table of Contents](#) - [Next Chapter](#)