Software-RAID HOWTO

Jakob Østergaard (*jakob@ostenfeld.dk*)

v. 0.90.7 19 gennaio 2000

Questo HOWTO spiega come usare il Software RAID con Linux. Esso si riferisce ad una specifica versione del Software RAID layer, ovvero lo 0.90 RAID layer scritto da Ingo Molnar e altri. Questo è il layer RAID che diventerà standard in Linux-2.4 ed è anche la versione usata dai kernel-2.2 in alcune distribuzioni. Il supporto al RAID 0.90 è disponibile sotto forma di patch a Linux-2.0 e Linux-2.2 ed è da molti considerato molto più stabile del vecchio supporto RAID presente in questi kernel.

Contents

1 Introduzione

Per una descrizione del vecchio layer RAI, quello che è standard nei kernel 2.0 e 2.2, date un occhiata all'eccellente HOWTO di Linas Vepstas (*linas@linas.org*) disponibile presso il Linux Documentation Project a *linuxdoc.org* .

Il sito principale per questo HOWTO è *http://ostenfeld.dk/~jakob/Software-RAID.HOWTO/* , dove saranno pubblicate le versioni aggiornate. L'HOWTO è scritto da Jakob Østergaard ed è basato su un gran numero di e-mail scambiate fra l'autore ed Ingo Molnar *(mingo@chiara.csoma.elte.hu)* – uno degli sviluppatori del supporto RAID –, la linux-raid mailing list *(linux-raid@vger.rutgers.edu)* varie altre persone.

La ragione per cui questo HOWTO è stato scritto è che, sebbene un Software-RAID HOWTO esistesse già, il precedente HOWTO descrive il vecchio Software RAID che si trova nei kernel 2.0 e 2.2 standard. Questo

HOWTO descrive invece l'uso del nuovo RAID che è stato sviluppato recentemente. Il nuovo RAID ha molte funzioni e caratteristiche non presenti nel vecchio RAID.

Se volete usare il nuovo RAID con i kernel 2.0 e 2.2, dovreste prelevare una patch per il vostro kernel da *ftp://ftp.[your-country-code].kernel.org/pub/linux/daemons/raid/alpha* , o più recentemente da *http://people.redhat.com/mingo/* . I kernel 2.2 standard non hanno un supporto diretto per il nuovo RAID descritto in questo HOWTO. Queste patch sono quindi necessarie. *Il supporto al vecchio RAID nei kernel 2.0 e 2.2 standard presenta dei bug e non presenta diverse importanti caratteristiche che sono invece presenti nel nuovo software RAID.*

Nel momento in cui viene scritto questo HOWTO, il supporto al nuovo RAID è stato inserito nell'albero di sviluppo dei kernel 2.3 e quindi sarà (molto probabilmente) presente nei Linux kernel 2.4, quando saranno disponibili. Fino ad allora ai kernel stabili devono essere applicate le patch manualmente.

Potreste usare i `-ac` rilasciati da Alan Cox, per il supporto RAID nei kernel 2.2. *Alcuni* di questi contengono il supporto al nuovo RAID e questo vi permetterà di non dovere applicare le patch al kernel da soli.

Alcune delle informazioni in questo HOWTO possono sembrare banali, se conoscete già bene il RAID. In questo caso potete saltare alcune parti.

1.1 Liberatoria

La liberatoria obbligatoria:

Sebbene il RAID sembri stabile a me e a molte altro persone, esso potrebbe non funzionare nel vostro caso. Se perdete tutti i vostri dati, il vostro lavoro, venite investiti da un camion o qualunque altra cosa, non è colpa né mia né degli sviluppatori. Attenzione, state usando il software RAID e queste informazioni a vostro rischio! Non c'è nessun tipo di garanzia che il software o queste informazioni siano corrette, né adatte ad un qualunque uso. Fate il salvataggio (back up) di tutti i vostri dati prima di fare esperimenti. Meglio essere sicuri che dispiaciuti.

Detto questo, devo dire che non ho mai avuto nessun problema di stabilità con il Software RAID, lo uso su alcune macchine senza alcun problema e non ho ancora visto altra gente con problemi di crolli casuali del sistema o instabilità causata dal RAID.

1.2 Nota alla versione italiana

Un po' di tempo fa sono stato un "utente" di questo HOWTO, che ho trovato particolarmente utile e chiaro nello spiegare come costruire un RAID array. Mi sono promesso allora di offrirne la versione italiana a tutti coloro che hanno difficoltà a "masticare" l'inglese. Il mio inglese è ben lungi dall'essere perfetto, ma spero di essere riuscito a mantenere la chiarezza dell'originale, per fare questo a volte mi sono dovuto allontanare dalla stretta traduzione letterale, spero di averlo fatto nel modo migliore. Ho cercato di "italianizzare" una buona parte dei termini tecnici, anche se per alcuni ho preferito mantenere l'originale inglese, perché i corrispondenti italiani risultavano essere fuorvianti e morfologicamente atroci. Per quel che riguarda i plurali inglesi, ho seguito la convenzione adottata fra gli altri da Umberto Eco e cioè di esprimerli con i corrispondenti singolari quando inseriti in un contesto in italiano. Sono certo che il mio lavoro è di gran lunga perfettibile, sia purché le mie conoscenze del RAID sono limitate e quindi posso aver espresso dei concetti in modo contorto se non poco corretto, sia perché di alcune parti ho avuto serie difficoltà di traduzione in italiano. Ogni eventuale suggerimento risulterà gradito: *alessio@arcetri.astro.it* , *tonno@stud.unipg.it* .

1.3 Cosa è necessario

Questo Howto assume che stiate usando uno degli ultimi kernel 2.2.x o 2.0.x con la corrispondente patch raid0145 e la versione 0.90 dei raidtools, o che stiate usando uno degli ultimi kernel 2.3 (versione > 2.3.46) o eventualmente il kernel 2.4. Sia le patch che i raidtools possono essere trovati presso *ftp://ftp.fi.kernel.org/pub/linux/daemons/raid/alpha* , ed in qualche caso presso *http://people.redhat.com/mingo/* . Le patch RAID, il pacchetto raidtools ed il kernel dovrebbero avere versioni il più possibile corrispondenti. A volte può essere necessario usare dei kernel più vecchi se le patch raid non sono disponibili per l'ultimo kernel.

2 Perché il RAID ?

Possono esserci molte buone ragioni per usare il RAID. Alcune sono: la capacità di combinare diversi dischi "reali" in un dispositivo "virtuale" più grande, l'aumento delle prestazioni e la ridondanza.

2.1 Aspetti Tecnici

Il RAID per Linux può funzionare sulla maggior parte dei dispositivi a blocchi. Non importa se usate dispositivi SCSI o IDE o una loro combinazione. Alcuni hanno usato il Network Block Device (NBD) con più o meno successo.

Assicuratevi che il bus (o i bus) a cui sono collegati i dischi siano abbastanza veloci. Non dovreste avere 14 dispositivi UW-SCSI su un bus UW, se ogni disco può fornire 10 MB/s e il bus può sostenere solo 40 MB/s. Inoltre, dovreste avere solo un disco per ogni bus IDE. Far lavorare i dischi come master e slave è tremendo per le prestazioni. L'IDE non lavora bene quando deve accedere a più di un disco per bus. Naturalmente, tutte le schede madri più recenti hanno due bus IDE, così che possiate montare due dischi in RAID senza dover acquistare degli ulteriori controller.

Il layer (strato) RAID non ha assolutamente nulla a che fare con il layer del filesystem. Potete mettere qualsiasi filesystem su un dispositivo RAID, così come su qualunque altro dispositivo a blocchi.

2.2 Termini

L'acronimo "RAID" indica il "Linux Software RAID". Questo HOWTO non copre nessuno degli aspetti dell'Hardware RAID.

