

Dernière mise-à-jour : 2020/01/30 03:28

LPI 204 - Gestion Avancée des Disques (8/60)

Ce cours couvre le contenu des trois sections suivantes du cursus LPI 201-450 :

- 204.1 - Configuring RAID (3/60)
- 204.2 - Adjusting Storage Device Access (2/60)
- 204.3 - Logical Volume Manager (3/60)

204.1 - Configuring RAID (3/60)

Description: Candidates should be able to configure and implement software RAID. This objective includes using and configuring RAID 0, 1 and 5.

Key Knowledge Areas

- Software raid configuration files and utilities

Terms and Utilities

- mdadm.conf
 - mdadm
 - /proc/mdstat
 - partition type 0xFD
-

204.2 - Adjusting Storage Device Access (2/60)

Description: Candidates should be able to configure kernel options to support various drives. This objective includes software tools to view & modify hard disk settings including iSCSI devices.

Key Knowledge Areas

- Tools and utilities to configure DMA for IDE devices including ATAPI and SATA
- Tools and utilities to configure Solid State Drives including AHCI and NVMe
- Tools and utilities to manipulate or analyse system resources (e.g. interrupts)
- Awareness of sdparm command and its uses
- Tools and utilities for iSCSI
- Awareness of SAN, including relevant protocols (AoE, FCoE)

Terms and Utilities

- hdparm, sdparm
- nvme
- tune2fs
- fstrim
- sysctl
- /dev/hd*, /dev/sd*, /dev/nvme*
- iscsiadm, scsi_id, iscsid and iscsid.conf
- WWID, WWN, LUN numbers

204.3 - Logical Volume Manager (3/60)

Description: Candidates should be able to create and remove logical volumes, volume groups, and physical volumes. This objective includes snapshots and resizing logical volumes.

Key Knowledge Areas

- Tools in the LVM suite
- Resizing, renaming, creating, and removing logical volumes, volume groups, and physical volumes
- Creating and maintaining snapshots
- Activating volume groups

Terms and Utilities

- /sbin/pv*
- /sbin/lv*
- /sbin/vg*
- mount
- /dev/mapper/
- lvm.conf

Dernière mise-à-jour : 2020/01/30 03:28

LPI 204 - Gestion Avancée des Disques (8/60)

Ce cours couvre le contenu des trois sections suivantes du cursus LPI 201-450 :

- 204.1 - Configuring RAID (3/60)
 - 204.2 - Adjusting Storage Device Access (2/60)
 - 204.3 - Logical Volume Manager (3/60)
-

204.1 - Configuring RAID (3/60)

Description: Candidates should be able to configure and implement software RAID. This objective includes using and configuring RAID 0, 1 and 5.

Key Knowledge Areas

- Software raid configuration files and utilities

Terms and Utilities

- mdadm.conf
- mdadm
- /proc/mdstat
- partition type 0xFD

204.2 - Adjusting Storage Device Access (2/60)

Description: Candidates should be able to configure kernel options to support various drives. This objective includes software tools to view & modify hard disk settings including iSCSI devices.

Key Knowledge Areas

- Tools and utilities to configure DMA for IDE devices including ATAPI and SATA
 - Tools and utilities to configure Solid State Drives including AHCI and NVMe
 - Tools and utilities to manipulate or analyse system resources (e.g. interrupts)
 - Awareness of sdparm command and its uses
 - Tools and utilities for iSCSI
 - Awareness of SAN, including relevant protocols (AoE, FCoE)
-

Terms and Utilities

- hdparm, sdparm
- nvme
- tune2fs
- fstrim
- sysctl
- /dev/hd*, /dev/sd*, /dev/nvme*
- iscsiadm, scsi_id, iscsid and iscsid.conf
- WWID, WWN, LUN numbers

204.3 - Logical Volume Manager (3/60)

Description: Candidates should be able to create and remove logical volumes, volume groups, and physical volumes. This objective includes snapshots and resizing logical volumes.

Key Knowledge Areas

- Tools in the LVM suite
- Resizing, renaming, creating, and removing logical volumes, volume groups, and physical volumes
- Creating and maintaining snapshots
- Activating volume groups

Terms and Utilities

- /sbin/pv*
 - /sbin/lv*
 - /sbin/vg*
 - mount
 - /dev/mapper/
-

- lvm.conf

Gestion des Disques Avancée - Raid Logiciel

Concepts RAID

Les solutions RAID ou *Redundant Array of Independent Disks* ou encore *Redundant Array of Inexpensive Disks* permettent la combinaison de plusieurs disques de façon à ce que ceux-ci soient vu comme un seul disque logique.

Les solutions RAID sont issues du travail fourni par l'université de Berkeley en Californie sur un projet de tolérances de pannes. Les systèmes RAID offre maintenant plusieurs avantages :

- Addition des capacités,
- Amélioration des performances,
- Apporter la tolérance de panne.

