

NOME

mdadm - istruzione di amministrazione delle unita' " md " [raid]

SINTASSI

mdadm [modo] <raiddevice> [opzioni] <componenti-devices>

DESCRIZIONE

I devices RAID sono unita' virtuali composti da due o piu' unita' a blocchi "fisiche" [in linux qualunque dispositivo a blocchi puo' essere messo in raid]. Queste unita' fisiche (solitamente hard disk) sono unite in un singolo device sul quale viene creato un file system. Alcuni tipi di RAID includono sistemi di rindondanza e possono cosi' sopravvivere al degrado di una o piu' unita' a blocchi fisiche. Il RAID software sotto linux e' implementato attraverso le unita' " md " [multiple devices]
Linux supporta RAID0, RAID1, RAID4, RAID5, RAID6, RAID10, MULTIPATH e FAULTY

MODO

mdadm ha diverse modalita' di funzionamento:

- Assemble unisce le unita' a blocchi (precedentemente dichiarate) in una matrice attiva.
- Build costruisce una matrice a partire dalle unita' a blocchi precedentemente dichiarate, tuttavia non vengono inizializzati i "superblocchi". Per questo motivo occorre una buona padronanza della tecnologia RAID per adoperare questa opzione.
- Create Crea un array partendo dalle unita' a blocchi dichiarate, creando anche i "superblock"

Follow

or Monitor controlla una o piu' " md " Questo e' significativo nei RAID 1, 4, 5, 6, 10 o multipath. RAID0 e lineare non hanno nessun vantaggio dal venir monitorate.

Grow

Manipolazione dell'array, in base alle caratteristiche dell'array stesso.

Incremental

Assembly Aggiunge un dispositivo a blocchi ad un determinato array. Se tale aggiunta rende l'array avviabile, esso sara' avviato.

Manage

Questa opzione e' per rimuovere device difettosi, oppure per aggiungere device validi ad un array

Misc

Questa opzione e' generica. Supporta la manipolazione su array attivi, cancellazione superblocchi obsoleti, ricerca nuovi superblocchi.

OPZIONI

Opzioni per scegliere i modi:

- A, --assemble Unisce un array preesistente
- B, --build Costruisce un array senza superblocchi
- C, --create Crea un nuovo array
- F, --follow,
 --monitor Seleziona la modalita' di monitoraggio
- G, --grow Manipola la forma e/o la dimensione di
 una matrice.

Aggiunge un singolo device in un array, ed ha la possibilita' di avviarlo. Se un device viene invocato senza specificare alcuna opzione, oppure la prima opzione e' una di queste: " --add --fail --remove " allora mdadm si dispone in modalita' " menage " . Ogni altra possibilita' indica a mdadm di assumere la modalita' " misc " .

OPZIONE CHE NON DIPENDONO ESPRESSAMENTE DA UNA MODALITA' DI FUNZIONAMENTO

- h, --help Mostra a video un messaggio di aiuto
- help-options Mostra a video un messaggio di aiuto con maggiori dettagli,
 dipendentemente da quanto espressamente chiesto.
- V, --version Mostra a video la versione del comando
- v, --verbose Aumenta la verbosita'. Puo' essere invocata due volte,
 " -vv " , aumentando ancor di piu' la verbosita'. Tale
 possibilita' viene applicata unicamente alle opzioni
 --detail --scan / --examine --scan
- q, --quiet Evitare di stampare messaggi puramente informativi. Invocato
 con tale opzione, mdadm cerchera' di lavorare "in silenzio",
 informando solo se accade qualcosa di veramente importante.
- b, --brief Diminuisce la verbosita'. Puo' essere utilizzato con --detail
 e --examine
- f, --force Forza l'esecuzione di un determinato comando.
- c, --config= Specifica un file di configurazione da utilizzare. Come
 impostazione predefinita tale file e' /etc/mdadm.conf , oppure
 /etc/mdadm/mdadm.conf . Se tali file non esistono, oppure
 se all'interno di tali files non vi sono informazioni utili,
 mdadm cerchera' tali utili informzioni all'interno del file
 /proc/partitions, soprattutto per cercare la lista delle
 partizioni attive e/o da utilizzare.
- s, --scan legge il file /proc/mdtsta alla ricerca di eventuali
 informazioni mancanti.
- e , --metadata= indica il tipo di superblock (raid metadata) da utilizzare.
 Come impostazione predefinita viene utilizzato 0,90 .
 E' possibile indicare come valori:

 0
 0,90

 questi due possibili valori hanno come limite 28 componenti
 e due terabyte .

 1
 1,0
 1,1

1,2

Utilizza il formato dei superblocchi nella nuova versione.
Questo ha delle limitazioni.

0, 0.90, default

Use the original 0.90 format superblock. This format
limits arrays to 28 component devices and limits compo-
nent devices of levels 1 and greater to 2 terabytes.

--homehost= [non riesco a tradurre]

OPZINI VALIDE PER create, build, grow

-n, --raid-devices= indica il numero dei device nell'array. Tale valore
piu' il numero dei devices "spare" deve coincidere
col numero dei component-devices

-x, --spare-devices= indica il numero dei devices "spare", cioe' di scorta
in un array. I devices spare possono essere aggiunti
(e rimossi) anche in seguito. Il numero dei "component
devices" specificati da linea di comando deve
coincidere con il numero dei "raid-devices" piu'
il numero dei "spare devices"

-z, --size= quantita' (in kbytes) di spazio da utilizzare per ogni unita'
nei RAID1/4/5/6 . Tale quantita' di spazio deve essere un
multiplo del "chunk size" e deve lasciare almeno 128 kbytes
di spazio alla fine dell'unita' da dedicare al superblock.
Se non specificato (e solitamente e' cosi') , la piu' piccola
unita' o partizione definisce le dimensioni, e se vi e' una
differenza di spazio superiore all'1% ecco che un messaggio
warning viene evidenziato.
Questo valore puo' essere settato tramite l'opzione --grow
nei RAID1/4/5/6. Se l'array e' stato creato con meno spazio
rispetto all'attuale, ecco che tale spazio inutilizzato
puo' essere reso disponibile con l'opzione --grow

-c, --chunk=indica la dimensione in kbytes del "chunk size".
L'impostazione predefinita e' 64

--rounding= Specify rounding factor for linear array (==chunk size)

-l, --level=Imposta il livello di RAID. Se utilizzato assieme
all'opzione --create , puo' assumere i seguenti valori:
linear, RAID0, stripe, RAID1, 1, mirror, RAID4, 4 , RAID5
5, RAID6, 6, RAID 10, 10, multipath, mp, faulty.
Oggettivamente, alcune di queste possibilita' sono sinonimi.
Se utilizzato assieme all'opzione --build , solo i seguenti
valori sono accettati: linear , stripe , RAID0, 0 , RAID1 ,
multipath, mp , faulty .
Non e' utilizzabile assieme all'opzione --grow

-p, --layout= questa opzione configura in modo molto fine il formato dei
dati nei RAID5 e RAID10, specialmente riguardo il "faulty"
e gli errori.