Quando si descrivono i setup, è utile fare riferimento al numero dei dischi e alle loro dimensioni. Per tutto l'HOWTO la lettera **N** è usata per identificare il numero di dischi attivi nell'array (senza contare gli spare-disk). La lettera **S** è la dimensione del più piccolo disco dell'array, se non diversamente specificato. La lettera **P** è usata come indice di prestazione di un disco dell'array, in MB/s. Di solito, si assume che tutti i dischi dell'array siano ugualmente veloci, il che può non essere sempre vero.

Occorre notare che le parole "dispositivo" ("device") e "disco" ("disk") significano la stessa cosa. Di solito i dispositivi usati per costruire un dispositivo RAID sono delle partizioni sui dischi e non necessariamente interi dischi. Combinare diverse partizioni su un disco di solito non ha senso, così le parole dispositivi e dischi significano "partizioni su dischi diversi".

2.3 I livelli RAID

Viene qui presentato brevemente ciò che è supportato nelle Linux RAID patch. Alcune delle informazioni sono dei ragguagli assolutamente basilari sul RAID. Saltate pure questa parte se conoscete il RAID. Potete sempre tornare a leggerla se doveste avere dei problemi :).

Le patch RAID attuali per Linux supportano i seguenti livelli:

- **Linear mode**
  - Due o più dischi sono combinati in un dispositivo fisico. I dischi sono "appesi" (accodati) l'uno all'altro, così lo scrivere sul dispositivo RAID riempirà prima il disco 0, poi il disco 1 e così via. Non è obbligatorio che i dischi abbiano la stessa dimensione. Infatti, non importa affatto :)
  - Non c'è ridondanza in questo livello. Se un disco si danneggia, probabilmente tutti i dati saranno persi. Potreste comunque essere fortunati e recuperare alcuni dati, perché il filesysytem starà perdendo solo un grande blocco consecutivo ("chunk") di dati.
  - Le prestazioni in lettura e scrittura non miglioreranno per delle singole letture/scritture. Ma se diversi utenti utilizzano il dispositivo, potreste essere fortunati nel caso in cui un utente usi il primo disco e l'altro stia accedendo a dei file che stanno sul secondo disco. Se succede questo, dovreste accorgervi di un incremento di prestazioni.
- **RAID-0**
  - Detto anche modalità (mode) “stripe”. Come il linear mode, eccetto che le letture e le scritture sono fatte in parallelo sui dischi. I dischi dovrebbero essere approssimativamente della stessa dimensione. Siccome tutti gli accessi sono effettuati in parallelo, i dischi si dovrebbero riempire nella stessa misura. Se un disco è più grande degli altri, lo spazio eccedente è ancora usato nel dispositivo RAID, ma l'accesso avverrà solo sul disco più grande durante le scritture alla fine del dispositivo RAID. Questo va naturalmente a detrimento (deterioramento??) delle prestazioni.
  - Come per il linear mode, non c'è nessuna ridondanza in questo livello. Diversamente dal linear mode, non sarà possibile recuperare alcun dato se un disco si danneggia. Se un disco viene rimosso da un RAID-0, il RAID non perderà solo un grande consecutivo blocco di dati, esso sarà riempito con piccoli buchi lungo tutto il dispositivo. e2fsck non sarà probabilmente in grado di recuperare molto da questo dispositivo.
  - Le prestazioni in lettura e scrittura cresceranno, poiché le letture e scritture sono fatte in parallelo sui dischi. Questa è solitamente la ragione per cui si implementa un RAID-0. Se i bus che collegano i dischi sono abbastanza veloci, si dovrebbe ottenere qualcosa di molto vicino a N*P MB/s.
- **RAID-1**
  - Questa è la prima modalità che presenta ridondanza. Il RAID-1 può essere usato su due o più dischi con zero o più spare-disk. Questa modalità mantiene un'immagine (mirror) esatta del contenuto di un disco sugli altri. Naturalmente i dischi devono essere della stessa dimensione. Se un disco è più grande di un altro, il dispositivo RAID avrà la dimensione del disco più piccolo.
  - Se fino a N-1 dischi vengono rimossi (o si danneggiano), tutti i dati saranno ancora intatti. Se ci sono spare-disk disponibili e se il sistema (leggi SCSI driver o chipset IDE, ecc.) sopravvive al blocco del sistema, la ricostruzione del mirror inizierà immediatamente su uno degli spare-disk, dopo aver individuato il disco danneggiato.
  - Le prestazioni in scrittura sono piuttosto peggiori che su un singolo dispositivo, perché delle copie identiche dei dati scritti devono essere inviate a ogni disco dell'array. Le prestazioni in lettura sono *di solito* piuttosto cattive a causa di una troppo semplificata strategia di read-balancing nel codice RAID. Comunque, è stata implementata un strategia di read-balancing migliorata, che potrebbe diventare disponibile per le patch del Linux kernel 2.2 (chiedere sulla linux-kernel list), e sarà molto probabilmente nel supporto RAID del kernel 2.4.
- **RAID-4**

- Questo livello RAID non è usato molto spesso. Può essere usato su tre o più dischi. Invece di fare un immagine (mirror) completa delle informazioni, esso tiene delle informazioni di parità su un disco e scrive i dati sugli altri dischi in una maniera simile al RAID-0. Siccome un disco è riservato per le informazioni di parità, la dimensione dell'array sarà (N-1)*S, dove S rappresenta la dimensione del più piccolo disco dell'array. Così come nel RAID-1, i dischi dovrebbero essere della stessa dimensione, altrimenti il valore S nella formula precedente sarà la dimensione del più piccolo disco dell'array.
- Se un disco si danneggia, le informazioni di parità possono essere utilizzate per ricostruire tutti i dati. Se si danneggiano due dischi tutti i dati saranno persi.
- La ragione per cui questo livello non è usato spesso è che l'informazione di parità è tenuta su un disco. Quindi questa informazione deve essere aggiornata *ogni* volta uno degli altri dischi viene scritto. Quindi, il disco che contiene l'informazione di parità diventa un collo di bottiglia, se esso non è molto più veloce degli altri dischi. Comunque, se vi accade di avere molti dischi lenti ed uno molto veloce, questo livello RAID può essere molto utile.

- **RAID-5**
  - Questa è forse la più utile modalità RAID quando di desidera combinare un gran numero di dischi e mantenere ancora una certa ridondanza. Il RAID-5 può essere usato su tre o più dischi, con zero o più spare-disk. Il dispositivo RAID-5 che viene fuori avrà la dimensione (N-1)*S, come nel RAID-4. La grande differenza fra il RAID-5 ed il RAID-4 è che le informazioni di parità sono distribuite in modo uguale fra i dischi di cui è composto l'array, evitando così il collo di bottiglia che si creava nel RAID-4.
  - Se uno dei dischi si danneggia, tutti i dati saranno ancora intatti, grazie alle informazioni di parità. Se degli spare-disk sono disponibili, la ricostruzione inizierà immediatamente dopo il guasto del dispositivo. Se due dischi si danneggiano simultaneamente, tutti i dati saranno persi. Il RAID-5 può sopravvivere al danneggiamento di un disco, ma non a quello di due o più.
  - Sia le prestazioni in scrittura che in lettura migliorano, ma è difficile predire di quanto.

2.3.1 Spare disks

Gli spare disks sono dischi che non fanno parte dell'array RAID fino a che uno dei dischi attivi smette di funzionare. Quando il guasto di un disco viene rilevato, questo dispositivo viene marcato come "cattivo" (bad) e la ricostruzione viene immediatamente iniziata su uno degli spare-disk a disposizione.

Quindi, gli spare-disk aggiungono un'utile extra sicurezza specialmente ai sistemi RAID-5. Ci si può permettere di far lavorare il sistema per un po', con un dispositivo guasto, poiché tutta la ridondanza è conservata per mezzo degli spare-disk.