Deux concepts sont fondamentaux à la compréhension des solutions RAID.

Disques en miroir

La technique des disques en miroir consiste à dupliquer l'écriture des données sur plusieurs disques. Le miroir peut être géré par un logiciel ou par du matériel.

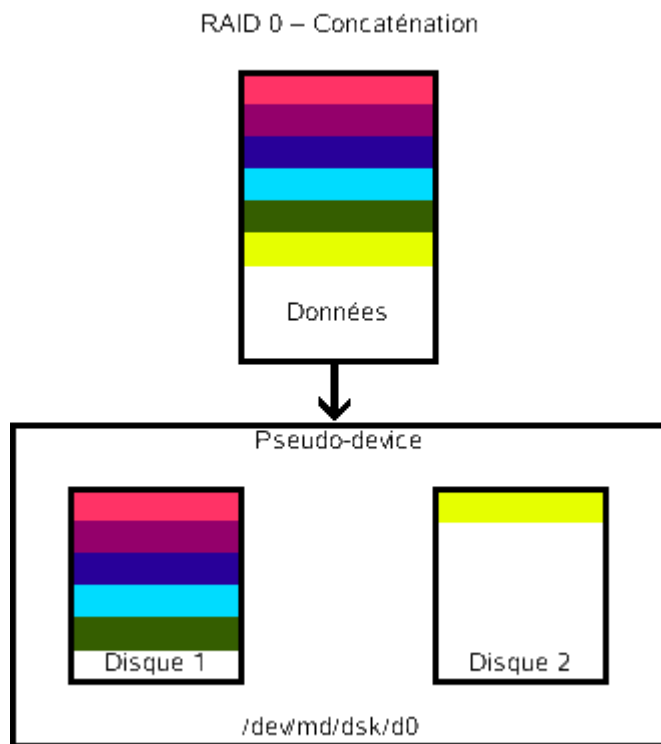
Bandes de données

La technique des bandes de données, autrement appelée *data striping* consiste à couper les données à enregistrer en segments séquentiels et contigus pour les enregistrer sur plusieurs disques physiques. L'ensemble des segments constitue alors un disque logique ou *striped disk*. Cette technique peut être améliorée en déposant une bande de parité, calculée à partir des données des autres bandes, afin de pouvoir reconstituer une bande de données défaillante.

Types de RAID

RAID 0 - Concaténation

Création de volume par récupération de l'espace libre sur un ou plusieurs disques. Le principe de la concaténation est la création d'un volume à bandes où chaque bande est une tranche.



Avantages

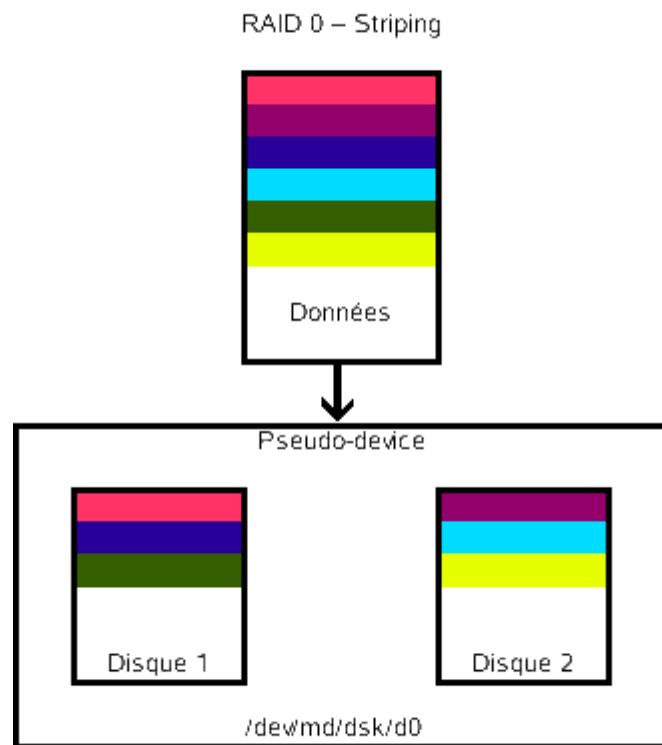
- Récupération de l'espace disque.

Inconvénients

- Pas de protection des données,
- Pas d'augmentation des performances d'E/S.

RAID 0 - Striping

Création de volume sur plusieurs disques afin d'augmenter les performances d'E/S. Le principe du striping est la création d'un volume à bandes réparties sur plusieurs tranches. La taille de la bande doit être fonction des données à écrire sur le volume (16k, 32k, 64k, etc.) Cette taille est choisie à la création du volume.