Il layout dei blocchi di parita' del RAID5 puo' essere
left-asymmetric , left-symmetric , right-asymmetric ,
right-symmetric , oppure in modo abbreviato la , ra , ls ,
rs . Come impostazione predefinita viene utilizzato il
left-symmetric

[seguono molte altre cose che non riesco a tradurre]

--parity= come --layout

-b, --bitmap= Specifica un file "bitmap" Maggiorni spiegazini a qst link
http://recensioni.ebay.it/Array-RAID1-ad-alte-prestazioni-e-capacita-in-casa_W0QQugidZ1000000006232716?ssPageName=BUYGD:CAT:-1:SEARCH:1

Ecco un sunto:

Gli array si creano molto facilmente appena collegati i dischi, con il comando mdadm; ecco un esempio:

```
mdadm --create /dev/md1 --raid-devices=2 --level=raid1  
--bitmap=internal /dev/sdd /dev/sde
```

Questo è il comando usato per creare l'array con i due Samsung (mdl), sdd ed sde sono i nomi attribuiti da Linux ai due dischi una volta connessi; notare come non sia richiesto partizionare i dischi nel caso in cui si voglia destinarli completamente all'array (/dev/sdd, non /dev/sdd!).

Di particolare importanza qui è il parametro "bitmap=internal", in quanto consente di specificare che la "write intention bitmap" deve essere salvata sul disco e non su un file esterno (come da default di mdadm). La write intention bitmap è l'indice che regola le scritture sui dischi e consente di aumentare le prestazioni in fase di scrittura

--bitmap-chunk= Imposta la grandezza " chunk " del file bitmap. Ogni bit corrisponde a diversi kbytes di dati memorizzati. Qualora viene utilizzato un file bitmap come impostazione predefinita viene utilizzata la maggior parte di spazio possibile

-W, --write-mostly subsequent devices lists in a --build, --create, or --add command will be flagged as 'write-mostly'. This is valid for RAID1 only and means that the 'md' driver will avoid reading from these devices if at all possible. This can be useful if mirroring over a slow link.

--write-behind Specify that write-behind mode should be enabled (valid for RAID1 only). If an argument is specified, it will set the maximum number of outstanding writes allowed. The default value is 256. A write-intent bitmap is required in order to use write behind mode, and write-behind is only attempted on drives marked as write-mostly.

--assume-clean Tell mdadm that the array pre-existed and is known to be clean.

It can be useful when trying to recover from a major failure as you can be sure that no data will be affected unless you actually write to the array. It can also be used when creating a RAID1 or RAID10 if you want to avoid the initial resync, however this practice - while normally safe - is not recommended. Use this only if you really know what you are doing.

--backup-file This is needed when --grow is used to increase the number of

raid-devices in a RAID5 if there are nospare devices available. See the section below on RAID_DEVICE CHANGES. The file should be stored on a separate device, not on the raid array being reshaped.

-N, --name= imposta un nome per l'array. Questo e' possibile solo all'atto della creazione di un array che utilizzi il superbloc versione 1. Il nome e' una stringa di testo che puo' essere utilizzata per identificare l'array.

-R, --run Forza l'avvio di un array anche se alcuni elementi dello stesso fanno parte di altri array od altri filesystema.

Normalmente mdadm chiede conferma prima di procedere.

`-f, --force` obbliga mdadm ad accettare la geometria ed il formato specificato senza chiedere conferma. Solitamente mdadm non procede alla creazione di un array con un'unica unità, e cercherà di creare un array raid 5 con un'unità mancante. Utilizzando l'opzione `--force` obblighiamo mdadm a non ragionare su questo fatto.

`-a, --auto{=no,yes,md,mdp,part,p}{NN}`

Instruct mdadm to create the device file if needed, possibly allocating an unused minor number. "md" causes a non-partitionable array to be used. "mdp", "part" or "p" causes a partitionable array (2.6 and later) to be used. "yes" requires the named md device to have a 'standard' format, and the type and minor number will be determined from this. See DEVICE NAMES below.

The argument can also come immediately after "-a". e.g. "-ap".

If `--auto` is not given on the command line or in the config file, then the default will be `--auto=yes`.

If `--scan` is also given, then any `auto=` entries in the config file will over-ride the `--auto` instruction given on the command line.

For partitionable arrays, mdadm will create the device file for the whole array and for the first 4 partitions. A different number of partitions can be specified at the end of this option (e.g. `--auto=p7`). If the device name ends with a digit, the partition names add a 'p', and a number, e.g. `"/dev/homelp3"`. If there is no trailing digit, then the partition names just have a number added, e.g. `"/dev/scratch3"`.

If the md device name is in a 'standard' format as described in DEVICE NAMES, then it will be created, if necessary, with the appropriate number based on that name. If the device name is not in one of these formats, then a unused minor number will be allocated. The minor number will be considered unused if there is no active array for that number, and there is no entry in `/dev` for that number and with a non-standard name.

`--symlink=no` Normalmente viene creato il device `/dev/md/`, e viene anche creato un link simbolico in `/dev/`, il cui nome inizia con " md ". L'opzione `--symlinks=no` impedisce la creazione di tale link. L'opzione `--symlink=yes` forza la creazione di tale link anche in assenza del file `/etc/mdadm.conf`

OPZIONI VALIDE PER assemble

`-u, --uuid=` uid dell'array da costruire. I devices che non hanno tale uid saranno esclusi

`-m, --super-minor="` Minor number " del device I device che non hanno questo " minor number " sono esclusi. quando viene creato un array [per esempio `/dev/md1`], allora il superblock conterra' in " minor number " 1. Anche se l'array sara' assemblato successivamente come `/dev/md2`

`-N, --name=` Specifica il nome col quale l'array sara' assemblato

`-f, --force` Forza la creazione dell'array anche se le informazioni del superblock appaiono inesatte.

`-R, --run` cerca di fare partire l'array anche se non tutti i componenti

di tale array sono presenti. mdadm come impostazione predefinita non avvia l'array in assenza di un componente. Normalmente, se non tutti i componenti sono presenti e l'opzione `--scan` non viene utilizzata, allora l'array sarà assemblato ma non avviato. Utilizzando l'opzione `--run` otteniamo l'avvio dell'array ugualmente.

`--no-degraded` questa opzione si comporta in modo diametralmente opposto all'opzione `--run`. Specificando `--no-degraded` ecco che otteniamo l'avvio dell'array solo a condizione che tutti i componenti siano attivi.

`-a, --auto{=no,yes,md,mdp,part}`
See this option under Create and Build options.

`-b, --bitmap=` specifica esattamente il file bitmap utilizzato alla creazione dell'array. Se un array ha un proprio file bitmap interno, allora non occorre specificare nulla.

`--backup-file=` se viene utilizzata l'opzione `--backup-file` per aumentare la ridondanza del raid5, ecco che in occasioni di crollo del raid possiamo utilizzare il file indicato [`--backup-file=FILE`] come argomento per l'opzione `--assemble` in modo da aumentare le possibilità di recupero dei dati.

`-U, --update=`
Update the superblock on each device while assembling the array. The argument given to this flag can be one of `sparc2.2`, `summaries`, `uuid`, `name`, `homehost`, `resync`, `byteorder`, `devicesize`, or `super-minor`.

The `sparc2.2` option will adjust the superblock of an array what was created on a Sparc machine running a patched 2.2 Linux kernel. This kernel got the alignment of part of the superblock wrong. You can use the `--examine --sparc2.2` option to mdadm to see what effect this would have.