Non si può essere sicuri che un sistema sopravviva al guasto di un disco. Il RAID layer dovrebbe gestire i guasti ai dischi piuttosto bene, ma i driver SCSI potrebbero crollare sulla gestione degli errori, o il chipset IDE potrebbe bloccarsi, oppure una quantità di altre cose potrebbe accadere.

2.4 Fare lo Swap su RAID

Non c'è nessuna ragione nell'usare il RAID per questioni di prestazioni dello swap. Il kernel stesso può creare delle stripe facendo lo swap su più dispositivi, se solo gli date la stessa priorità nel file fstab.

Un fstab ben fatto si presenta così:

```
/dev/sda2       swap           swap    defaults,pri=1   0 0
/dev/sdb2       swap           swap    defaults,pri=1   0 0
```

```
/dev/sdc2       swap            swap    defaults,pri=1   0 0
/dev/sdd2       swap            swap    defaults,pri=1   0 0
/dev/sde2       swap            swap    defaults,pri=1   0 0
/dev/sdf2       swap            swap    defaults,pri=1   0 0
/dev/sdg2       swap            swap    defaults,pri=1   0 0
```

Questa configurazione permette alla macchina di fare lo swap in parallelo su sette dispositivi SCSI. Non c'è nessuna necessità del RAID, visto che questa è da sempre una caratteristica intrinseca del kernel.

Un'altra ragione per usare il RAID per lo swap è l'elevata disponibilità. Se per esempio si costruisce un sistema che fa il boot su un dispositivo RAID-1, il sistema dovrebbe essere in grado di sopravvivere al danneggiamento di un disco. Ma se il sistema stava facendo lo swap sul dispositivo guasto, ci sarà sicuramente un blocco. Fare lo swap su un dispositivo RAID-1 risolverebbe questo problema.

Ci sono state molte discussioni se fare lo swap fosse stabile sui dispositivi RAID. Questo è un dibattito continuo, che dipende per la maggior parte su altri aspetti del kernel. Nel momento in cui viene scritto questo HOWTO, sembra che fare lo swap su RAID sia perfettamente stabile, *eccetto* quando l'array è in fase di ricostruzione (per esempio dopo che un nuovo disco è stato inserito in un array danneggiato). Quando il kernel 2.4 sarà rilasciato, questa è una questione che sarà sistemata piuttosto rapidamente, ma fino ad allora, dovreste testare profondamente il sistema da soli fino a che sarete soddisfatti per la stabilità oppure concluderete che non volete fare lo swap su RAID.

Potete costruire un RAID in un file di swap su un filesystem sul vostro dispositivo RAID, o potete costruire un dispositivo RAID come una partizione di swap, come preferite. Come sempre, il dispositivo RAID è solo un dispositivo a blocchi.

3 Questioni hardware

Questo paragrafo tratta alcune delle questioni hardware implicate durante il lavoro di un software RAID.

3.1 Configurazione IDE

E' davvero possibile far lavorare un RAID su dei dischi IDE. Si possono realizzare anche eccellenti prestazioni. Infatti, il prezzo attuale dei dischi e dei controller IDE, rende l'IDE degno di considerazione, quando si costruisce un nuovo sistema RAID.

- **Stabilità "fisica":** I dischi IDE sono tradizionalmente stati sempre di peggiore qualità meccanica rispetto agli SCSI. Anche oggi, la garanzia sui dischi IDE è tipicamente di un anno , mentre è spesso da tre a cinque anni sui dischi SCSI. Sebbene non sia facile dire che i dischi IDE sono per definizione costruiti in modo peggiore, occorre stare attenti perché i dischi IDE di *alcune* marche *possono* guastarsi più spesso dei dischi SCSI simili. Comunque altre marche usano esattamente la stessa meccanica sia per i dischi SCSI sia per quelli IDE. Tutto questo ci porta alla conclusione che: Tutti i dischi si guastano prima o poi, quindi occorre essere preparati a questa evenienza.
- **Integrità dei dati:**All'inizio, l'IDE non poteva in alcun modo assicurare che i dati inviati sul bus IDE sarebbero stati gli stessi dati scritti sul disco. Questo era dovuto alla totale mancanza di parità, controllo (checksum), ecc. Con lo standard Ultra-DMA, i dischi IDE ora compiono un controllo (checksum) sui dati che ricevono e quindi diventa molto più difficile avere dei dati corrotti .
- **Prestazioni:** Non ho intenzione scrivere delle prestazioni dell'IDE. La vera storia in breve è:
  - I dischi IDE sono veloci (12 MB/s and oltre)

- L'IDE carica la Cpu più dello SCSI (ma chi se ne importa?)
- Usare solo**un** disco IDE per IDE bus, i dischi slave deteriorano le prestazioni

- **Sopravvivenza ai Guasti:** I driver IDE in genere sopravvivono al guasto di un dispositivo IDE. Il RAID layer marcherà il disco come guasto e se si sta lavorando su un sistema RAID livello 1 o superiore, la macchina dovrebbe lavorare ancora bene finché non viene arrestata per la manutenzione.

E' **molto** importante, che venga usato **un** solo disco IDE per IDE bus. Non solo due dischi rovinano le prestazioni, ma il guasto di un disco spesso comporta il blocco del bus, e quindi il blocco di tutti i dischi su quel bus. In un sistema RAID a prova di guasto (fault-tolerant) (livelli RAID 1,4,5), il guasto di un disco può essere gestito, ma il danneggiamento di due dischi (i due dischi sullo stesso bus che si bloccano a causa del guasto di uno dei due) renderà l'array inutilizzabile. Inoltre, quando il disco master su un bus si guasta, lo slave o il controller IDE possono risultare tragicamente confusi.

Esistono degli economici PCI IDE controller sul mercato. E' possibile acquistare due o quattro bus per circa 80$. Considerando il prezzo più basso dei dischi IDE rispetto agli SCSI, direi che un array di dischi IDE potrebbe essere veramente un'interessante soluzione se vi potete accontentare di avere a disposizione un numero relativamente basso (circa 8 probabilmente) di dischi da collegare al sistema (a meno che non abbiate molti slot PCI per collegare degli IDE controller).

L'IDE ha maggiori problemi di cablatura quando si creano grandi array. Anche se avete a disposizione abbastanza slot PCI, è difficile che possiate sistemare più di 8 dischi in un sistema continuando a mantenerlo funzionante senza corruzione dei dati (causato dalla lunghezza eccessiva dei cavi IDE).

3.2 Sostituzione di dischi "al volo" (Hot Swap)

Questa è stata una delle questioni maggiormente trattate sulla linux-kernel list per un po' di tempo. Sebbene la sostituzione "al volo" dei dischi sia supportata a qualche livello, non è ancora qualcosa che possa essere fatta facilmente.

3.2.1 Sostituzione "al volo" (Hot-swapping) dei dischi IDE

NO ! L'IDE non gestisce assolutamente la sostituzione "al volo". Di sicuro, può funzionare, se il driver IDE è compilato come modulo (possibile solo con i kernel della serie 2.2), e se lo ricaricate dopo avere sostituito il disco. Ma potreste anche finire con un controller IDE "fritto" e avere un tempo di fermo macchina molto maggiore che se aveste sostituito il drive con il sistema arrestato.

Il problema principale, eccetto le questioni elettriche che possono distruggere il vostro hardware, è che il bus IDE deve essere riesaminato dopo che i dischi sono stati sostituiti. Il driver IDE non può al momento farlo. Se il nuovo disco è al 100% uguale al vecchio (geometria, ecc.), *potrebbe* funzionare anche senza riesame del bus, ma in verità questo vuol dire camminare sulla lama del rasoio.