Avantages

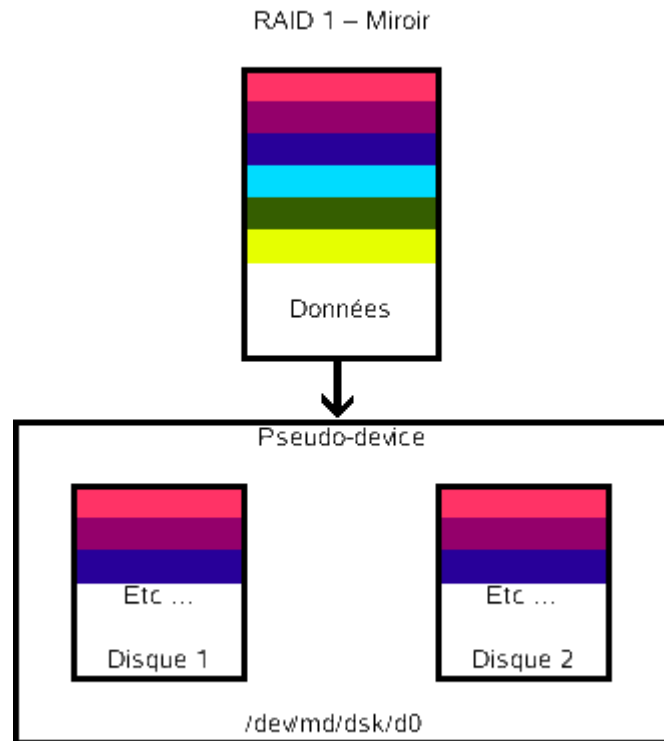
- Augmentation des performances d'E/S par écriture en parallèle sur les disques.

Inconvénients

- Pas de protection des données.

RAID 1 - Miroir

Création d'un volume où les disques sont en miroir. Quand les deux disques sont connectés à des contrôleurs de disques différents, on parle de *duplexing* :



Avantages

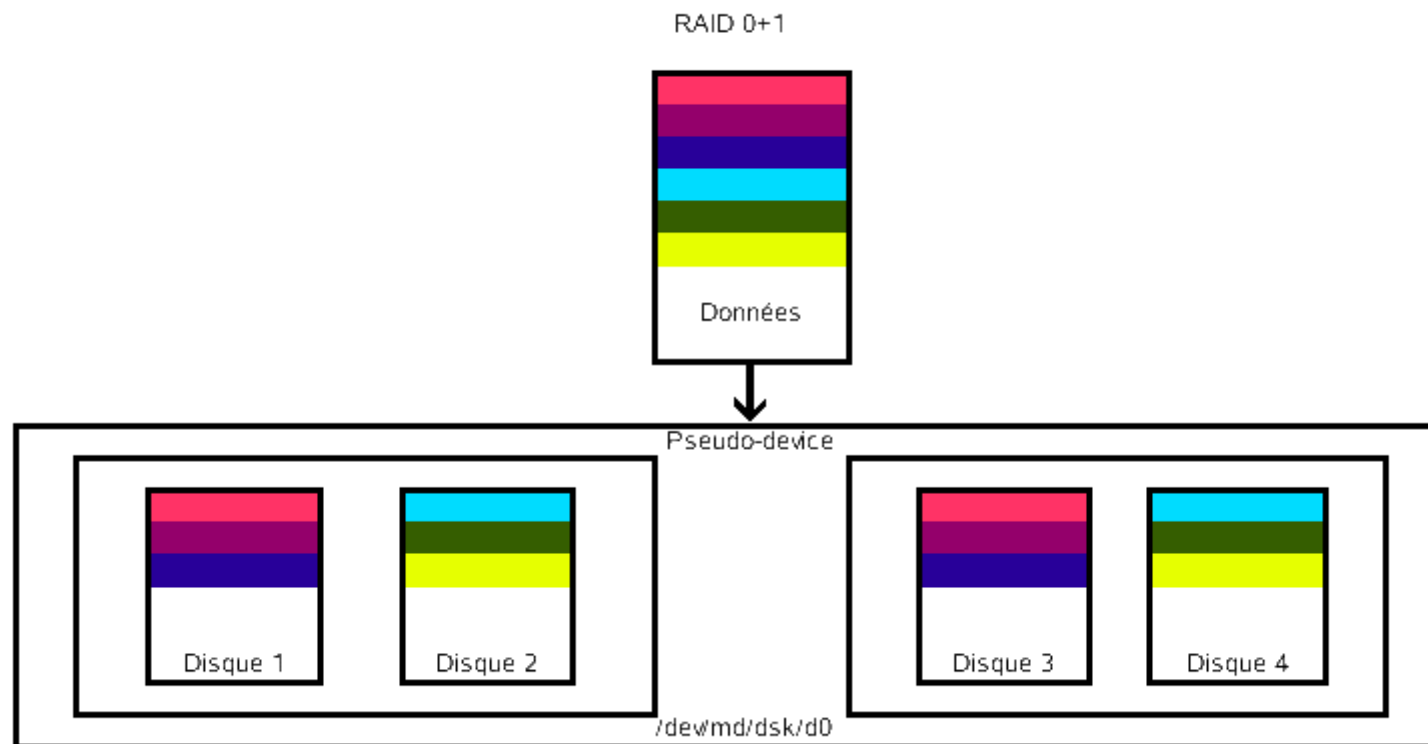
- Protection des données contre une défaillance d'un disque.