The `super-minor` option will update the preferred minor field on each superblock to match the minor number of the array being assembled. This can be useful if `--examine` reports a different "Preferred Minor" to `--detail`. In some cases this update will be performed automatically by the kernel driver. In particular the update happens automatically at the first write to an array with redundancy (RAID level 1 or greater) on a 2.6 (or later) kernel.

The `uuid` option will change the uuid of the array. If a UUID is given with the `--uuid` option that UUID will be used as a new UUID and will NOT be used to help identify the devices in the array. If no `--uuid` is given, a random uuid is chosen.

The `name` option will change the name of the array as stored in the superblock. This is only supported for version-1 superblocks.

The `homehost` option will change the homehost as recorded in the superblock. For version-0 superblocks, this is the same as updating the UUID. For version-1 superblocks, this involves updating the name.

The `resync` option will cause the array to be marked dirty meaning that any redundancy in the array (e.g. parity for raid5, copies for raid1) may be incorrect. This will cause the raid system to perform a "resync" pass to make sure that all redundant information is correct.

The `byteorder` option allows arrays to be moved between machines with different byte-order. When assembling such an array for the first time after a move, giving `--update=byteorder` will cause `mdadm` to expect superblocks to have their byteorder reversed, and will correct that order before assembling the array. This is only valid with original (Version 0.90) superblocks.

The `summaries` option will correct the summaries in the superblock. That is the counts of total, working, active, failed, and spare devices.

The `devicesize` will rarely be of use. It applies to version 1.1 and 1.2 metadata only (where the metadata is at the start of the device) and is only useful when the component device has changed size (typically become larger). The version 1 metadata records the amount of the device that can be used to store data, so if a device in a version 1.1 or 1.2 array becomes larger, the metadata will still be visible, but the extra space will not. In this case it might be useful to assemble the array

with

`--update=devicesize`. This will cause `mdadm` to determine the maximum usable amount of space on each device and update the relevant field in the metadata.

`--auto-update-homehost`

This flag is only meaningful with auto-assembly (see discussion below). In that situation, if no suitable arrays are found for this homehost, `mdadm` will scan for any arrays at all and will assemble them and update the homehost to match the current host.

OPZIONI VALIDE PER `manage`

`-a, --add` aggiunge un device all'array

`--re-add` inserisce un device in precedenza tolto dall'array

`-r, --remove` toglie un device dall'array. tale device deve essere inattivo.
Per esempio un device guasto oppure un device di spare

`-f, --fail` identifica un determinato device come guasto

`--set-faulty` come per l'opzione `--fail`

Each of these options require that the first device list is the array to be acted upon and the remainder are component devices to be added, removed, or marked as fault. Several different operations can be specified for different devices, e.g.

```
mdadm /dev/md0 --add /dev/sda1 --fail /dev/sdb1 --remove /dev/sdb1
```

Each operation applies to all devices listed until the next operations.

If an array is using a write-intent bitmap, then devices which have been removed can be re-added in a way that avoids a full reconstruction but instead just updated the blocks that have changed since the device was removed. For arrays with persistent metadata (superblocks) this is done automatically. For arrays created with `--build` `mdadm` needs to be told that this device we removed recently with `--re-add`.

Devices can only be removed from an array if they are not in active use. i.e. that must be spares or failed devices. To remove an active device, it must be marked as faulty first.

OPZIONI VALIDE PER `misc`

-Q, --query analizza un dispositivo fisico (hard disk) al fine di capire se appartiene ad un array " md "

-D, --detail Riferisce i dettagli di uno o piu' device " md "

-E, --examine visualizza il contenuto del superbloc di un device " md "

--sparc2.2 UN array " md " creato ai tempi del kernel 2.2 ha differenze sostanziali con un array " md " creato in epoca recente. L'opzione --sparc2.2 [unitamente all'opzione --examine] si occupa di analizzare tale array e di porvi rimedio. Successivamente e' possibile assemblare l'array utilizzando le opzioni --assemble --update=sparc2.2

-X, --examine-bitmap mostra le informazioni inerenti il bitmap file

-R, --run avvia un array, anche se costruito in parte.

-S, --stop ferma un array

-o, --readonly attiva la modalita' di sola lettura di un determinato array

-w, --readwrite attiva la modalita' di lettura e scrittura di un determinato array

--zero-superblock Se il device " md " contiene un superbloc valido, allora tale superbloc viene sovrascritto con una serie di zeri. Utilizzando anche l'opzione --force tale superbloc viene sovrascritto in qualunque caso, anche se fosse in un formato non valido oppure danneggiato.

-t, --test utilizzato unitamente all'opzione --detail , mostra l'exit status del device " md "

-W, --wait For each md device given, wait for any resync, recovery, or reshape activity to finish before returning. mdadm will return with success if it actually waited for every device listed, otherwise it will return failure.

OPZIONI VALIDE PER incremental assembly mode

--rebuild-map, -r ricostruisce il map file (locato in /var/run/mdadm/map) che mdadm utilizza per ricordare quali array sono correntemente assemblati e come sono assemblati.

--run, -R attiva un device " md " anche se non tutti i device sono disponibili. A patto che sia raggiunto il numero minimo di device utili per fare funzionare l'array.

--scan, -s l'opzione --scan ha senso solo se utilizzata unitamente all'opzione -R . Così facendo verra' scansionato il map file degli array assemblati , e verra' tentato l'avvio degli array ancora fermi. Se gli array elencati in mdadm.conf necessitano di un file bitmap, esso sara' allegato prima.

OPZIONI VALIDE PER monitor

-m, --mail indica un indirizzo di posta elettronica [valido] ove inviare comunicazioni di errore .

-p, --program, --alert indica un programma da eseguire

qualora un avvenimento sia riscontrato

- y, --syslog indica che tutti gli eventi devono essere registrati tramite syslog. I messaggi hanno la "facility" "daemon" e diverse priorita'.
- d, --delay mdadm controlla gli array md ogni 60 secondi come impostazione predefinita. utilizzando l'opzione --delay ecco che possiamo variare questo valore .
- f, --daemonise indica a mdadm di lavorare sullo sfondo, per il monitoraggio degli array mdadm. Questo causa il "fork" del processo, e l'avvio di un processo figlio. Il pid del processo figlio viene visualizzato a terminale. E' molto utilizzata assieme all'opzione --scan , ed e' possibile fare in modo che mdadm invii un messaggio per posta elettronica.
- i, --pid-file Quialora mdadm venga avviato in modalita' " demone " , allora l'opzione -i consente di scrivere il pid del processo all'interno di un file specificato, oppure di stampare tale pid attraverso lo standard output
- l, --oneshot controlla gli array md solo una volta. Queste generera' un evento riferito agli array degradati ed agli "sparemissing" events. E' possibile inserire il comando mdadm --monitor --scan 1 all'interno di cron, in modo da ottenere notifiche sul funzionamento degli array , in modo regolare e costante.
- t, --test Genera un messaggio di errore test per ogni array effettivamente avviato. Tale messaggio puo' essere veicolato tramite mail. Cio' puo' essere utile per controllare l'efficacia del sistema di notifica.