3.2.2 Sostituzione "al volo" (Hot-swapping) di dischi SCSI

Nemmeno il normale hardware SCSI offre la sostituzione "al volo". **Potrebbe** però funzionare comunque. Se il vostro SCSI driver supporta il riesame del bus e il collegamento e la rimozione di dispositivi, potreste essere in grado di sostituire al volo i dischi. Comunque, in un normale bus SCSI non si dovrebbero probabilmente scollegare i dispositivi mentre il sistema è ancora alimentato. Ma, di nuovo, potrebbe funzionare (e potreste anche finire con dell'hardware "fritto").

Lo SCSI layer **dovrebbe** sopravvivere se un disco cessa di funzionare, ma non tutti gli SCSI driver sono ancora in grado di gestire questo. Se il vostro driver SCSI si blocca quando un disco si guasta, il sistema crollerà con esso, tutto sommato il collegamento "al volo" non è poi così interessante.

3.2.3 Sostituzione "al volo" (Hot-swapping) con SCA

Con lo SCA, dovrebbe essere possibile collegare i dispositivi "al volo". Comunque, io non posseggo l'hardware necessario a provare questa funzione, non sono a conoscenza di nessuno che ci abbia provato, così non posso dare nessuna "ricetta" su come farlo.

Se volete "giocare" con questo, dovreste comunque conoscere abbastanza del funzionamento dello SCSI e del RAID. Così non scriverò qui qualcosa che non posso essere sicuro che funzioni, posso invece darvi alcuni indizi:

- Fate un grep per cercare **remove-single-device** in **linux/drivers/scsi/scsi.c**
- Date un occhiata a**raidhotremove** e **raidhotadd**

Non tutti i driver SCSI supportano il collegamento e la rimozione dei dispositivi. Nella serie dei kernel 2.2, almeno l'Adaptec 2940 ed il Symbios NCR53c8xx sembrano supportarlo, gli altri potrebbero e non potrebbero. Apprezzerei molto se qualcuno avesse ulteriori fatti da riportare sull'argomento...

4 RAID setup

4.1 General setup

Questo è quello di cui avete bisogno per qualunque livello RAID:

- Un kernel. Preferibilmente un kernel stabile della serie 2.2.x, oppure l'ultimo 2.0.x. (Se il kernel 2.4 fosse stato rilasciato quando leggerete questo documento, usate quest'ultimo)
- Le RAID patch. C'è di solito una patch disponibile per i kernel recenti (Se trovate un kernel 2.4, le patch sono già state incluse e quindi vi potete dimenticare di questo punto)
- I RAID tool.
- Pazienza, Pizza, e la vostra bevanda con caffeina preferita.

Tutto questo software può essere trovato presso `ftp://ftp.fi.kernel.org/pub/linux` I RAID tool e le patch sono nella subdirectory `daemons/raid/alpha` subdirectory. I kernel si possono trovare nella subdirectory `kernel`.

Applicate la patch al kernel, configuratelo per includere il supporto al livello RAID che volete usare. Compilatelo ed installatelo.

Poi, decompattate, configurate, compilate ed installate i RAID tool.

Ok, fatto. Se fate un reboot ora, dovreste avere un file di nome `/proc/mdstat`. Ricordate, questo file è vostro amico. Guardate cosa contiene facendo un `cat /proc/mdstat`. Esso dovrebbe dirvi che avete la corretta RAID personality (modalità RAID) registrata e che nessun dispositivo RAID è al momento attivo.

Create ora le partizioni che volete includere nel RAID array.

Da ora, analizziamo specificamente ogni modalità in maniera separata.

4.2 Linear mode

Ok, avete a disposizione due o più partizioni che non hanno necessariamente le stesse dimensioni (ma naturalmente potrebbero averle), volete a questo punto "appenderle" una all'altra.

Preparate il file `/etc/raidtab` per dare una descrizione del vostro sistema. Io ho preparato un raidtab per due dischi in linear mode, il file si presenta nel seguente modo:

```
raiddev /dev/md0
        raid-level      linear
        nr-raid-disks   2
        chunk-size      32
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Gli spare-disk non sono supportati in questa modalità. Se un disco si guasta, l'array si blocca con esso. Non ci sono informazioni da mettere su uno spare disk.

Probabilmente vi chiederete come mai abbia specificato una "`chunk-size`" qui, quando il linear-mode non fa altro che "appendere" i dischi in un grande array senza alcun parallelismo. Bene, avete perfettamente ragione, è strano. Mettete una qualche chunk size e non preoccupatevene ulteriormente.

Ok, creiamo l'array. Date il comando

```
  mkraid /dev/md0
```

Questo inizializzerà il vostro array, scriverà i persistent superblock e farà partire l'array.

Date un'occhiata a `/proc/mdstat`. Dovreste vedere che l'array sta lavorando.

Ora, potete creare un filesystem, come fareste su qualunque altro dispositivo, montarlo, includerlo in fstab e così via.

4.3 RAID-0

Avete due o più dischi approssimativamente della stessa dimensione e volete combinare le loro capacità di "memorizzazione" e combinare anche le loro prestazioni facendoci degli accessi in parallelo.

Preparate il file`/etc/raidtab` per descrivere la vostra configurazione. Un file raidtab di esempio appare così:

```
raiddev /dev/md0
        raid-level      0
        nr-raid-disks   2
        persistent-superblock 1
        chunk-size     4
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Come nel linear-mode, gli spare disk non sono supportati. Il RAID-0 non ha ridondanza, così quando un disco si guasta, l'array lo segue.

Ancora una volta date il comando

```
  mkraid /dev/md0
```

per inizializzare l'array. Questo dovrebbe inizializzare i superblocchi e far partire il dispositivo RAID. Date un occhiata a `/proc/mdstat` per vedere cosa sta succedendo. Dovreste vedere che il vostro dispositivo ora sta lavorando.

/dev/md0 ora è pronto per essere formattato, montato, usato e abusato.

4.4 RAID-1

Avete due o più dischi approssimativamente delle stesse dimensioni e volete che ognuno sia l'immagine (mirror) esatta dell'altro. Eventualmente potete avere più dischi, che volete tenere come spare-disk, che automaticamente diverranno parte dell'array se uno dei dischi attivi si guasta.

Preparate il file `/etc/raidtab` nel seguente modo:

```
raiddev /dev/md0
        raid-level      1
        nr-raid-disks   2
        nr-spare-disks  0
        chunk-size     4
        persistent-superblock 1
        device          /dev/sdb6
        raid-disk       0
        device          /dev/sdc5
        raid-disk       1
```

Se avete spare disks, potete aggiungerli alla fine della specifica del dispositivo nel modo seguente

```
        device          /dev/sdd5
        spare-disk      0
```

Ricordate di dichiarare la voce nr-spare-disks in modo corrispondente.

Ok, abbiamo preparato tutto per far partire il RAID. L'immagine (mirror) deve essere costruita, cioè il contenuto (che al momento non è importante, in quanto il device deve ancora essere formattato) dei due dischi deve essere sincronizzato.

Date il comando

```
  mkraid /dev/md0
```

per dare il via all'inizializzazione del mirror.

Controllate il file `/proc/mdstat` file. Dovrebbe dirvi che il dispositivo /dev/md0 è partito, che l'immagine (mirror) è in corso di ricostruzione e un ETA dello stato della ricostruzione.

Dovrebbe dirvi che il dispositivo /dev/md0 è partito, che l'immagine (mirror) è in corso di ricostruzione e un ETA dello stato della ricostruzione.

La ricostruzione è fatta utilizzando la larghezza di banda dell'I/O inutilizzata. Così, il sistema dovrebbe ancora essere piuttosto pronto a rispondere, sebbene gli hard disk led dovrebbero lampeggiare allegramente.

Provate a formattare il dispositivo, mentre la ricostruzione è in corso. Funzionerà. Potete anche montarlo ed usarlo mentre la ricostruzione è in corso. Naturalmente, se il disco sbagliato si rompe mentre la ricostruzione è in corso, non avete speranze.

4.5 RAID-4

Notate bene! Non ho mai fatto un test di un sistema del genere. Il setup seguente è una mia fondata congettura, ma nessuna sua parte è mai stata fatta girarare.