Inconvénients

- Coûteux à cause de l'augmentation du nombre de disques.

RAID 1+0 - Striping en Miroir

Le RAID 1+0 ou encore 0+1 est une technique qui réunit le RAID 0 et le RAID 1. On l'appelle aussi un RAID **exotique**:



Avantages

- Protection des données contre une défaillance d'un disque.
- Augmentation des performances d'E/S par écriture en parallèle sur les disques.

Inconvénients

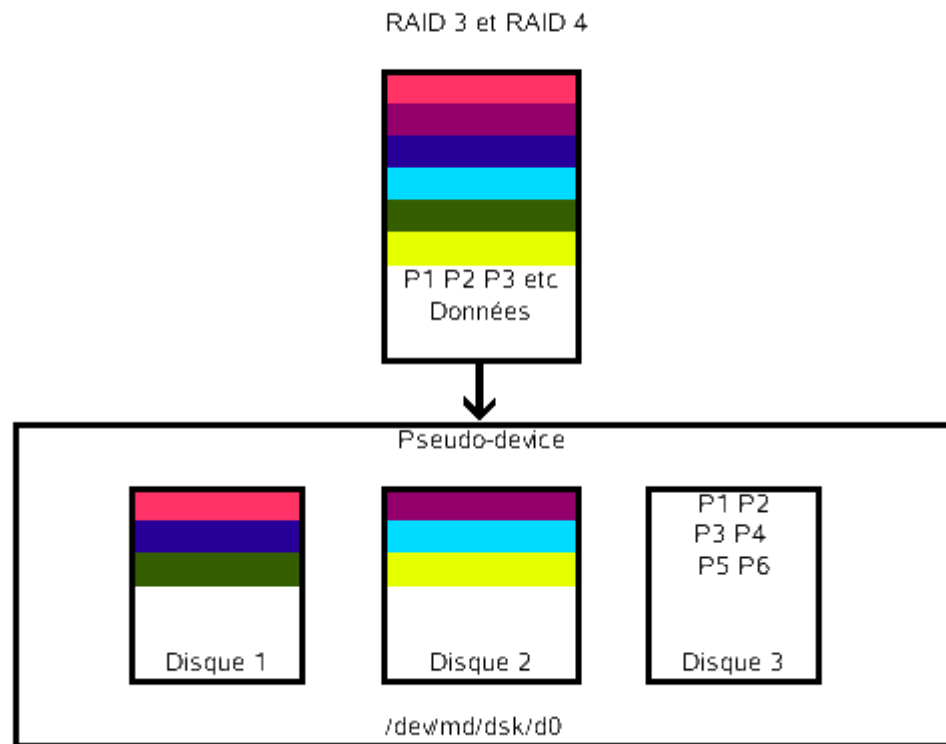
- Coûteux à cause de l'augmentation du nombre de disques.

RAID 2 - Miroir avec Contrôle d'Erreurs

Le RAID 2 est une technique de miroir avec contrôle de correction d'erreurs (EEC). De nos jours cette technique est peu utilisée, ayant été remplacée par les RAID 3, 4 et 5.

RAID 3 et 4 - Striping avec Parité

Les RAID 3 et 4 sont des technologies avec bandes de parité distribuées sur un seul disque :



En RAID 3, la taille des segments n'est pas modifiable et est fixée à 512 octets (en RAID 3 : un segment = un secteur de disque dur = 512 octets).

En RAID 4, la taille des segments est variable et se modifie en temps réel. Cela implique que les informations de parité doivent être mise à jour à chaque écriture afin de vérifier si la taille des segments a été modifiée.

