Managing software RAID with mdadm

Basics

mdadm **(multiple devices admin)** is an extremely useful tool for running RAID systems. It's is a tool for creating, managing, and monitoring RAID devices using the md driver. It can be used as a replacement for the raidtools, or as a supplement. You can use whole disks (/dev/sdb, /dev/sdc) or individual partitions (/dev/sdb1, /dev/sdc1) as a component of an array.

The benefits of using mdadm are:
1. mdadm can diagnose, monitor and gather detailed information about your arrays.
2. mdadm is a single centralized program and not a collection of disperse programs, so there's a common syntax for every RAID management command.
3. mdadm can perform almost all of its functions without having a configuration file and does not use one by default.

mdadm software tools work for all Linux distributions, with the same syntax.

Installing mdadm

Make sure you do a system update and then install the latest mdadm program into your system :

```
# yum clean all
# yum update
# yum install mdadm -y
```

The configuration file

– The /etc/mdadm.conf file is used to identify which devices are RAID devices and to which array a specific device belongs. This is required to auto-build your RAID devices at boot.
– By default, this file is not available, and needs to be created manually.
– Once you are done creating the RAID devices, you can simply create the mdadm.conf file by redirecting output of the command :

```
# mdadm --detail -scan
ARRAY /dev/md0 level=linear num-devices=2 metadata=1.2 name=localhost
   devices=/dev/sdb,/dev/sdc
```

As seen in the output above, I have a linear array md0 with 2 devices /dev/sdb and /dev/sdc.

```
# mdadm --verbose --detail -scan > /etc/mdadm.conf
```

Creating the RAID devices

To list the options to create RAID device with mdadm use the –help option. There are several options when creating RAID with mdadm. I will be listing few important ones.

```
# mdadm --create --help
        -C | --create /dev/mdn
        -l | --level  0|1|4|5
        -n | --raid-devices device [..]
        -x | --spare-devices device [..]
```

Linear mode
– Two or more disks are combined into one physical device.
– The disks are “appended” to each other, so writing linearly to the RAID device will fill up disk 0 first, then disk 1 and so on.
– The disks does not have to be of the same size.
– There is no redundancy in this level.
– The read and write performance will not increase for single reads/writes. But if several users use the device, several users using different disks at same time, you will see a performance gain.

To create two disks in linear mode running mdadm, just type a single command line:

```
# mdadm --create --verbose /dev/md0 --level=linear --raid-devices=2 /
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

The same command can be run using the shorter version of the options :

```
# mdadm --Cv /dev/md0 --l linear -n2 /dev/sdb /dev/sdc
```

RAID 0
– Also called “stripe” mode.
– The devices should have the same size.
– There is no redundancy in this level either. No data rescue is possible if a drive fails.
– The read and write performance will increase, because reads and writes are done in parallel on the devices.

To create two disks in RAID 0 mode:

```
# mdadm --create --verbose /dev/md0 --level=0 --raid-devices=2 /dev/s
```

RAID 1
– This level has redundancy.
– RAID-1 can be used on two or more disks with zero or more spare-disks.
– This mode maintains an exact mirror of the information on one disk on the other disk(s).
– Of Course, the disks must be of equal size.
– If one disk is larger than another, your RAID device will be the size of the smallest disk.
– If up to N-1 disks are removed (or crashes), all data are still intact. If there are spare disks available, and if the system survived the crash, reconstruction of the mirror will immediately begin on one of the spare disks, after detection of the drive fault.
– Write performance is often worse than on a single device as same data has to be written simultaneously on 2 or more devices.

You can setup RAID 1 with two disks and one spare disk:

```
# mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/s
```

RAID 4
– This RAID level is not used very often.
– It can be used on three or more disks.
– Instead of completely mirroring the information, it keeps parity information on one drive, and writes data to the other disks in a RAID-0 like way.
– Because one disk is reserved for parity information, the size of the array will be (N-1)*S, where S is the size of the smallest drive in the array.

– If one drive fails, the parity information can be used to reconstruct all data. If two drives fail, all data is lost.

To setup RAID 4 with 4 disks and one spare disk:

```
# mdadm --create --verbose /dev/md0 --level=4 --raid-devices=4 /dev/s
```

RAID 5
– RAID-5 can be used on three or more disks, with zero or more spare-disks.
– The resulting RAID-5 device size will be (N-1)*S, just like RAID-4.
– The big difference between RAID-5 and -4 is, that the parity information is distributed evenly among the participating drives, avoiding the bottleneck problem in RAID-4.
– If one of the disks fail, all data are still intact, thanks to the parity information. If spare disks are available, reconstruction will begin immediately after the device failure. If two disks fail simultaneously, all data are lost. RAID-5 can survive one disk failure, but not two or more.
– Reads are similar to RAID-0 reads, writes are generally expensive as parity has to be written which becomes the overhead.

To setup RAID 5 with 3 disks and 1 spare disk using mdadm:

```
# mdadm --create --verbose /dev/md0 --level=5 --raid-devices=3 /dev/s
```

Creating filesystem on RAID devices

To create a ext4 filesystem on the RAID device and to mount it :

```
# mkfs.ext4 /dev/md0
# mkdir /data01
# mount /dev/md0 /data01
```

Make sure you make an entry in /etc/fstab, to make it persistent across reboots.

```
# vi /etc/fstab
/dev/md0        /data01         ext4    defaults        0    0
```

Verifying Configuration

/proc/mdstat is a file maintained by the kernel which contains the real time information about the RAID arrays and devices.

```
# cat /proc/mdstat
Personalities : [linear]
md0 : active linear sdc[1] sdb[0]
      4194288 blocks super 1.2 0k rounding

unused devices: [none]
```

To get detailed information on a specific array use :

```
# mdadm --detail /dev/md0
/dev/md0:
        Version : 1.2
  Creation Time : Mon Nov  3 06:03:03 2014
     Raid Level : linear
     Array Size : 4194288 (4.00 GiB 4.29 GB)
   Raid Devices : 2
  Total Devices : 2
    Persistence : Superblock is persistent

    Update Time : Mon Nov  3 06:03:03 2014
          State : clean
 Active Devices : 2
Working Devices : 2
 Failed Devices : 0
  Spare Devices : 0

       Rounding : 0K

           Name : localhost.localdomain:0  (local to host localhost.l
           UUID : a50ac9f2:62646d92:725255bd:7f9d30e3
         Events : 0

    Number   Major   Minor   RaidDevice State
       0       8       16        0      active sync   /dev/sdb
       1       8       32        1      active sync   /dev/sdc
```

Stop/Start(assemble) and remove RAID array

To stop and existing array and completely remove it from the system :

```
# mdadm --stop /dev/md0
# mdadm --remove /dev/md0
```

To start (assemble) a stopped array :

```
# mdadm --asemble /dev/md0
```

NOTE : The assemble command reads the /etc/mdadm.conf file to start the array. In case you did not save your configuration in mdadm.conf before stopping the array, this command would fail. You can use the below command to recreate the mdadm.conf file :
# mdadm –examine –scan > /etc/mdadm.conf

Managing devices in array

adding a device

To add a new device to the array :

```
# mdadm --add /dev/md0 /dev/sdd
```

removing a device

We can fail a device (-f) from an array and then remove (-r) it:

```
# mdadm --manage /dev/md0 -f /dev/sdd
# mdadm --manage /dev/mdadm -r /dev/sdd
```