Avete tre o più dischi approssimativamente della stessa dimensione, uno dei dischi è significativamente più veloce degli altri e voi volete combinarli in un dispositivo più grande, mantenendo ancora delle informazioni di ridondanza. Eventualmente avete un certo numero di dischi che vorreste usare come spare-disk.

Preparate il file /etc/raidtab nel modo seguente:

```
raiddev /dev/md0
        raid-level      4
        nr-raid-disks   4
        nr-spare-disks  0
        persistent-superblock 1
        chunk-size      32
        device          /dev/sdb1
        raid-disk       0
        device          /dev/sdc1
        raid-disk       1
        device          /dev/sdd1
        raid-disk       2
        device          /dev/sde1
        raid-disk       3
```

Se disponessimo di spare disk, essi devono essere inseriti in un modo simile, seguendo le specifiche dei dischi RAID;

```
        device         /dev/sdf1
        spare-disk     0
```

come al solito.

Il vostro array può essere inizializzato con il comando

```
   mkraid /dev/md0
```

come al solito.

Dovreste dare un'occhiata alla sezione sulle opzioni speciali per mke2fs prima di formattare il dispositivo.

4.6 RAID-5

Avete tre o più dischi approssimativamente della stessa dimensione che volete combinare in un dispositivo più grande, mantenendo ancora un certo grado di ridondanza per la sicurezza dei dati. Eventualmente potete avere un certo numero di dischi da usare com spare disk, che non fanno parte dell'array fino a che un altro disco si guasta.

Se usate N dischi di cui il più piccolo ha dimensione S, la dimensione dell'intero array sarà (N-1)*S. Questo spazio è "perso" per le informazioni di parità (ridondanza). Quindi, se uno dei dischi si guasta i dati saranno ancora intatti. Ma se due dischi si guastano, tutti i dati andranno persi.

Preparate il file /etc/raidtab nel seguente modo:

```
raiddev /dev/md0
        raid-level      5
        nr-raid-disks   7
        nr-spare-disks  0
        persistent-superblock 1
        parity-algorithm        left-symmetric
        chunk-size      32
        device          /dev/sda3
        raid-disk       0
        device          /dev/sdb1
        raid-disk       1
        device          /dev/sdc1
        raid-disk       2
        device          /dev/sdd1
        raid-disk       3
        device          /dev/sde1
        raid-disk       4
        device          /dev/sdf1
        raid-disk       5
        device          /dev/sdg1
        raid-disk       6
```

Se disponete di spare disk, essi dovrebbero essere inseriti in una maniera simile, seguendo le specifiche dei dischi raid;

```
        device         /dev/sdh1
        spare-disk     0
```

e così via.

Una chunk size di 32 KB è un buon valore di default per la maggior parte dei filesystem di uso generale. L'array su cui la precedente raitab viene usata, è un dispositivo da 7 dischi da 6 GB = 36 GB (ricordando che (n-1)*s = (7-1)*6 = 36). Su di esso è costruito un ext2 filesystem con una dimensione del blocco da 4 KB. Potreste aumentare sia la chunk size dell'array sia la dimensione del blocco del filesystem se il vostro filesystem è o molto più grande o è costituito da file molto grandi.

Ok, abbiamo parlato abbastanza. Avete preparato la raidtab, andiamo a vedere se funziona. Date il comando

```
  mkraid /dev/md0
```

e state a vedere cosa succede. Se tutto è andato a buon fine i vostri dischi dovrebbero iniziare a lavorare come matti, iniziando la ricostruzione dell'array. Date un occhiata a `/proc/mdstat` per vedere cosa sta succedendo.

Se la creazione del dispositivo è avvenuta con successo, il processo di ricostruzione è iniziato a questo punto. Il contenuto del vostro array non è consistente finché questa fase di ricostruzione non è terminata. Comunque, l'array è completamente funzionanate (eccetto che per la gestione dei guasti naturalmente) e quindi potete formattarlo anche durante la fase di ricostruzione.

Date un occhiata al paragrafo sulle opzioni speciali di mke2fs prima di formattare l'array.

Date un occhiata al paragrafo sulle opzioni speciali di mke2fs prima di formattare l'array.

Ok, ora che avete il RAID device funzionante, potete sempre arrestarlo e farlo ripartire usando i comandi

```
raidstop /dev/md0
```

o

```
raidstart /dev/md0
```

Invece di mettere questi comandi nei file init e fare il reboot un fantastilione di volte per arrivare ad un sistema funzionante, continuate a leggere e capirete come poter far funzionare l'autorilevamento (autodetection).

4.7 Il Persistent Superblock

'Tanto tempo fa..." (TM), i raidtools avrebbero letto il vostro file `/etc/raidtab` e poi avrebbero inizializzato l'array. Comunque, questo avrebbe richiesto che il filesystem su cui risiedeva `/etc/raidtab` fosse montato. Questo risulta essere sfavorevole se avete intenzione di fare il boot da raid..

Inoltre il vecchio approccio portava a delle complicazioni quando si montavano i filesystem sui dispositivi RAID. Essi non potevano essere messi nel file `/etc/fstab` come al solito, ma avrebbero dovuto essere montati negli init script.

I persistent superblock risolvono questi problemi. Quando un array è inizializzato con l'opzione `persistent-superblock` nel file `/etc/raidtab` uno speciale superblock viene scritto all'inizio di tutti i dischi che compongono l'array. Questo permette al kernel di leggere la configurazione dei dispositivi RAID direttamente dai dischi che ne fanno parte, invece di ottenerla da qualche file di configurazione che potrebbe non essere disponibile in qualche momento.

Dovreste comunque mantenere un file `/etc/raidtab` file, consistente, poiché potreste aver bisogno di questo file per le successive ricostruzioni dell'array.

Il persistent superblock è obbligatorio se volete l'auto rilevamento (autodetection) dei vostri dispositivi RAID al boot del sistema. Tutto ciò è descritto nel paragrafo **Autorilevamento (autodetection)**.

4.8 Chunk size

La chunk-size necessita di una spiegazione. Voi non potete mai scrivere completamente in parallelo su una batteria di dischi. Se avete due dischi e volete scriverci un byte, dovreste scrivere quattro bit su ogni disco, effettivamente, ogni bit pari andrebbe sul disco 0 e gli altri sul disco 1. L'hardware non supporta tutto questo. Invece, noi scegliamo alcune chunk size, che definiamo come come la più piccola massa "atomica" di dati che possa essere scritta su un dispositivo. Una scrittura di 16KB con una chunk-size di 4KB, farà si che il primo e il terzo chunk da 4KB siano scritti sul primo disco ed il secondo e il quarto chunk siano scritti sul secondo disco, nel caso di un RAID-0 con due dischi. Quindi, per scritture di grosse quantità di dati, si può notare un miglioramento dall'usare dei chunk piuttosto grandi, mentre gli array che contengono principalmente piccoli files beneficieranno di una piccola dimensione dei chunk.

Le dimensioni dei chunk devono essere specificate per tutti i livelli RAID, incluso il linear-mode. Comunque la chunk-size non fa alcuna differenza per il linear-mode.

Per avere prestazioni ottimali, dovreste sperimentare con i valori, così come con la dimensione del blocco del filesystem che costruite sull'array.

L'argomento dell'opzione chunk-size in

```
/etc/raidtab
```

---

specifica la chunk-size in kilobyte. Così "4" significa "4 KB".

4.8.1 RAID-0

I dati sono scritti “quasi” in parallelo nei dischi dell’array. Effettivamente, i blocchi di dimensione `chunk-size` sono scritti in ogni disco serialmente.

Se specificate una chunk-size di 4 KB e scrivete 16 KB su un array di tre dischi, il sistema RAID scriverà 4 KB nei dischi 0, 1, 2, in parallelo, e poi i rimanenti 4 KB sul disco 0.