Avantages

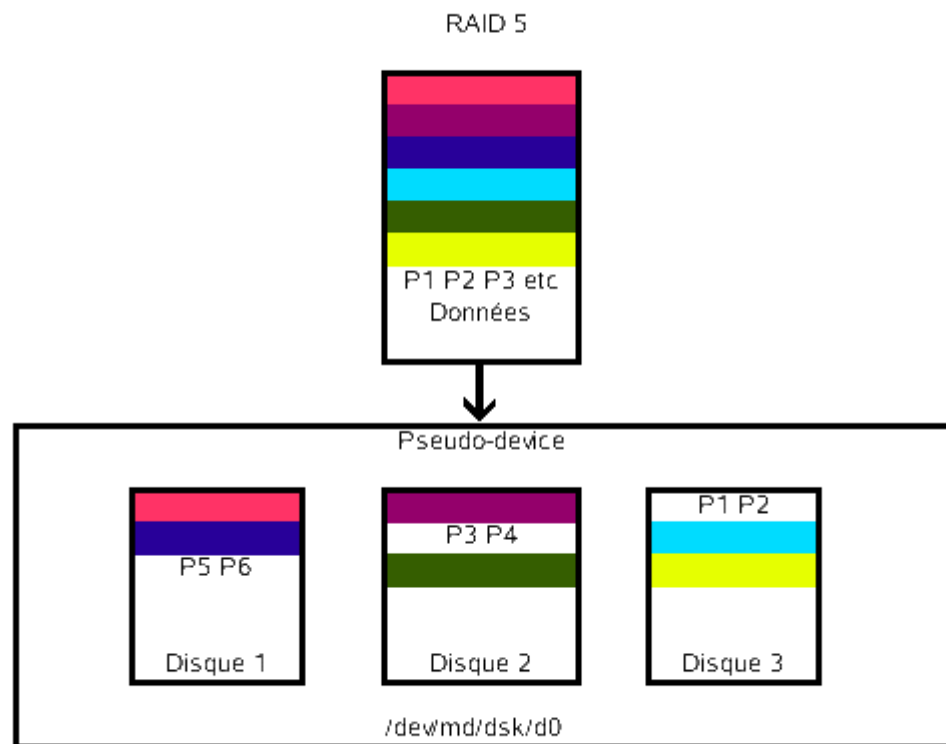
- Protection des données contre une défaillance d'un disque.

Inconvénients

- Création d'un goulot d'étranglement des données à cause de l'écriture des données de parité sur un seul disque.

RAID 5 - Striping avec Parité Distribuée

Le RAID 5 est une technologie avec bandes de parité distribuées sur plusieurs disques :



Avantages

- Protection des données contre une défaillance d'un disque,
- Evite le goulot d'étranglement d'un seul disque de parité.

Inconvénients

- Lecture moins performante qu'avec RAID 3 et 4.

Au delà de RAID 5

Il existe aussi deux autres technologies RAID, toutes deux issues de la technologie RAID 5 :

- RAID 6
 - *Disk Striping with Double Distributed Parity*
- RAID TP
 - *Disk Striping with Triple Distributed Parity*

RAID Logiciel sous CentOS

Préparation du disque

Partitionnement

Suite à votre LAB de la leçon **Gestion des Disques et le Swap**, votre disque comporte 12 partitions :

```
[root@centos7 ~]# fdisk -l
```

```
Disk /dev/sda: 21.5 GB, 21474836480 bytes, 41943040 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000c5a90
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	411647	204800	83	Linux
/dev/sda2		411648	20891647	10240000	83	Linux
/dev/sda3		20891648	25083903	2096128	82	Linux swap / Solaris
/dev/sda4		25083904	41943039	8429568	5	Extended
/dev/sda5		25085952	26109951	512000	fd	Linux raid autodetect
/dev/sda6		26112000	26521599	204800	8e	Linux LVM
/dev/sda7		26523648	27138047	307200	8e	Linux LVM
/dev/sda8		27140096	28164095	512000	fd	Linux raid autodetect
/dev/sda9		28166144	28985343	409600	8e	Linux LVM
/dev/sda10		28987392	30011391	512000	fd	Linux raid autodetect
/dev/sda11		30013440	31037439	512000	fd	Linux raid autodetect
/dev/sda12		31039488	31449087	204800	83	Linux

Mise en Place du RAID 5 Logiciel

Dans le cas de cet exemple les quatre partitions concernées par la mise en place d'un RAID 5 sont :

/dev/sda5	25085952	26109951	512000	fd	Linux raid autodetect
/dev/sda8	27140096	28164095	512000	fd	Linux raid autodetect
/dev/sda10	28987392	30011391	512000	fd	Linux raid autodetect
/dev/sda11	30013440	31037439	512000	fd	Linux raid autodetect

La création d'une unité RAID avec la commande **mdadm** se fait grâce aux options passées en arguments à la commande :

```
mdadm --create <unité RAID> [options] <unités physiques>
```


Par exemple, saisissez la commande suivante :

```
[root@centos7 ~]# mdadm --create /dev/md1 --level=5 --raid-devices=3 /dev/sda5 /dev/sda8 /dev/sda11
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md1 started.
```

Les options dans la ligne de commande sont :

Option Courte	Option Longue	Description
-l	- -level	Le niveau RAID - linear, 0,1,2,4 ou 5
-n	- -raid-devices=<nombre>	Le nombre de périphériques actifs dans le RAID

Les autres options de la commande **mdadm** peuvent être visualisées grâce à la commande suivante :

```
[root@centos7 ~]# mdadm --help-options
Any parameter that does not start with '-' is treated as a device name
or, for --examine-bitmap, a file name.
The first such name is often the name of an md device. Subsequent
names are often names of component devices.

