

RAID Software : Proteggere i dati con l'aiuto del kernel (2 di 5)



Nel precedente articolo sono state introdotte le diverse tipologie di RAID ed i concetti di parità per la gestione della ridondanza.

Di seguito viene affrontata nel dettaglio la creazione e l'amministrazione dei RAID attraverso il software Linux *mdadm* per configurare il sistema in modo che gestisca i diversi tipi di ridondanza sui dischi disponibili.

RAID software con Linux

Per poter gestire via software i RAID, è necessario che il kernel utilizzato disponga del supporto "Multiple devices driver support" insieme al "RAID Support" ed a tutti i moduli relativi ai livelli di RAID che si vogliono utilizzare. Nei kernel precompilati che vengono installati dalle più diffuse distribuzioni questi moduli sono compresi, quindi è necessario porre attenzione solo in caso di kernel compilati "a mano".

Il modulo principale che si occupa della gestione dei RAID è denominato "md" (Multiple Disk) che è un alias di "md-mod". Per verificare che il kernel installato sulla macchina utilizzata abbia questo supporto è sufficiente provare a caricare il modulo :

```
$ modprobe md
```

Se il comando non restituisce errori, all'interno del filesystem virtuale /proc verrà creato un file denominato mdstat contenente le informazioni sullo stato della controller RAID software :

```
$ cat /proc/mdstat
Personalities :
unused devices: <none>
```

Lo stato del RAID, non avendo effettuato alcuna configurazione, è nullo, ma l'output ricevuto conferma che il sistema supporta il tipo di operazioni che vogliamo effettuare.

A questo punto andranno create le devices, i dispositivi virtuali associati ai RAID. Generalmente in Linux, tali devices vengono nominate con mdX dove X è un numero che partendo da 0 identifica tutti gli array gestiti dal sistema.

Creare i RAID con mdadm

In Linux originariamente, la gestione del RAID software era affidata ad una suite di programmi denominata raidtools che comprendeva una serie di eseguibili attraverso i quali era possibile creare, modificare ed amministrare i RAID software. Nelle recenti distribuzioni il pacchetto raidtools è stato rimpiazzato dal programma mdadm. Oltre a contenere notevoli migliorie, esso presenta l'indubbio vantaggio di consentire la gestione totale dei RAID di sistema attraverso un unico comando.

mdadm è incluso come pacchetto binario in tutte le distribuzioni recenti, mentre i sorgenti sono disponibili a questo indirizzo : <http://www.cse.unsw.edu.au/~neilb/source/mdadm/> .

Prima di cominciare è necessario avere ben chiaro ciò che si vuole ottenere. Tenendo come riferimento i pregi ed i difetti di ciascun livello illustrati poco sopra bisogna capire quale livello si adatta di più all'obiettivo che si vuole raggiungere.

Se ad esempio, la sicurezza dei dati è un fattore secondario e si necessita di performance e quanto

più spazio possibile, sarà sensato creare un RAID 0, se invece i dati registrati sono di importanza critica, allora converrà optare per un RAID 1 a discapito delle performance, oppure per un RAID 5 nel caso in cui i dischi utilizzabili siano tre o più e si voglia aumentare la velocità di accesso ai dati. Una volta fatta la scelta sul livello da implementare, è necessario scegliere se si vuole mettere sotto RAID l'intero sistema, compresa quindi la partizione root (ossia "/"), oppure solo alcune partizioni. Tratteremo successivamente la messa in RAID della partizione root, gli esempi illustrati riguardano l'implementazione su partizioni diverse da quella principale. Nel caso particolare sono state utilizzate le partizioni hda5, hda6 ed hda7 tutte di 250 MegaByte. Avrebbe più senso utilizzare partizioni che risiedano su dischi differenti, ed in ambienti di produzione questo è vivamente consigliato, ma per comprendere i meccanismi di funzionamento di mdadm questo tipo di situazione è più che sufficiente. Vincolante è la dimensione delle partizioni che deve essere quantomeno simile per tutte quelle impiegate.

Al comando mdadm, andranno passati come parametri il nome della device che si vorrà creare (ad esempio /dev/md0), il numero di dischi impiegati nell'array ed infine le partizioni che ne faranno parte.