Una chunk-size di 32 KB è un punto di partenza ragionevole per la maggior parte degli array. Ma il valore ottimale dipende molto dal numero dei dischi costituenti l’array, dal contenuto del filesystem che ci viene messo sopra e da molti altri fattori. Sperimentate, per ottenere le migliori prestazioni.

4.8.2 RAID-1

Per le scritture la chunk-size non ha influenza sull’array, in quanto i dati devono essere scritti su tutti i dischi dell’array. Per le letture comunque, la chunk-size specifica quanti dati leggere serialmente dai dischi facenti parte dell’array. Poiché tutti i dischi attivi nell’array contengono la stessa informazione, le letture possono essere fatte in parallelo in modo simile al RAID-0.

4.8.3 RAID-4

Quando una scrittura è fatta su un array RAID-4, le informazioni dei parità devono sempre essere aggiornate sul disco di parità. La chunk-size è la dimensione del blocco di parità. Se un byte viene scritto su un array RAID-4, allora dei blocchi di dimensione `chunk-size` saranno letti dagli N-1 dischi, verrà calcolata l’informazione di parità e i blocchi di dimensione `chunk-size` saranno scritti nel disco di parità.

La chunk-size influenza le prestazioni in lettura nello stesso modo che nel RAID-0, visto che le letture da un array RAID-4 vengono effettuate nello stesso modo.

4.8.4 RAID-5

Sugli array RAID-5 la chunk-size ha esattamente lo stesso significato che nel RAID-4.

Una chunk-size ragionevole per un array RAID-5 è 128 KB, come sempre, potete sperimentare con essa.

Date anche un occhiata al paragrafo sulle opzioni speciali di mke2fs. Questo influenza le performance di un array RAID-5.

4.9 Opzioni per mke2fs

Esiste un’opzione speciale per formattare un dispositivo RAID-4 o RAID-5 con mke2fs. L’opzione `-R stride=nn` permetterà a mke2fs di piazzare meglio delle strutture dati specifiche dell’ext2 in modo intelligente sul dispositivo RAID.

Se la chunk-size è di 32 KB, ciò significa che blocchi da 32 KB di dati consecutivi saranno presenti su un disco. Se volessimo costruire un ext2 filesystem con una dimensione del blocco da 4 KB, si capisce che avremmo otto blocchi del filesystem per ogni chunk dell’array. Noi possiamo passare questa informazione all’utility mke2fs al momento di creare il filesystem:

```
mke2fs -b 4096 -R stride=8 /dev/md0
```

Le prestazioni dei RAID-{4,5} sono fortemente influenzate da questa opzione. Non sono sicuro di come l’opzione stride influenzi gli altri livelli RAID. Se qualcuno avesse maggiori informazioni su questo, per favore me le faccia avere.

La dimensione del blocco dell'ext2fs influenza *fortemente* le prestazioni del filesystem. Dovreste usare sempre la dimensione del blocco da 4 KB su ogni filesystem più grande di qualche centinaia di megabyte, a meno che non stiate lavorando con un gran numero di piccoli file.

4.10 Autorilevamento (Autodetection)

L'autorilevamento permette ai dispositivi RAID di essere automaticamente riconosciuti dal kernel al boot del sistema, subito dopo che il rilevamento solito delle partizioni è stato eseguito.

Tutto ciò richiede diverse cose:

1. Avete bisogno del supporto all'autorilevamanto (autodetection) nel kernel. Controllate
2. Dovreste aver creato i dispositivi RAID usando i persistent-superblock
3. Il tipo di partizioni dei dispositivi usati nel RAID deve essere impostato al valore **0xFD** (usate fdisk per impostare il tipo "fd")

NOTA: Siate certi che che il vostro RAID NON stia lavorando prima di cambiare il tipo della partizione. Usate `raidstop /dev/md0` per arrestare il dispositivo.

Se preparate tutto in modo congruente ai precedenti punti 1, 2 e 3, l'autorilevamento dovrebbe avvenire. Provate a fare un reboot. Quando il sistema riparte, fate un cat di `/proc/mdstat`che dovrebbe dirvi che il vostro dispositivo RAID sta funzionando .

Durante il boot, dovreste vedere dei messaggi simili a:

```
Oct 22 00:51:59 malthe kernel: SCSI device sdg: hdwr sector= 512
 bytes. Sectors= 12657717 [6180 MB] [6.2 GB]
Oct 22 00:51:59 malthe kernel: Partition check:
Oct 22 00:51:59 malthe kernel:  sda: sda1 sda2 sda3 sda4
Oct 22 00:51:59 malthe kernel:  sdb: sdb1 sdb2
Oct 22 00:51:59 malthe kernel:  sdc: sdc1 sdc2
Oct 22 00:51:59 malthe kernel:  sdd: sdd1 sdd2
Oct 22 00:51:59 malthe kernel:  sde: sde1 sde2
Oct 22 00:51:59 malthe kernel:  sdf: sdf1 sdf2
Oct 22 00:51:59 malthe kernel:  sdg: sdg1 sdg2
Oct 22 00:51:59 malthe kernel: autodetecting RAID arrays
Oct 22 00:51:59 malthe kernel: (read) sdb1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdb1,1>
Oct 22 00:51:59 malthe kernel: (read) sdc1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdc1,2>
Oct 22 00:51:59 malthe kernel: (read) sdd1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sdd1,3>
Oct 22 00:51:59 malthe kernel: (read) sde1's sb offset: 6199872
Oct 22 00:51:59 malthe kernel: bind<sde1,4>
Oct 22 00:51:59 malthe kernel: (read) sdf1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdf1,5>
Oct 22 00:51:59 malthe kernel: (read) sdg1's sb offset: 6205376
Oct 22 00:51:59 malthe kernel: bind<sdg1,6>
Oct 22 00:51:59 malthe kernel: autorunning md0
Oct 22 00:51:59 malthe kernel: running: <sdg1><sdf1><sde1><sdd1><sdc1><sdb1>
Oct 22 00:51:59 malthe kernel: now!
```

```
Oct 22 00:51:59 malthe kernel: md: md0: raid array is not clean --
  starting background reconstruction
```

Questo è quello che si ottiene dall'autorilevamento di un array RAID-5 che non è stato arrestato in modo pulito (per esempio se si è avuto un blocco della macchina). La ricostruzione viene iniziata automaticamente. Montare questo dispositivo è perfettamente sicuro, poiché la ricostruzione è trasparente e tutti i dati sono consistenti (è solo l'informazione di parità che è inconsistente - ma non c'è nessuna necessità di essa finché un dispositivo non si guasta).

I dispositivi che vengono avviati automaticamente sono anche arrestati automaticamente quando si ferma la macchina (shutdown). Non vi curate degli init script. Usate solo i device /dev/md come ogni altro device /dev/sd o /dev/hd.

Si, è veramente così semplice.

Potreste voler dare un occhiata ai vostri init-script per cercare i comandi raidstart/raidstop. Essi si trovano di solito negli init script della RedHat. Sono utilizzati solo per il vecchio RAID e non servono a niente nel nuovo RAID con l'autorilevamento. Potete semplicemente rimuovere queste linee e tutto continuerà a funzionare bene.

4.11 Fare il boot su RAID

Ci sono diversi modi per costruire un sistema che monti il suo root filesystem su un dispositivo RAID. Al momento solo l'installazione grafica di Linux RedHat 6.1 permette l'installazione diretta di un dispositivo RAID. E'comunque possibile realizzare la cosa.

L'ultima distribuzione ufficiale di lilo (Versione 21) non gestisce i dispositivi RAID e quindi il kernel non può essere caricato al boot dal dispositivo RAID. Se usate questa versione, il vostro `/boot` filesystem dovrà stare su un dispositivo non-RAID. Un modo per assicurarsi che che il sistema faccia il boot comunque, è quello di creare delle partizioni `/boot` uguali su tutti i dischi del vostro RAID, in questo modo il BIOS può sempre caricare i dati per esempio dal primo disco disponibile. Questo presume che non facciate il boot da un disco guasto nel vostro sistema..