OPZIONI VALIDE PER assemble

```
utilizzo: mdadm --assemble md-device options-and-component-devices...
```

```
utilizzo: mdadm --assemble --scan md-devices-and-options...
```

```
utilizzo: mdadm --assemble --scan options...
```

Questo tipo di utilizzo di mdadm unisce in un unico array " md " . Per ogni array " md " , mdadm deve conoscere il device " md " ed il numero di componenti.

Nel primo esempio (senza l'opzione --scan) il primo device e' un device " md " . Nel secondo esempio tutti i device elencati sono trattati come come device " md " , e per ognuno di essi viene tentato l'avvio. Nel terzo esempio (dove nessun device e' elencato) tutti i device " md " elencati nel file di configurazione vengono avviati

Se un device e' elencato ma l'opzione --scan non e' stata impartita, allora tale informazione e' ricavata dal file di configurazione /etc/mdadm.conf

L'identita' di un array puo' essere ottenuta tramite l'opzione --uuid , unitamente all'opzione --super-minor , oppure puo' essere ottenuta dal fle di configurazione , oppure puo' essere ricavata dal superbloc del primo "component device" elencato dalla linea di comando

I devices possono essere ottenuti tramite l'opzione `--assemble` presente nel file di configurazione. Solamente i device che possiedono un superbloc contenente la reale identita' del device saranno considerati validi

Un file di configurazione diverso da `/etc/mdadm.conf` viene utilizzato solo se espressamente indicato tramite l'opzione `--config`, oppure richiesto tramite l'opzione `--scan`. Il file di configurazione standard e' `/etc/mdadm.conf`

Se `--scan` non viene utilizzato, allora il file di configurazione sara' usato solo per ottenere l'identita' degli array.

Solitamente l'array viene avviato subito dopo l'operazione di assemblaggio. Tuttavia, se l'opzione `--scan` non viene impartita e non sono presenti un numero sufficienti di devices, ecco che l'array non viene avviato. In questo caso, per forzare l'avvio dell'array occorre utilizzare l'opzione `--run`

Se il device non esiste ecco che viene creato. Il nome deve essere in forma standard, altrimenti occorre specificare chiaramente tale nome

Cio' puo' essere utile per la gestione dei device partizionati, che non hanno un "number device" stabile. Oppure se si utilizza `udev`, il quale non puo' gestire device "md" a causa della differenza di gestione.

Se l'opzione "auto" o "part" o "p" [solo da linea di comando] viene utilizzata, allora `mdadm` creera' un array partizionabile, utilizzando la prima numerazione possibile non in uso.

Se l'opzione "auto" e' "yes" oppure "md" [da linea dicomando] oppure e' nulla, allora `mdadm` creera' un array tradizionale, non partizionabile.

Ci si aspetta che le funzionalita' "auto" vengano utilizzate per etichettare i device "md" con nomi alfanumerici [es: `/dev/500gb`] piuttosto che coi nomi numerici tradizionali.

Quando viene utilizzata questa opzione per creare un array partizionabile, i file "device" per le prime quattro partizioni vengono create. Se occorre una numerazioen differente, essa puo' essere aggiunta tramite le "auto" option. Per esempio "auto=part8". I nomi delle partizioni sono creati aggiungendo una stringa letterale al nome del device, con l'aggiunta di una "p" se il nome del device termina con un valore numerico

L'opzione `--auto` e' anche disponibile nei modi "build" e "create". In questi modi non occorre utilizzare il file di configurazione. Anche la possibilita' "auto" config non e' applicabile in questi modi.

Auto Assembly

Quando `--assemble` e' utilizzato in unione a `--scan` e nessun device e' elencato, `mdadm` cercera' prima di tutto di assemblare tutti i devices presenti nel file di configurazione.

If a homehost has been specified (either in the config file or on the command line), `mdadm` will look further for possible arrays and will try to assemble anything that it finds which is tagged as belonging to the given homehost. This is the only situation where `mdadm` will assemble arrays without being given specific device name or identify information

for the array.

If mdadm finds a consistent set of devices that look like they should comprise an array, and if the superblock is tagged as belonging to the given home host, it will automatically choose a device name and try to assemble the array. If the array uses version-0.90 metadata, then the minor number as recorded in the superblock is used to create a name in /dev/md/ so for example /dev/md/3. If the array uses version-1 metadata, then the name from the superblock is used to similarly create a name in /dev/md. The name will have any 'host' prefix stripped first.

If mdadm cannot find any array for the given host at all, and if --auto-update-homehost is given, then mdadm will search again for any array (not just an array created for this host) and will assemble each assuming --update=homehost. This will change the host tag in the superblock so that on the next run, these arrays will be found without the second pass. The intention of this feature is to support transitioning a set of md arrays to using homehost tagging.

The reason for requiring arrays to be tagged with the homehost for auto assembly is to guard against problems that can arise when moving devices from one host to another.

BUILD MODE

Utilizzo: mdadm --build device --chunk=X --level=Y --raid-devices=Z devices

Questo utilizzo e' simile a --create. la differenza e' che utilizzando --build verra' creato un array senza superblock. Con questi array (senza superblock) non vi e' differenza tra creazione e successivo assemblamento.

I livelli di raid possibili sono raid0 , linear, multipath, oppure faulty, e' possibile utilizzare un sinonimo qualunque. Tutti i devices saranno avviati una volta completi.

CREATE MODE

utilizzo: mdadm --create device --chunk=X --level=Y --raid-devices=Z devices

Questo utilizzo inizializza un array, associa i devices necessari ed infine attiva il device

Se l'opzione --auto viene utilizzata [spiegata in modo dettagliato nella sezione assemble] , allora il device " md " sara' creato con un numero di device number, se necessario.

Quando un device viene aggiunto, esso viene controllato per vedere se contiene un raid superblock oppure un filesystem. Esso viene anche controllato per vedere se la variazione del device supera l'1% . Se una discrepanza viene rilevata, l'array non si avvia in modo automatico. L'utilizzo dell'opzione --run supera questa precauzione.

Per creare un array degradato ove alcuni device non sono presenti, occorre semplicemente inserire la parola " missing " al posto dell'indicazione del device. Questo indichera' a mdadm di lasciare uno spazio vuoto in corrispondenza del device non indicato. Nel caso di un array raid4 oppure di un array raid5 solo un device missing e' tollerato. Nel caso di un array raid6 sono tollerati anche due device " missing ". Nel caso di un raid1 e' possibile procedere alla creazione dell'array utilizzando un solo device, ed aggiungendo in seguito il device mancante.

All'atto della creazione di un raid5, mdadm creera' automaticamente un array degradato con uno "spare disk" di riserva. Questo poiche' la ricostruzione dell'array e' piu' veloce utilizzando il disk "spare" piuttosto che utilizzare le informazioni di parita'. Questa impostazione puo' essere sovrascritta utilizzando la sintassi --force

Durante la creazione di un array coi metadati versione 1 , un nome host e' richiesto espressamente. Se tale nome non e' indicato tramite l'opzione --name mdadm scegliera' un nome basato sull'ultima parte del nome del dispositivo. Cosi' se /dev/md3 verra' creato, allora il nome ottenuto sara' 3. Se si creera' un array del tipo /dev/home , allora il nome " home " sara' utilizzato.