RAID 0

Come primo esempio verrà creato un RAID 0 (striping) con le due partizioni hda5 ed hda6. Il comando che andrà eseguito, sarà il seguente :

```
$ mdadm --create /dev/md0 -a --level=0 --raid-disks=2 /dev/hda5 /dev/hda6
mdadm: array /dev/md0 started.
```

il messaggio "mdadm: array /dev/md0 started." indica che il processo di creazione del RAID 0 è stato avviato con successo. Ciò è verificabile consultando il contenuto del file /proc/mdstat :

```
$ cat /proc/mdstat
Personalities : [raid0]
md0 : active raid0 hda6[1] hda5[0]
      498688 blocks 64k chunks
```

```
unused devices: <none>
```

L'output mostra come tra le "Personalities" (i tipi di raid disponibili) sia presente il RAID 0 e che la controller software gestisce la device md0 che poggia su di un Raid0 formato da hda5 e hda6 (nell'ordine sono i dischi con identificativo 0 e 1).

La device md0 è composta da 498.688 blocchi (ciascuno da 1024 byte), per un totale di circa 500 Mb che equivale alla somma delle dimensioni dei nostri dischi.

L'ultimo valore mostrato è la "Chunk size" che riguarda la dimensione dei blocchi minima registrabile su ciascun disco dell'array. Ad esempio se va registrato un dato da 128k, il sistema registrerà 64k sul primo disco ed i rimanenti 64 sul secondo. Nel caso del Raid1 la chunk size non viene indicata in quanto i dati registrati non devono essere divisi per i dischi ma replicati così come sono su ciascuno di questi. Nel RAID 5 la chunk size è la dimensione del blocco di parità.

RAID 1

Nell'ambito della creazione di un RAID 1 il comando non cambierà di molto, l'unico parametro differente sarà "level" per il quale andrà indicato 1, il numero corrispondente al RAID "striping" :

```
$ mdadm --create /dev/md0 --level=1 --raid-disks=2 /dev/hda5 /dev/hda6
mdadm: array /dev/md0 started.
```

Anche in questo caso è possibile seguire lo stato del nuovo RAID creato :

```
$ cat /proc/mdstat
Personalities : [raid0] [raid1]
md0 : active raid1 hda6[1] hda5[0]
      249344 blocks [2/2] [UU]
```

```
[=====>...] resync = 86.0% (216000/249344) finish=0.0min  
speed=6156K/sec
```

```
unused devices: <none>
```

Si nota dall'output ottenuto che la device `md0` è in stato "resync". Perché a differenza del RAID 0, unico RAID che non prevede la replicazione dei dati, i due dischi devono sincronizzarsi, processo per il quale, a seconda della grandezza relativa alle partizioni impiegate, è necessario del tempo. Terminato il "resync", il contenuto di `mdstat` sarà il seguente :

```
$ cat /proc/mdstat  
Personalities : [raid0] [raid1]  
md0 : active raid1 hda6[1] hda5[0]  
      249344 blocks [2/2] [UU]
```

```
unused devices: <none>
```

Adesso il RAID 1 è completamente attivo. L'output rispetto alle operazioni precedenti è leggermente differente, infatti oltre alle informazioni relative al tipo di raid impiegato ed al numero di blocchi disponibili nella device `md0` esistono anche due voci che indicano lo stato dei dischi che fanno parte del RAID. La voce `[2/2]` indica con il primo numero il totale dei dischi che fanno parte dell'array ed il secondo il numero di device attive nell'array. La voce `[UU]` indica invece che entrambi i dischi sono in stato "Up". Come vedremo in seguito, se uno dei dischi entra in stato "Failure" la lettera "U" verrà rimpiazzata dal carattere "_".

RAID 4

La creazione di un Raid4 implica che le partizioni impiegate debbano essere tre o più. Una di queste andrà dichiarata come "spare" e conterrà le informazioni di parità :

```
$ mdadm --create /dev/md0 --level=4 --raid-disks=2 /dev/hda5 /dev/hda6 --spare-  
disks=1 /dev/hda  
mdadm: array /dev/md0 started.
```

Osservando attraverso il filesystem `/proc` lo stato del RAID, si nota come le partizioni `hda6` ed `hda5` risultano entrambe in stato UP, mentre è in corso la sincronizzazione della partizione `hda7`, che presenta la dicitura "(S)" che sta ad indicare come questa sia la partizione di spare :

```
$ cat /proc/mdstat  
Personalities : [raid0] [raid1] [raid5]  
md0 : active raid5 hda7[2](S) hda6[1] hda5[0]  
      249344 blocks level 4, 64k chunk, algorithm 0 [2/2] [UU]  
      [=====>...] resync = 88.1% (220368/249344) finish=0.0min  
speed=6485K/sec
```

```
unused devices: <none>
```