Con la RedHat 6.1 una patch per lilo 21 è diventata disponibile, essa può gestire il `/boot` su un RAID-1. Notate che essa non funzionerà per qualunque altro livello, RAID-1 (mirroring) è il solo livello RAID supportato. Questa patch (`lilo.raid1`) può essere trovata presso `dist/redhat-6.1/SRPMS/SRPMS/lilo-0.21-10.src.rpm` su ogni RedHat mirror. La versione di lilo a cui viene applicata la patch accetterà la voce `boot=/dev/md0` nel file `lilo.conf` e renderà ogni ogni disco immagine (mirror) pronto per il boot.

Un altro modo per assicurare che il sistema riesca sempre a fare il boot è di creare un floppy di boot dopo che tutto il sistema è stato costruito. Se il disco sul quale il filesystem `/boot`risiede si blocca, è sempre possibile fare il boot da floppy.

4.12 Root filesystem su un RAID

Al fine di avere un sistema che faccia il boot su RAID, il root filesystem (/) deve essere montato su un dispositivo RAID. Due metodi per realizzare questo sono dati sotto. Visto che nessuna delle attuali distribuzioni (almeno di cui io sia a conoscenza) supporta l'installazione su un dispositivo RAID, questi metodi presumono che voi abbiate installato su una normale partizione e poi - quando l'installazione è finita - spostiate il contenuto del vostro root filesystem non-RAID sul nuovo dispositivo RAID.

4.12.1 Metodo 1

Questo metodo presume che voi abbiate degli spare-disk, che non fanno parte del sistema RAID che state configurando, su cui possiate installare il sistema operativo.

- Per primo, installate un normale sistema su questi extra dischi.
- Prendete il kernel che pensate di usare, prendete le raid patch e i raidtools e fate il boot del sistema con questo nuovo kernel con il supporto RAID. Siate sicuri che il supporto al RAID sia **nel** kernel, e non caricato come modulo.
- Ok, a questo punto dovreste configurare e creare il RAID che pensate di usare come root filesystem. Questa è una procedura standard descritta altrove in questo documento.
- Per essere sicuri che tutto è andato a buon fine, provate a fare il reboot del sistema e controllate se il nuovo RAID si attiva al boot. Dovrebbe.
- Create un filesystem sul nuovo array (usando `mke2fs`), e montatelo sotto /mnt/newroot
- Ora, copiate il contenuto del vostro attuale root filesystem (cioè lo spare-disk) nel nuovo root filesystem (cioè l'array). Ci sono molti modi per farlo, uno ad esempio è

  ```
  cd /
  find . -xdev | cpio -pm /mnt/newroot
  ```

- Dovreste modificare il file `/mnt/newroot/etc/fstab` per utilizzare il giusto dispositivo(il`/dev/md?` root device) per il root filesystem.
- Ora, smontiamo l'attuale `/boot` filesystem, e montiamo invece il dispositivo di boot su `/mnt/newroot/boot`. Questo è necessario per far lavorare correttamente lilo nel prossimo punto.
- Aggiornate `/mnt/newroot/etc/lilo.conf` per puntare al giusto dispositivo. Il dispositivo di boot deve ancora essere un normale disco (non un dispositivo RAID), ma il root device deve puntare al nuovo RAID. Quando l'avete fatto, date il comando

  ```
  lilo -r /mnt/newroot
  ```

  , LILO dovrebbe essere eseguito senza errori.
- Fate il reboot del sistema e guardate se tutto va come vi aspettate :)

Se fate tutto ciò con dischi IDE, siate sicuri di comunicare al BIOS che tutti i dischi sono di tipo "auto-detect", così che il BIOS permetterà alla vostra macchina di fare il boot, anche se un disco viene perso.

4.12.2 Metodo 2

Questo metodo necessita che voi usiate una raidtools/patch che include la direttiva failed-disk. Questa sarà la tools/patch per tutti i kernel verione 2.2.10 e successive.

Voi potete usare questo metodo **solo** un RAID di livello 1 e superiore. L'idea è quella di installare il sistema su un disco che è, di proposito, marcato come guasto all'interno del RAID, allora copiate il sistema sul RAID che lavorerà in modalità "degraded" e finalmente farete usare al RAID il non più necessario "install-disk", saltando la vecchia installazione, ma facendo lavorare il RAID in modalità "non degraded".

- Per prima cosa, installate un normale sistema su un disco (che diverrà successivamente parte del RAID). E' importante che questo disco (o partizione) non sia il più piccolo. Se lo fosse, non sarà possibile aggiungerlo al RAID successivamente!
- Poi, procuratevi un kernel, le patch, i raidtools ecc. ecc. Come al solito. Fate fare il boot al sistema con un nuovo kernel che abbia il supporto RAID di cui avete bisogno compilato nel kernel stesso.
- Ora, costruite il RAID con il vostro attuale root-device come `failed-disk` nel file `raidtab`. Non mettete il `failed-disk` come primo disco nel file`raidtab`, questo comporterebbe dei problemi all'avvio del RAID. Create il RAID e costruite un filesystem su di esso.
- Provate a fare un reboot e guardate se il RAID parte correttamente
- Copiate i file di sistema e riconfigurate il tutto per utilizzare il RAID come root-device, come descritto nel precedente paragrafo.
- Quando il sistema farà il boot correttamente dal RAID, potrete modificare il file `raidtab` per includervi il `failed-disk` come un`raid-disk`. Ora, fate `raidhotadd` con il vostro disco al RAID.
- Dovreste aver ottenuto un sistema che fa il boot da un "non-degraded" RAID.

4.13 Dovreste aver ottenuto un sistema che fa il boot da un "non-degraded" RAID.

Perché un kernel sia in grado di montare un root filesystem, il supporto per il dispositivo su cui il filesystem risiede deve essere presente nel kernel. Perciò, al fine di montare il root filesystem su un device RAID, il kernel *deve* avere il supporto RAID.

La maniera normale per essere certi che il kernel possa vedere il dispositivo RAID è semplicemente quella di compilare il kernel con tutto il supporto RAID necessario. Siate sicuri di aver compilato il supporto RAID *nel* kernel, e *non* come modulo caricabile. Il kernel non può caricare un modulo (dal root filesystem) finché il root filesystem non è montato.

Comunque, poiché RedHat-6.0 è distribuita con un kernel che ha il supporto per il nuovo RAID come modulo, qui verrà descritto come usare il kernel standard della RedHat-6.0, continuando a fare il boot del sistema da RAID.

4.13.1 Fare il boot con il RAID come modulo

Dovrete "istruire" LILO ad usare un RAM-disk per ottenere questo. Usate il comando `mkinitrd` per creare un ramdisk contenente tutti i moduli del kernel necessari a montare la partizione di root. Questo può essere fatto nel seguente modo:

```
mkinitrd --with=<module> <ramdisk name> <kernel>
```

Per esempio:

```
mkinitrd --with=raid5 raid-ramdisk 2.2.5-22
```

Questo ci rende sicuri che il modulo RAID specificato sia presente al boot quando il kernel monterà il root device.

4.14 Trabocchetti

Non dovete mai MAI **mai** ripartizionare i dischi che fanno parte di un array attivo. Se dovete alterare la tabella delle partizioni su un disco che fa parte di un RAID, arrestate l'array e poi ripartizionate.

E' facile mettere troppi dischi su un bus. Un normale Fast-Wide SCSI bus può sopportare 10 MB/s, che è meno di quanto alcuni dischi possano fare oggi. Mettere sei di questi dischi sul bus non vi darà di certo il guadagno di prestazioni che vi aspettate.