Un nuovo array otterra' un UUID casuale a 128 bit unico. Se vi e' un'esigenza specifica, e' possibile scegliere un UUID per un determinato array utilizzando l'opzione --uuid= . Rendiamo noto che creare due array i quali utilizzano il medesimo UUID e' una ricetta per un sicuro disastro. L'utilizzo dell'opzione --uuid= e' prioritaria sull'eventuale opzione --homehost=

Le opzioni generali valide con --create sono:

--run forza l'avvio di un determinato array , anche se alcuni device risultano in uso.

--readonly avvia un array in modalita' sola lettura. non ancora pienamente supportato.

OPZIONI VALIDE PER manage

utilizzo: mdadm device options... devices...

Questa possibilita' consente l'utilizzo di singoli dispositivi in un determinato array. Sono possibili diversi comandi con una unica linea di comando. Per esempio:

```
mdadm /dev/md0 -f /dev/hda1 -r /dev/hda1 -a /dev/hda1
come prima cose il dispositivo /dev/hda1 viene marcato come " fault "
nell'array /dev/md0 successivamente tale dispositivo sara' rimosso,
in ultima battuta tale dispositivo sara' aggiunto all'array /dev/md0
come spare
```

OPZIONI VALIDE PER misc

utilizzo: mdadm options ... devices ...

La modalita' MISC prevede un numero di operazioni distinte per operare su un singolo device. Tali operazioni sono:

--query Il device viene esaminato per capire se si tratta di un array md oppure di un device utilizzato da un device md. Tali informazioni vengono riportate.

--detail Il device deve essere un device md valido ed attivo. mdadm mostrera' una descrizione dettagliata dell'array. Le opzioni --brief oppure --scan genereranno un output che e' possibile inserire all'interno del file mdadm.conf . Lo status exit di mdadm sara' 0 qualora mdadm non riesca ad ottenere le informazioni sul device (devices). Qualora l'opzione --test venga utilizzata ecco che i valori di exit status diventano i seguenti:

- 0 l'array funziona correttamente
- 1 l'array ha un device non funzionante
- 2 l'array ha diversi devices non funzionanti e non e' utilizzabile
- 4 vi e' un errore durante il processo di acquisizione delle informazioni

--examine Il device dovrebbe essere un componente di un array md. mdadm leggerà il superbblock del device e mostrerà il contenuto. Se e' impartita l'opzione --brief oppure --scan allora i componenti di un array sono mostrati assieme, e l'output e' utilizzabile come contenuto per il file mdadm.conf
Utilizzare l'opzione --scan senza indicare alcun device, causerà la scansione di tutti i device presenti nel file di configurazione.

--stop L'array md sarà disattivato a patto di non essere in uso.

--runforza l'avvio di un array anche parzialmente assemblato.

--readonly con tale opzione un array md viene posto in modalità sola lettura.

--readwrite con questa opzione, un array precedentemente posto in modalità sola lettura, viene riportato in modalità lettura e scrittura.

--scan per tutte le operazioni ad eccezione di --examine, --scan farà in modo che le operazioni saranno applicate ad ogni array listato nel file /proc/mdstat. Se le opzioni --scan e --examine vengono utilizzate assieme ecco che ogni array elencato nel file /proc/mdstat viene esaminato.

OPZIONI VALIDE PER monitor

utilizzo: mdadm --monitor options ... devices

Questo utilizzo indica a mdadm di effettuare un controllo periodico su un certo numero di array md, e di relazionare su ogni avvenimento. mdadm non uscirà da solo una volta deciso quale array controllare, così solitamente questa opzione viene utilizzata in background (sullo sfondo) .

mdadm ha la capacità di trasportare un disk spare da un array ad un altro array, a patto che entrambi gli array appartengano allo stesso spare-group e che l'array di destinazione non preveda espressamente un suo disk spare .

Se ogni device e' indicato tramite linea di comando, mdadm monitorerà solamente tali device. Altrimenti tutti gli array elencati nel file di configurazione saranno monitorati. Successivamente se l'opzione --scan viene impartita, allora ogni device presente nel file /proc/mdstat sarà monitorato.

Il risultato del controllo di un array e' un evento. Tali eventi possono essere passati ad un programma separato (se specificato) e possono anche essere inoltrati per posta elettronica.

Quando un evento viene passato ad un programma, tale programma viene eseguito una volta per ogni evento, ed accetta 2 o 3 argomenti da linea di comando. Il primo di tali argomenti e' il nome dell'evento (vedi schema qui sotto) . Il secondo argomento e' il nome del device md interessato, ed il terzo e' il nome di un ipotetico array connesso, ad esempio un device guasto (failed) .

Se l'opzione --scan e' impartita, allora un programma oppure un indirizzo di posta elettronica devono essere indicati da linea di comando oppure dal file di configurazione . Se nulla viene indicato, allora mdadm semplicemente non monitorizza nulla. Senza l'opzione --scan , mdadm continuerà il controllo fino a che troverà qualcosa da controllare. Qualora nessun indirizzo di posta elettronica venga indicato, tutti gli output saranno diretti allo standard output.

Gli eventi possibili sono:
(lasco in inglese, vista la
mia scarsa qualita' di
traduttore)

DeviceDisappeared

An md array which previously was configured appearsto no longer be configured. (syslog priority: Critical)

If mdadm was told to monitor an array which is RAID0 or Linear, then it will report DeviceDisappeared with the extra information Wrong-Level. This is because RAID0 and Linear do not support the device-failed, hot-spare and resync operations which are monitored.

RebuildStarted

An md array started reconstruction. (syslog priority: Warning)

RebuildNN

Where NN is 20, 40, 60, or 80, this indicates that rebuild has passed that many percentage of the total. (syslog priority: Warning)

RebuildFinished

An md array that was rebuilding, isn't any more, either because it finished normally or was aborted. (syslog priority: Warning)

Fail An active component device of an array has been marked as faulty. (syslog priority: Critical)

FailSpare

A spare component device which was being rebuilt to replace a faulty device has failed. (syslog priority: Critical)

SpareActive

A spare component device which was being rebuilt to replace a faulty device has been successfully rebuilt and has been made active. (syslog priority: Info)

NewArray

A new md array has been detected in the /proc/mdstat file. (syslog priority: Info)

DegradedArray

A newly noticed array appears to be degraded. This message

is not generated when mdadm notices a drivefailure which causes degradation, but only when mdadm notices that an array is degraded when it first sees the array. (syslog priority: Critical)

MoveSpare

A spare drive has been moved from one array in a spare-group to another to allow a failed drive to be replaced. (syslog priority: Info)

SparesMissing

If mdadm has been told, via the config file, that an array should have a certain number of spare devices, and mdadm detects that it has fewer than this number when it first sees the array, it will report a SparesMissing message. (syslog priority: Warning)

TestMessage

An array was found at startup, and the --test flag was given. (syslog priority: Info)

Soltanto Fail, Failspare, DegradedArray, SparesMissing, e testMessage generano messaggi di posta elettronica da inviare. Tutti gli eventi causano l'avvio di un programma. Tale programma e' avviato con due o tre argomenti. Tali argomenti sono: il nome dell'evento, il nome dell'array ed eventualmente il nome di un secondo array.