Quando l'operazione di sincronizzazione verrà completata lo stato del RAID sarà il seguente :

```
$ cat /proc/mdstat  
Personalities : [raid0] [raid1] [raid5]  
md0 : active raid5 hda7[2](S) hda6[1] hda5[0]  
      249344 blocks level 4, 64k chunk, algorithm 0 [2/2] [UU]
```

```
unused devices: <none>
```

RAID 5

Anche nel caso di un Raid5 le partizioni impiegate devono essere tre o più di tre :

```
$ mdadm --create /dev/md0 --level=5 --raid-disks=3 /dev/hda5 /dev/hda6 /dev/hda7
```

```
mdadm: array /dev/md0 started.
```

A differenza delle precedenti operazioni di “resync”, la controller software effettua un “recovery”. Questo perché il Raid5 non replica i dati in maniera lineare tra i dischi impiegati ma li divide gestendone la parità :

```
$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5]
md0 : active raid5 hda7[3] hda6[1] hda5[0]
      498688 blocks level 5, 64k chunk, algorithm 2 [3/2] [UU_]
      [=====>....]  recovery = 84.0% (209920/249344) finish=0.1min
      speed=3463K/sec
```

```
unused devices: <none>
```

Terminata la fase di “recovery” lo stato del Raid5 sarà il seguente :

```
$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5]
md0 : active raid5 hda7[2] hda6[1] hda5[0]
      498688 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]
```

```
unused devices: <none>
```

I blocchi impiegati in questo livello sono 498688, corrispondenti a circa 500 Mb, come nell’esempio del Raid0, con la [shttp://www.google.it/ola](http://www.google.it/ola) differenza che le partizioni impiegate in questo caso sono tre e che quindi il totale di spazio disponibile equivale a due terzi della somma delle dimensioni delle tre partizioni.

La “chunk size” è la stessa, mentre viene indicato il tipo di algoritmo utilizzato dalla controller software che è 2 ed equivale al default. Se approfondendo i metodi di gestione della parità (che non verranno trattati nell’articolo) si volessero scegliere altri tipi di algoritmo implementabili, basterà in fase di creazione dell’array indicare con il parametro “-parity” il tipo di algoritmo desiderato.

SET FAULTY o come simulare la sostituzione di un disco :

Prima di procedere con la creazione del filesystem sulla device md0 creata, verrà simulato il crash di un disco all’interno del Raid5 appena creato, in modo da capire come agire in questo tipo di situazioni.

Supponiamo infatti che anziché essere tre partizioni locali, queste siano divise in tre dischi distinti. Forziamo il malfunzionamento della prima partizione dell’array, hda5 :

```
$ mdadm /dev/md0 --set-faulty /dev/hda5
mdadm: set /dev/hda5 faulty in /dev/md0
```

Il RAID 5 è ancora attivo, ma sta funzionando con solo due partizioni su tre. In caso di rottura di un’ulteriore disco, cesserebbe di funzionare :

```
$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5]
md0 : active raid5 hda7[2] hda6[1] hda5[3](F)
      498688 blocks level 5, 64k chunk, algorithm 2 [3/2] [_UU]
```

```
unused devices: <none>
```

[_UU] indica che i dischi disponibili sono due su tre ed il primo di questi non sta funzionando. Procediamo quindi con la rimozione dall’array e successivamente con la rimozione fisica dalla macchina :

```
$ mdadm /dev/md0 --removehttp://www.google.it/ /dev/hda5
mdadm: hot removed /dev/hda5
```

A questo punto, una volta collegato il nuovo disco, questo andrà reintrodotta nell'array :

```
$ mdadm /dev/md0 --add /dev/hda5
mdadm: hot added /dev/hda5
```

La controller di sistema inizierà ad effettuare il “recovery” del nuovo disco introdotto ed al termine di questa operazione, l'array sarà nuovamente attivo :

```
$ cat /proc/mdstat
Personalities : [raid0] [raid1] [raid5]
md0 : active raid5 hda5[0] hda7[2] hda6[1]
      498688 blocks level 5, 64k chunk, algorithm 2 [3/3] [UUU]

unused devices: <none>
```

[UUU] indica infatti che i tre dischi sono di nuovo attivi.