Some common options are:
  --help          -h    : General help message or, after above option,
                        mode specific help message
  --help-options   : This help message
  --version        -V    : Print version information for mdadm
  --verbose        -v    : Be more verbose about what is happening
  --quiet          -q    : Don't print un-necessary messages
  --brief          -b    : Be less verbose, more brief
  --export         -Y    : With --detail, --detail-platform or --examine use
                        key=value format for easy import into environment
  --force          -f    : Override normal checks and be more forceful

  --assemble      -A    : Assemble an array
  --build          -B    : Build an array without metadata
```

```
--create      -C    : Create a new array
--detail      -D    : Display details of an array
--examine     -E    : Examine superblock on an array component
--examine-bitmap -X  : Display the detail of a bitmap file
--examine-badblocks: Display list of known bad blocks on device
--monitor     -F    : monitor (follow) some arrays
--grow        -G    : resize/ reshape and array
--incremental -I    : add/remove a single device to/from an array as appropriate
--query       -Q    : Display general information about how a
                     device relates to the md driver
--auto-detect : Start arrays auto-detected by the kernel
```

Les *modes majeurs* de la commande **mdadm** peuvent être visualisés grâce à la commande suivante :

```
[root@centos7 ~]# mdadm --help
mdadm is used for building, managing, and monitoring
Linux md devices (aka RAID arrays)
Usage: mdadm --create device options...
        Create a new array from unused devices.
mdadm --assemble device options...
        Assemble a previously created array.
mdadm --build device options...
        Create or assemble an array without metadata.
mdadm --manage device options...
        make changes to an existing array.
mdadm --misc options... devices
        report on or modify various md related devices.
mdadm --grow options device
        resize/reshape an active array
mdadm --incremental device
        add/remove a device to/from an array as appropriate
mdadm --monitor options...
        Monitor one or more array for significant changes.
mdadm device options...
```

Shorthand for `--manage`.

Any parameter that does not start with `'-'` is treated as a device name or, for `--examine-bitmap`, a file name.

The first such name is often the name of an md device. Subsequent names are often names of component devices.

For detailed help on the above major modes use `--help` after the mode e.g.

```
mdadm --assemble --help
```

For general help on options use

```
mdadm --help-options
```

Dernièrement, chaque *mode majeur* dispose de son propre aide :

```
[root@centos7 ~]# mdadm --assemble --help
Usage: mdadm --assemble device options...
       mdadm --assemble --scan options...
```

This usage assembles one or more raid arrays from pre-existing components.

For each array, mdadm needs to know the md device, the identity of the array, and a number of sub devices. These can be found in a number of ways.

The md device is given on the command line, is found listed in the config file, or can be deduced from the array identity.

The array identity is determined either from the `--uuid`, `--name`, or `--super-minor` commandline arguments, from the config file, or from the first component device on the command line.

The different combinations of these are as follows:

If the `--scan` option is not given, then only devices and identities listed on the command line are considered.

The first device will be the array device, and the remainder will be

examined when looking for components.

If an explicit identity is given with `--uuid` or `--super-minor`, then only devices with a superblock which matches that identity is considered, otherwise every device listed is considered.

If the `--scan` option is given, and no devices are listed, then every array listed in the config file is considered for assembly. The identity of candidate devices are determined from the config file. After these arrays are assembled, `mdadm` will look for other devices that could form further arrays and tries to assemble them. This can be disabled using the `'AUTO'` option in the config file.

If the `--scan` option is given as well as one or more devices, then Those devices are md devices that are to be assembled. Their identity and components are determined from the config file.

If `mdadm` can not find all of the components for an array, it will assemble it but not activate it unless `--run` or `--scan` is given. To preserve this behaviour even with `--scan`, add `--no-degraded`. Note that "all of the components" means as many as were present the last time the array was running as recorded in the superblock. If the array was already degraded, and the missing device is not a new problem, it will still be assembled. It is only newly missing devices that cause the array not to be started.

Options that are valid with `--assemble (-A)` are:

<code>--bitmap=</code>		: bitmap file to use with the array
<code>--uuid=</code>	<code>-u</code>	: uuid of array to assemble. Devices which don't have this uuid are excluded
<code>--super-minor=</code>	<code>-m</code>	: minor number to look for in super-block when choosing devices to use.
<code>--name=</code>	<code>-N</code>	: Array name to look for in super-block.
<code>--config=</code>	<code>-c</code>	: config file
<code>--scan</code>	<code>-s</code>	: scan config file for missing information
<code>--run</code>	<code>-R</code>	: Try to start the array even if not enough devices