Ogni evento e' associato ad un array, e possibilmente un secondo device. per Fail, FailSpare e SpareActive il secondo device e' l'elemento di importanza maggiore. Per Movespare il secondo device e' l'array da cui proviene il disk spare.

mdadm e' in grado di spostare gli spare disk da un array ad un altro array, a patto che entrambi gli array siano marcati con medesimo "spare-group" nel file di configurazione. Il nome "spare-group" puo' essere una stringa qualunque. E' necessaria quando solo quando gli spare group utilizzano label differenti.

Quando mdadm rileva che un array all'interno di uno spare group ha un numero di device attivi inferiore al minimo e nessun spare disk e' presente, cerchera' in un'altro array del medesimo spare group ove tutti i device siano funzionanti se vi sia uno spare disk. mdadm cerchera' di rendere disponibile lo spare disk al primo array, quello col numero di device inferiore al minimo. Se lo spare disk viene rimosso con successo, ma non risulta possibile il suo inserimento nell'array differente, ecco che esso viene ricollocato al posto originario.

OPZIONI VALIDE PER grow

Il modo grow e' utilizzato per modificare la dimensione oppure la forma di un array attivo. Per questo, il kernel deve supportare le modifiche necessarie. Diversi tipi di "crescita" sono stati inseriti durante lo sviluppo del kernel 2.6, compresa la ristrutturazione del raid5 con diversi device attivi

Attualmente il supporto disponibile e' per :

- . cambiare l'attributo " size " per RAID1, RAID5, RAID6
- . incrementare gli attribut "raid-disks" per RAID1, RAID5, RAID6

. aggiungere o rimuovere una scrittura " intent bitmap " per ogni array che supporta tale operazione.

CAMBIAMENTO DI DIMENSIONI

Normalmente quando un array e' costruito, la "dimensione" deriva dalla grandezza dei drive (device). Se tutti i device di un array (uno alla volta) vengono sostituiti con device maggiormente capienti, allora si potrebbe avere un array con maggior spazio disponibile. In questa situazione cambiare " size " utilizzando il modo " grow " rendera' disponibile lo spazio aggiuntivo. Se lo spazio viene aggiunto in questa maniera, un processo " resync " partira' per fare in modo che le nuove parti dell'array siano sincronizzate.

Notare che lo spazio extra guadagnato non viene automaticamente assegnato al file system. Il filesystem necessita di una sua procedura apposita per utilizzare il nuovo spazio extra.

CAMBIAMENTO LIVELLO DI RAID

Un raid1 puo' lavorare con ogni numero di device superiore a 1, (anche se 1 device solo non e' molto usato). potrebbero esserci occasioni ove si voglia aumentare o diminuire il numero dei devices attivi.

Quando riduci il numero dei devices in un raid1, lo slot che viene rimosso da un array sara' vacante. Cioe' quei devices presenti in quegli slot devono essere marcati " fail " e rimossi.

Quando il numero dei device viene aumentato, ogni " hot spare " presente viene attivato immediatamente.

Aumentare il numero dei device in un raid5 richiede molto piu' sforzo. Ogni blocco dell'array deve essere letto e scritto in una nuova posizione. A partire dalla versione 2.6.17, il kernel linux puo' farlo con sicurezza, compresi riavvii e riscritture interrotte.

Quando inizia il trasferimento dei dati verso un raid5, non e' possibile avere la certezza di sopravvivere ad un eventuale crash. Per fornire una adeguata sicurezza, mdadm disabilita la scrittura verso questo array, ed i dati presenti in quella sezione vengono replicati. Tale copia e' normalmente posizionata in un file separato specificato tramite l'opzione --backup-file. Se tale opzione viene utilizzata ed il sistema raid5 collassa nel periodo critico, lo stesso file puo' essere passato ad " --assemble " per recuperare il backup e ricostruire l'array.

BITMAP CHANGES

Un " write intentat bitmap " puo' essere aggiunto oppure rimosso da un array attivo. bitmap interni oppure bitmap locati in file separati possono essere aggiunti. Notare che se un bitmap locato in un file separato viene aggiunto, e tale file bitmap si trova all'interno del raid affetto, il sistema andra' in stallo. Il bitmap deve essere necessariamente in un file system differente.

OPZIONI VALIDE PER incremental mode

utilizzo: mdadm --incremental [--run] [--quiet] component-device

utilizzo: mdadm --incremental --rebuild

utilizzo: mdadm --incremental --run --scan

Tale modalita' e' progettata per essere utilizzata in unione ad un device " discovery system ". Quando tale device viene individuato in un sistema, esso puo' essere passato a mdadm --incremental per essere aggiunto in modo condizionale all'array appropriato.

mdadm esegue un numero di test per determinare se il device e' parte di un array, e di quale array fa' parte. Se un array adeguato viene trovato, oppure puo' essere creato, mdadm associa il device a tale array e lo avvia.

Note that mdadm will only add devices to an array which were previously working (active or spare) parts of that array. It does not currently support automatic inclusion of a new drive as a spare in some array.

mdadm richiede che un bug presente nel kernel 2.6.19 sia corretto. Fortunatamente tale bug fu risolto nel kernel 2-6-20. in alternativa e' possibile applicare la patch presente nei sorgenti di mdadm. Se mdadm rileva che tale bug e' ancora presente, esso abortira' ogni tentativo di utilizzare --incremental .

I test che mdadm esegue sono:

- + Il device e' menzionato in mdadm.conf? In caso positivo esso e' presente all'interno di una linea DEVICE , nel file mdadm.conf. Se la linea DEVICE non e' presente allora come impostazione predefinita mdadm accetta tutti i device. Nella medesima maniera se DEVICE contiene una " word partition " particolare, allora ogni device viene accettato.
- + Il device deve avere un md superblock valido. Se una particolare versione di metadata e' richiesta tramite --metadata oppure -e allora solo quel tipo di metadata viene consentito, altrimenti mdadm cerca ogni possibile versione di metadata. Se nessun tipo di md metadata viene rilevato, allora il device viene scartato.
- + I metadati di un array non corrispondono alle aspettative? I metadati possono incontrarsi in due maniere: l'array e' listato all'interno di mdadm.conf, (tramite UUID, tramite nome, tramite lista device oppure tramite minor-number), oppure l'array e' creato tramite un nome host specifico. Tale nome host puo' essere ottenuto tramite il file mdadm.conf oppure direttamente dalla linea di comando. Se mdadm non e' in grado di identificare l'array come appartenente all'host corrente, allora il device sara' scartato.
- + mdadm mantiene una lista degli array attivi in /var/run/mdadm/map oppure in /var/run/mdadm.map se la posizione precedentemente indicata non dovesse esistere. Se non esiste un array corrispondente ai metadati sul nuovo dispositivo, allora mdadm deve scegliere un nome dispositivo ed un numero di unita'. Tutto questo sulla base di un qualsiasi nome presente in mdadm.conf, oppure di ogni informazione presente nei metadati. Se un numero di unita' viene suggerito, esso viene usato. altrimenti un numero di unita' libero viene utilizzato. Solitamente mdadm cerca di creare un array partizionabile, tuttavia se la linea CREATE in mdadm.conf suggerisce di creare un array non partizionabile, questa indicazione verra' seguita.
- + Una volta che un array e' creato o rilevato e il (i) device

aggiunto, mdadm deve decidere se l'array e' pronto per essere avviato. mdadm paragonera' il numero dei devices disponibili (non spare) col numero dei device riportato dai metadati. Se il numero dei devices e' sufficiente allora l'array viene avviato. Qualora fosse indispensabile avviare comunque un array anche se tale array non ha a disposizione tutti i devices necessari, allora e' possibile utilizzare l'opzione --run . Se l'array e' un raid1 occorre che almeno un device sia utilizzabile. Se e' un raid5 allora sara' avviato appena sara' disponibile il numero minimo di devices utili. Si noti che nessuno di questi due approcci e' l'ideale. If it is can be known that all device discovery has completed, then mdadm -IRs can be run which will try to start all arrays that are being incrementally assembled. They are started in "read-auto" mode in which they are read-only until the first write request. This means that no metadata updates are made and no attempt at resync or recovery happens. Further devices that are found before the first write can still be added safely.