Registrare la struttura del RAID

Durante la creazione del RAID mdadm ha registrato su ogni disco facente parte dell'array il “Persistent SuperBlock”. Quest'area, posta all'inizio del disco, memorizza le informazioni di appartenenza del disco all'interno del RAID. Questo consente al kernel di rilevare la struttura del RAID in fase di boot senza doversi leggere le informazioni da un file di configurazione ed ovviamente facilita la possibilità di far risiedere la partizione root su RAID.

Può succedere però che in caso di rottura di uno dei dischi si debba ricostruire la struttura senza le informazioni presenti nel “Persistent Superblock”.

Per ovviare a questo inconveniente è consigliabile registrare le informazioni del RAID creato all'interno del file */etc/mdadm.conf*, che verrà consultato solo all'occorrenza.

I semplici passi da seguire sono questi :

```
$ echo 'DEVICE /dev/hda[567]' > /etc/mdadm/mdadm.conf
$ mdadm --examine --scan --config=mdadm.conf >> /etc/mdadm/mdadm.conf
```

A questo punto il file *mdadm.conf* conterrà quanto segue :

```
$ cat /etc/mdadm/mdadm.conf
DEVICE /dev/hda[567]
ARRAY /dev/md0 level=raid5 num-devices=3
UUID=59e81780:c30962ed:c4aa02bd:4dd9dcae
```

Ovviamente il parametro “DEVICE” rispecchia gli esempi proposti sinora, se si impiegheranno dischi diversi questi andranno indicati con le stesse modalità. Se nell'array che si è costruito vengono utilizzate ad esempio le partizioni *hda1*, *hda2* e *hdb1* e *hdb2* e *hdc3*, il parametro “DEVICE” sarà il seguente :

```
DEVICE /dev/hd[ab][12] /dev/hdc3
```

Creare un filesystem sulla device /dev/md0

L'ultimo passo da effettuare prima di cominciare ad utilizzare l'array per la registrazione di dati è quello di creare un filesystem sulla device */dev/md0* :

```
$ mke2fs -j /dev/md0
mke2fs 1.39-WIP (31-Dec-2005)
Etichetta del filesystem=
Tipo S0: Linuxhttp://www.google.it/
Dimensione blocco=1024 (log=0)
Dimensione frammento=1024 (log=0)
124928 inode, 498688 blocchi
24934 blocchi (5.00%) riservati per l'utente root
Primo blocco dati=1
```

```
61 gruppi di blocchi
8192 blocchi per gruppo, 8192 frammenti per gruppo
2048 inode per gruppo
Backup del superblocco salvati nei blocchi:
    8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409
...
...
```

A questo punto, montando la device su una directory locale, il filesystem sarà disponibile per l'utilizzo nel sistema :

```
$ mount /dev/md0 /raid5
$ df -h /raid5/
Filesystem          Dimens. Usati Disp. Uso% Montato su
/dev/md0             472M  8,1M  440M   2% /raid5
```

Per rendere permanente la disponibilità del filesystem sarà necessario aggiungere la seguente riga in /etc/fstab :

```
/dev/md0          /raid5          ext3    defaults          0          0
```

Ad ogni riavvio del sistema, il RAID verrà montato nella directory /raid5.

Monitorare i RAID

In Debian, oltre ad avere un eseguibile richiamabile da linea di comando, il pacchetto mdadm contiene anche un demone che monitora lo stato dei RAID e segnala con una e-mail eventuali malfunzionamenti. La configurazione di questo demone è contenuta nel file /etc/default/mdadm :

```
START_DAEMON=true
MAIL_TO="root"
AUTOSTART=true
```

I parametri sono auto esplicativi, si può scegliere se avviare mdadm come un demone, a quale mail segnalare malfunzionamenti ed infine se il demone si deve avviare durante il boot di sistema.

Per le distribuzioni che non prevedono l'impiego di un demone per la gestione delle notifiche di malfunzionamento, è comunque possibile monitorare lo stato dei RAID passando il parametro "--monitor" al comando mdadm :

```
$ mdadm --monitor --mail=casella@dominio.com --delay=1800 /dev/md0
```

Conclusioni