```
for a full array are present
--force      -f  : Assemble the array even if some superblocks appear
                : out-of-date. This involves modifying the superblocks.
--update=    -U  : Update superblock: try '-A --update=?' for option list.
--no-degraded      : Assemble but do not start degraded arrays.
--readonly      -o  : Mark the array as read-only. No resync will start.
```

Constatez maintenant les informations concernant le RAID 5 créé :

```
[root@centos7 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md1 : active raid5 sda11[3] sda8[1] sda5[0]
      1021952 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/3] [UUU]
unused devices: <none>
```

Grâce à la commande mdadm, il est possible d'obtenir d'avantage d'informations :

```
[root@centos7 ~]# mdadm --query /dev/md1
/dev/md1: 998.17MiB raid5 3 devices, 0 spares. Use mdadm --detail for more detail.
```

L'option - **detail** produit le résultat suivant :

```
[root@centos7 ~]# mdadm --detail /dev/md1
/dev/md1:
  Version : 1.2
  Creation Time : Tue Aug  2 15:10:12 2016
  Raid Level : raid5
  Array Size : 1021952 (998.17 MiB 1046.48 MB)
  Used Dev Size : 510976 (499.08 MiB 523.24 MB)
  Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent

  Update Time : Tue Aug  2 15:10:17 2016
```

```
State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0

Layout : left-symmetric
Chunk Size : 512K

Name : centos7.fenestros.loc:1 (local to host centos7.fenestros.loc)
UUID : fd8bac53:1457a4f4:30feab84:e0f9d8e0
Events : 18
```

Number	Major	Minor	RaidDevice	State	
0	8	5	0	active sync	/dev/sda5
1	8	8	1	active sync	/dev/sda8
3	8	11	2	active sync	/dev/sda11

Notez la ligne **Persistence : Superblock is persistent**. En effet, cette implémentation de RAID inscrit les caractéristiques du volume dans un *super bloc* persistant en début de chaque unité de type bloc dans le volume.



Recherchez la signification des termes Layout, Chunk size, Major et Minor.

Cependant, il est nécessaire de renseigner le fichier **/etc/mdadm.conf** afin que le RAID soit construit à chaque démarrage :

```
[root@centos7 ~]# echo 'DEVICES /dev/sda5 /dev/sda8 /dev/sda11' > /etc/mdadm.conf
[root@centos7 ~]# mdadm --detail --scan >> /etc/mdadm.conf
[root@centos7 ~]# cat /etc/mdadm.conf
DEVICES /dev/sda5 /dev/sda8 /dev/sda11
ARRAY /dev/md1 metadata=1.2 name=centos7.fenestros.loc:1 UUID=fd8bac53:1457a4f4:30feab84:e0f9d8e0
```

Chaque unité peut être examinée individuellement :

```
[root@centos7 ~]# mdadm --examine /dev/sda5
/dev/sda5:
    Magic : a92b4efc
    Version : 1.2
    Feature Map : 0x0
    Array UUID : fd8bac53:1457a4f4:30feab84:e0f9d8e0
    Name : centos7.fenestros.loc:1 (local to host centos7.fenestros.loc)
    Creation Time : Tue Aug  2 15:10:12 2016
    Raid Level : raid5
    Raid Devices : 3

    Avail Dev Size : 1021952 (499.08 MiB 523.24 MB)
    Array Size : 1021952 (998.17 MiB 1046.48 MB)
    Data Offset : 2048 sectors
    Super Offset : 8 sectors
    Unused Space : before=1960 sectors, after=0 sectors
    State : clean
    Device UUID : 4569e15e:20201bce:365776f2:75aed23f

    Update Time : Tue Aug  2 15:10:17 2016
    Bad Block Log : 512 entries available at offset 72 sectors
    Checksum : 4c9948d0 - correct
    Events : 18

    Layout : left-symmetric
    Chunk Size : 512K

    Device Role : Active device 0
    Array State : AAA ('A' == active, '.' == missing, 'R' == replacing)
```

A ce stade il est intéressant de noter comment réagir lors d'une défaillance d'un disque. Dans notre cas nous allons indiquer au système que la partition /dev/sda5 est devenue défaillante :

```
[root@centos7 ~]# mdadm --manage --set-faulty /dev/md1 /dev/sda5
mdadm: set /dev/sda5 faulty in /dev/md1
```

L'utilisation de la ligne de commande suivante nous confirme le statut de /dev/sda5 :