ESEMPI:

```
mdadm --query /dev/name-of-device
```

mdadm cerchera' tutte le informazione relative a /dev/nome_device , soprattutto cerchera' di determinare se e' parte di un array.

```
mdadm --assemble --scan
```

questo comando assemblera' ed avviaera' tutti gli array elencati nel file di configurazione standard. Tale comando e' solitamente presente negli script di avvio.

```
mdadm --stop --scan
```

questo comando eseguirà fermera' tutti gli array che possono essere fermati (cioe' quelli non correntemente in uso). E' presente di solito negli script di arresto dei raid .

```
mdadm --follow --scan --delay=120
```

Se (e solo se) e' presente un indirizzo di posta elettronica nel file di configurazione, allora tutti gli array vengono controllati ogni due minuti .

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/hd[ac]1
```

Crea /dev/md0 come un raid 1 costituito da /dev/hda1 e /dev/hdc1 .

```
echo 'DEVICE /dev/hd*[0-9] /dev/sd*[0-9]' > mdadm.conf  
mdadm --detail --scan >> mdadm.conf
```

Verra' creato un prototipo di file di configurazione il cui contenuto contiene una descrizione di tutti gli array i cui device risultano essere periferiche ide oppure scsi . Tale file dovrebbe essere riesaminato prima di essere utilizzato.

```
echo 'DEVICE /dev/hd[a-z] /dev/sd*[a-z]' > mdadm.conf  
mdadm --examine --scan --config=mdadm.conf >> mdadm.conf
```

Verranno cercati gli array che e' possibile avviare, la ricerca avviene su devices ide e scsi (non su partizioni), e le informazioni ricavate vengono riportate nel file di configurazione. E' molto probabile che tale file contenga un eccesso di informazioni, soprattutto in riferimento ai device [devices = entries] . Tale file deve essere rivisto ed aggiustato nel suo contenuto prima del reale utilizzo.

```
mdadm --examine --brief --scan --config=partitions
mdadm -Ebsc partitions
```

Crea una lista de device basandosi sulla lettura di /proc/partitions , esamina quanto trovato alla ricerca di un raid superbloc e stampa una lista di quanto trovato.

```
mdadm -Ac partitions -m 0 /dev/md0
```

Esamina tutte le partizioni e tutti i devices elencati in /proc/partition ed assembla /dev/md0 con un raid superbloc il cui " minor namber " e' 0

```
mdadm --monitor --scan --daemonise > /var/run/mdadm
```

Se il file di configurazione contiene un indirizzo di posta elettronica, allora mdadm viene avviato sullo sfondo [in background] in modalita' controllo per tutti gli array. Il numero identificativo del processo [pid] viene scritto in /var/run/mdadm .

```
mdadm -Iq /dev/somedevice
```

Cerca di aggregare i dispositivi appena rilevati all'interno di un array, cercando il piu' appropriato.

```
mdadm --incremental --rebuild --run --scan
```

Ricostruisce la mappa a partire da ogni array corrente, ed avvia ogni array avviabile .

```
mdadm --create --help
```

Mostra un messaggio di aiuto riferito all'opzione --create .

```
mdadm --config --help
```

Mostra un messaggio di errore riferito all'opzione --config

```
mdadm --help
```

Mostra un messaggio di aiuto generalizzato.

FILES

/proc/mdstat

Se viene utilizzato /proc/filesystem, allora /proc/mdstat elenca tutti i device md attivi, riportando le informazioni degli stessi.

mdadm utilizza questo sistema per trovare gli array quando l'opzione --scan viene utilizzata in modo MISC , e controlla il processo di ricostruzione degli array nel modo MONITOR

/etc/mdadm.conf

Il file di configurazione elenca i devices che possono essere controllati alla ricerca un md superbblock, ed ottenere informazioni identificative [es. UUID] riguardo i tipi conosciuti di array md . Vedere mdadm.conf(5) per ulteriori dettagli.

/var/run/mdadm/map

Quando il modo --incremental e' utilizzato, questo file ottiene un elenco degli array correttamente creati. Se /var/run/mdadm non esiste " come directory " allora /var/run/mdadm.map e' utilizzato nella medesima maniera.

NOMI DEI DEVICES

I devices elencati nella directory /dev/ possono assumere diversi formati. mdadm utilizza un formato " standard " proprio , che usa come guida durante la creazione dei " files devices " tramite l'opzione --auto .

I nomi standard per gli array non partizionabili [l'unico tipo di array disponibile nei kernel 2.4 e precedenti] sono:

/dev/mdNN
/dev/md/NN

dove NN indica un numero. Il nome standard per gli array partizionabili [disponibili a partire dai kernel 2.6] e' uno dei seguenti:

/dev/md/dNN
/dev/md_dNN

Il numero delle partizioni deve essere indicato aggiungendo " pMM " . Esempio: /dev/md/dlp2

NOTA

mdadm era precedentemente conosciuto come mdctl .

mdadm e' completamente distaccato dal pacchetto raidtools, e per il suo funzionamento non necessita del file /etc/raidtab .

VEDI ANCHE

<http://ostenfeld.dk/~jakob/Software-RAID.HOWTO/>

L'ultima versione di mdadm sara' sempre disponibile presso:

<http://www.kernel.org/pub/linux/utils/raid/mdadm/>

NOTIZIE AGGIUNTIVE

TIPI DI RAID [<http://it.wikipedia.org/wiki/RAID>]

- RAID 0 somma dei volumi. Un numeri di hard disk vengono visti come un hard disk unico.
- RAID 1 mirror. I dati su di un hard disk vengono duplicati esattamente su di un altro hard disk.
- RAID 2 perfezionamento del RAID 1
- RAID 3 perfezionamento del RAID 1 a cui e' aggiunto un hard disk dedicato alla parita'.
- RAID 4 divisione dei dati a livello di blocco, con un hard disk dedicato alla parita'.
- RAID 5 I dati e tutti gli algoritmi per il calcolo della parita' vengono equamente divisi tra gli hard disk. E' il tipo di raid piu' diffuso.
- RADI 6 perfezionamento del RAID 5 . Qui i dati di parita' vengono distribuiti due volte su tutti gli hard disk

link

<http://userlocal.com/articles/raid1-slackware-12.php>
<http://www.slacky.eu/forum/viewtopic.php?p=192610>
<http://pumpump.blogspot.com/2007/07/installing-slackware-12-on-linux.html>
<http://web.mit.edu/rhel-doc/OldFiles/4/RH-DOCS/rhel-ig-s390-multi-it-4/s1-s390info-raid.html>
<http://lamb.homeip.net/wp/>
http://www.grolug.org/mirrors/informatica_Masini/MyLinuxse57.html

SOSTITUZIONE DI UN HARD DISK DANNEGGIATO IN UN SISTEMA LINUX RAID 1

Nell'esempio seguente opereremo su di un sistema avente due hard disk, ognuno dei quali partizionato in due zone, in questa maniera:

```
/dev/sda1 e /dev/sdb1 costituiscono l'array in RAID1 /dev/md0  
/dev/sda2 e /dev/sdb2 costituiscono l'array in RAID1 /dev/md1
```

Il disco sdb si è guastato e deve essere sostituito. Occorre rimuoverlo operando nella seguente maniera:

```
mdadm manage /dev/md0 fail /dev/sdb1  
mdadm manage /dev/md0 remove /dev/sdb1
```

Abbiamo marcato come " fail " il device /dev/sdb1 e lo abbiamo rimosso dall'array. ripetiamo il procedimento sull'altro device

```
mdadm manage /dev/md1 fail /dev/sdb2  
mdadm manage /dev/md1 remove /dev/sdb2
```

Adesso si dovrà spegnere il computer e sostituire l'hard disk. Installato il nuovo disco riaccendete il pc e loggati come root andranno ricreate sul nuovo sdb le stesse partizioni presenti su sda (e' preferibile utilizzare hard disk della medesima capacita') :

```
sfdisk -d /dev/sda | sfdisk /dev/sdb  
fdisk -l (per controllare che le partizioni siano uguali)
```

Aggiungiamo /dev/sdb1 a /dev/md0 e /dev/sdb2 a /dev/md1

```
mdadm manage /dev/md0 add /dev/sdb1
```

ripetiamo il procedimento per /dev/md1

```
mdadm manage /dev/md1 add /dev/sdb2
```

Abbiamo terminato. Il sistema raid e' nuovamente operativo.

ESEMPI DI UTILIZZO DI MDADM

```
mdadm --create --verbose /dev/md0 --level=1 --raid-devices=2 /dev/sda /dev/sdb
```

crea il raid livello 1 utilizzando gli hard disk /dev/sda e /dev/sdb
utilizzandoli
appieno. Quando dedichiamo un intero hard disk al raid ecco che non occorre
indicare
la partizione [/dev/sda e NON /dev/sda1]

```
mdadm --detail /dev/md0
```

analizza il raid /dev/md0 e ne mostra lo stato a video.

```
mdadm -A /dev/md0 /dev/sda /dev/sdb
```

avvia il raid md0 senza appoggiarsi al file di configurazione /etc/mdadm.conf

```
mdadm -S /dev/md0
```

arresta il raid md0

```
mdadm /dev/md0 --add /dev/sdc
```

aggiunge un hard disk al raid md0

```
mdadm /dev/md0 --fail /dev/sda --remove /dev/sda
```

marca l'hard disk /dev/sda [appartenente al raid md0] come guasto
e lo rimuove dal raid

```
mdadm -C /dev/md0 --level=multipath --raid-devices=4 /dev/sda1 /dev/sdb1  
/dev/sdc1 /dev/sdd1
```

crea il raid multipath md0 utilizzando la prima partizione dei primi 4 hard
disk .

```
mdadm /dev/md0 -f /dev/sda1
```

marca come " faulty " [guasto] l'hard disk /dev/sda1

```
mdadm /dev/md0 -r /dev/sda1
```

rimuove dall'array il device /dev/sda1

```
mdadm /dev/md0 -a /dev/sda1
```

aggiune il device /dev/sda1 all'array

```
mdadm --create --verbose /dev/md1 --level=1 --raid-devices=2 missing /dev/hdb1
```

creiamo il raid /dev/md0 specificando per il momento un solo hard disk.
E' una procedura usuale quando migriamo un sistema installato su un hard disk
singolo e lo trasportiamo su un sistema raid.

```
mdadm --brief --detail --verbose /dev/md1
```

Informazioni sull'array comprese le partizioni fisiche che formano il metadvice
.

APPENDICE 1

inserisco anche l'ipotetico file rc.mdadm il cui compito e' monitorare il
funzionamento dei dispositivi " md "

```
#!/bin/sh
# Script to control mdadm raid array monitoring
# Written by John Jenkins mrgoblin@sastk.org
# $Id: rc.mdadm 9 2007-09-17 21:40:09Z mrgoblin $
#
# Please Note: /etc/rc.d/rc.local is likely the best place to call
# this script from. If you do call this at start-up please remember
# to also add it to your shutdown scripts or you will be left with
# a stale PID file. Using the "fstop" parameter will remove this.
#

# Set some variables
MDPID=""
PIDFILE="/var/run/mdadm"
CONFIG="/etc/mdadm.conf"
PROG="/sbin/mdadm"

# you may list each device seperated by spaces
DEVICES="/dev/md[0-9]"

# Sanity Checks
if [ ! -f ${CONFIG} -o ! -x ${PROG} ]; then
    echo 'Either your mdadm binary or config file is missing! '
    echo 'Is Mdadm installed? '
    echo
    exit 1
fi

# Get current PID
if [ -f /var/run/mdadm ];then
    MDPID=$(cat /var/run/mdadm)
fi

# The functions
start() {
    if [ "X${MDPID}" = "X" ];then
        ${PROG} --monitor --daemonise "${DEVICES}" > ${PIDFILE}
    else
        echo "Mdadm Monitor already running :"
```

```

        exit 1
    fi
}
stop() {
    if [ "X${MDPID}" = "X" ];then
        echo "Mdadm Monitor not running :"
        exit 1
    else
        echo -n "Stopping Mdadm monitor :"
        kill -15 ${MDPID} && rm ${PIDFILE} && echo " Success " || echo "
Failed to stop
Mdadm Monitor "
        unset MDPID
    fi
}
status() {
    if [ "X${MDPID}" = "X" ];then
        echo "Mdadm Monitor not running :"
        Size)/ {
            ${PROG} -D ${DEVICES} |awk '/(^\/dev|State\ :|Raid\ Level|Array
            print $0 }'
            echo
        else
            echo "Mdadm Monitor running on PID ${MDPID} :"
            Size)/ {
                ${PROG} -D ${DEVICES} |awk '/(^\/dev|State\ :|Raid\ Level|Array
                print $0 }'
                echo
            fi
        }
}
forced_stop() {
    echo "Forcibly stopping Mmdadm Monitor :"
    /bin/killall -9 mdadm
    if [ -f "${PIDFILE}" ]; then
        rm -f "${PIDFILE}"
    fi
    unset MDPID
}

# Parse input
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    fstop)
        forced_stop
        ;;
    status)
        status
        ;;
    *)
        echo "*** Usage: $0 {start|stop|fstop|restart|status}"
        exit 1
esac

```

exit 0

\$Id: rc.mdadm 9 2007-09-17 21:40:09Z mrgoblin \$
EOF

#####

Nota del traduttore:

mi accorgo di non avere fatto un lavoro decente. cio' e' dovuto alla poca dimestichezza con l'inglese tecnico. Su usenete non ho trovato una traduzione di mdadm, per cui in perfetto spirito open source, l'ho scritta io !!
spero possa servire a qualcuno ed invito tutti ovviamente a migliorare qst scritto.