```
[root@centos7 ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Tue Aug  2 15:10:12 2016
    Raid Level : raid5
    Array Size : 1021952 (998.17 MiB 1046.48 MB)
  Used Dev Size : 510976 (499.08 MiB 523.24 MB)
    Raid Devices : 3
  Total Devices : 3
  Persistence : Superblock is persistent

    Update Time : Tue Aug  2 15:14:59 2016
      State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 1
Spare Devices : 0

    Layout : left-symmetric
  Chunk Size : 512K

    Name : centos7.fenestros.loc:1 (local to host centos7.fenestros.loc)
    UUID : fd8bac53:1457a4f4:30feab84:e0f9d8e0
    Events : 20

   Number  Major   Minor   RaidDevice State
     0       0       0         0    removed
     1       8       8         1  active sync  /dev/sda8
     3       8      11         2  active sync  /dev/sda11
```


0	8	5	-	faulty	/dev/sda5
---	---	---	---	--------	-----------

Il est maintenant nécessaire de supprimer /dev/sda5 de notre RAID 5 :

```
[root@centos7 ~]# mdadm --manage --remove /dev/md1 /dev/sda5
mdadm: hot removed /dev/sda5 from /dev/md1
```

A l'examen de notre RAID, on constate que /dev/sda5 a été supprimé :

```
[root@centos7 ~]# mdadm --detail /dev/md1
/dev/md1:
    Version : 1.2
  Creation Time : Tue Aug  2 15:10:12 2016
    Raid Level : raid5
    Array Size : 1021952 (998.17 MiB 1046.48 MB)
  Used Dev Size : 510976 (499.08 MiB 523.24 MB)
    Raid Devices : 3
  Total Devices : 2
  Persistence : Superblock is persistent

    Update Time : Tue Aug  2 15:16:41 2016
      State : clean, degraded
Active Devices : 2
Working Devices : 2
Failed Devices : 0
Spare Devices : 0


    Layout : left-symmetric
  Chunk Size : 512K


    Name : centos7.fenestros.loc:1 (local to host centos7.fenestros.loc)
    UUID : fd8bac53:1457a4f4:30feab84:e0f9d8e0
    Events : 21
```

Number	Major	Minor	RaidDevice	State	
0	0	0	0	removed	
1	8	8	1	active sync	/dev/sda8
3	8	11	2	active sync	/dev/sda11

Constatez maintenant l'existence de votre RAID :

```
[root@centos7 ~]# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md1 : active raid5 sda11[3] sda8[1]
      1021952 blocks super 1.2 level 5, 512k chunk, algorithm 2 [3/2] [_UU]
unused devices: <none>
```



Notez que le RAID a été démarré avec 2 unités au lieu de trois.

Pour ajouter un autre disque à notre RAID afin de remplacer /dev/sda5 il convient d'utiliser l'option **--add** :

```
[root@centos7 ~]# mdadm --manage --add /dev/md1 /dev/sda10
mdadm: added /dev/sda10
```

L'examen du RAID indique que /dev/sda10 a été ajouté en tant que *spare* et à l'issue de quelques minutes le RAID 5 a été reconstruite :

```
[root@centos7 ~]# mdadm --detail /dev/md1
/dev/md1:
  Version : 1.2
  Creation Time : Tue Aug  2 15:10:12 2016
  Raid Level : raid5
  Array Size : 1021952 (998.17 MiB 1046.48 MB)
  Used Dev Size : 510976 (499.08 MiB 523.24 MB)
  Raid Devices : 3
  Total Devices : 3
```

```
Persistence : Superblock is persistent

Update Time : Tue Aug  2 15:19:06 2016
State : clean
Active Devices : 3
Working Devices : 3
Failed Devices : 0
Spare Devices : 0

Layout : left-symmetric
Chunk Size : 512K

Name : centos7.fenestros.loc:1 (local to host centos7.fenestros.loc)
UUID : fd8bac53:1457a4f4:30feab84:e0f9d8e0
Events : 40

Number   Major   Minor   RaidDevice State
  4         8       10         0    active sync  /dev/sda10
  1         8        8         1    active sync  /dev/sda8
  3         8       11         2    active sync  /dev/sda11
```

Naturellement, il est nécessaire de renseigner le fichier **/etc/mdadm.conf** du changement afin que le RAID soit construit à chaque démarrage :

```
[root@centos7 ~]# echo 'DEVICES /dev/sda10 /dev/sda8 /dev/sda11' > /etc/mdadm.conf
[root@centos7 ~]# mdadm --detail --scan >> /etc/mdadm.conf
[root@centos7 ~]# cat /etc/mdadm.conf
DEVICES /dev/sda10 /dev/sda8 /dev/sda11
ARRAY /dev/md1 metadata=1.2 name=centos7.fenestros.loc:1 UUID=fd8bac53:1457a4f4:30feab84:e0f9d8e0
```