Un maggior numero di controller SCSI vi daranno delle prestazioni migliori, se i bus SCSI sono quasi saturati da dischi collegati. Non vedrete un incremento di prestazioni dall'uso di due 2940 con due vecchi dischi SCSI rispetto a far lavorare i due dischi su un solo controller.

Se vi dimenticate dell'opzione persistent-superblock, il vostro array non ripartirà volentieri dopo che è stato arrestato. Ricreate allora l'array con l'opzione abilitata nel file raidtab.

Se un RAID-5 fallisce la ricostruzione dopo che un disco è stato rimosso e riinserito, questo potrebbe dipendere dall'ordine dei dispositivi nel file raidtab. Provate a spostare la prima coppia "device ..." / "raid-disk ..." in fondo alla descrizione dell'array nel file raitab.

La maggior parte degli "error reports" che trovate sulla linux-kernel list, sono di persone che in qualche modo hanno sbagliato ad usare la giusta patch raid con la corrispondente versione dei raidtools. Siate sicuri di utilizzare 0.90 RAID, e che state usando i raidtools per esso.

5 Fare il test

Se pensate di usare il RAID per avere tolleranza ai guasti (fault-tolerance), potreste voler testare il vostro sistema per vedere se funziona veramente. Ora, come simulare un guasto?

In breve, non potete, eccetto forse che facendo passare un ascia infuocata attraverso il disco di cui volete simulare il guasto. Non potrete mai sapere cosa accadrà quando un disco cessa di funzionare. Esso potrebbe bloccare elettricamente tutto il bus a cui è collegato, rendendo tutti i dischi su quel bus inaccessibili. Sebbene non mi sia noto nessun accadimento del genere. Il disco potrebbe solo notificare un guasto di lettura/scrittura al layer SCSI/IDE, il che permetterà al layer RAID di gestire questa situazione in modo positivo. Questo è per fortuna il modo in cui le cose vanno di solito.

5.1 Simulare il malfunzionamento di un disco

Se volete simulare il guasto di un disco, scollegatelo. Dovreste farlo per mezzo del pulsante di **spegnimento**. Se siete interessati a testare se i vostri dati possano sopravvivere con un disco in meno del numero solito, non c'è problema nel fare l' "hot-plug cowboy" ora. Spegnete il sistema, scollegate il disco e fate un nuovo boot.

Controllate nel syslog e date un'occhiata a`/proc/mdstat` per vedere cosa sta facendo il RAID. Funziona?

Ricordate, voi **dovete** far girare il RAID-{1,4,5} sul vostro array per poter sopravvivere al guasto di un disco. Linear-raid o RAID-0 perdono tutto quando un disco si guasta.

Quando avrete di nuovo collegato il disco (con la macchina spenta, naturalmente, ricordate), potete aggiungere il "nuovo" dispositivo al RAID nuovamente, con il comando `raidhotadd`.

5.2 Simulare il danneggiamento dei dati

Il RAID (sia hardware- che software-), considera che se la scrittura su un disco non genera un errore, allora la scrittura è avvenuta correttamente. Allora, se il vostro disco danneggia i dati senza generare errori, tutti i vostri dati *saranno* corrotti. Questo naturalmente è molto spiacevole che accada, ma è possibile e porterebbe ad avere un filesystem corrotto.

Il RAID non può e non è progettato per controllare il danneggiamento dei dati sui supporti. Perciò non ha alcun senso il corrompere di proposito i dati (utilizzando `dd` per esempio) su un disco per vedere come il sistema RAID riesce a gestire questa situazione. E' molto probabile (a meno che si corrompa il RAID superblock) che il RAID layer non si accorga di niente riguardo al danneggamento, ma che il vostro filesystem sul dispositivo RAID sia danneggiato.

Questo è il modo in cui le cose si suppone che funzionino. Il RAID non è una garanzia per l'integrità dei dati, esso ci permette solo di conservare i nostri dati se un disco si blocca (con i livelli RAID maggiori od uguali ad 1, naturalmente).

6 Ricostruzione

Se avete già letto il resto di questo HOWTO, dovreste già avere un'idea di che cosa significhi la ricostruzione di un array danneggiato. Riassumendo:

- Spegnete il sistema
- Sostituite il disco guasto
- Accendete di nuovo il sistema
- Usate il comando `raidhotadd /dev/mdX /dev/sdX` per inserire nuovamente il disco nell'array
- Prendetevi un caffè mentre la ricostruzione automatica avviene

Ecco tutto.

Bene, di solito funziona così, a meno che voi siate sfortunati e il vostro RAID sia stato reso inutilizzabile perché più di un disco si è guastato. Questo può realmente succedere se un certo numero di dischi è collegato sullo stesso bus e uno dei dischi blocca il bus quando si guasta. Gli altri dischi, anche se non sono guasti, saranno irraggiungibili per il RAID layer, perché il bus è bloccato e quindi saranno marcati come danneggiati. Su un RAID-5 su cui è possibile sostituire un disco, il guasto di due o più di essi può essere fatale.

Il seguente paragrafo è la spiegazione che Martin Bene mi ha dato, e descrive un possibile recupero dal terrificate scenario mostrato sopra. Esso implica l'uso della direttiva `failed-disk` nel nostro file `/etc/raidtab`, così funzionerà solo con i kernel 2.2.10 e successivi.

6.1 Recupero dal malfunzionamento di più dischi

Lo scenario è:

- Un controller si blocca e mette due dischi offline nello stesso momento,
- Tutti i dischi su un bus SCSI non possono più essere raggiunti se un disco si blocca,
- Un cavo si sgancia...

In breve: abbastanza spesso si ha un guasto *temporaneo* di diversi dischi nello stesso momento; dopo di che i RAID superblock non sono più sincronizzati e voi non potete più inizializzare (init) il vostro RAID array.

Per prima cosa: riscrivete il RAID superblock con il comando `mkraid -force`

Al fine di farlo lavorare correttamente, avete bisogno di un file `/etc/raidtab` aggiornato - se esso non corrisponde **ESATTAMENTE** ai dispositivi e all'ordine dei dischi originali, non funzionerà.

Controllate il syslog prodotto cercando di far partire l'array, vedrete l'"event count" per ogni superblock; di solito è meglio lasciare fuori il disco con il più basso "event count", per esempio il più vecchio.

Se date il comando `mkraid` senza `failed-disk`, il processo di ricostruzione partirà immediatamente ed inizierà a ricostruire i blocchi di parità - il che non è necessariamente quello che volete in questo momento.

Con `failed-disk` potete specificare quali dischi volete che siano attivi e forse potete provare diverse combinazioni per ottenere i migliori risultati. BTW, monta il filesystem solo read-only durante queste prove... Questo metodo è stato usato da almeno due persone con cui sono in contatto.

7 Prestazioni

Questa parte contiene un certo numero di benchmark di un sistema reale che usa il software RAID.

I benchmark sono stati fatti con il programma `Bonnie` e ogni volta con file grandi due o più volte la dimensione della RAM fisica presente sulla macchina.

I benchmark misurano *solo* la larghezza di banda (bandwidth) in ingresso e in uscita su un solo grande file. Questa è una cosa interessante da conoscere, se siamo interessati al massimo indice di trasferimento (throughput) per scritture/letture di grandi quantità di dati. In ogni caso, questi numeri ci dicono poco delle prestazioni dell'array se esso fosse usato come un "news spool", un web-server, ecc. Teniamo sempre in mente, che i numeri dei benchmark sono il risultato dell'esecuzione di un programma "sintetico". Pochi dei programmi che appartengono alla vita reale fanno quello che fa `Bonnie` e, sebbene questi numeri di I/O siano interessanti da guardare, non sono indicativi delle prestazioni dei programmi reali. Non troppo almeno.

Per ora ho solo i risultati ottenuti sulla mia macchina. Il sistema è:

- Doppio Pentium Pro 150 MHz
- 256 MB RAM (60 MHz EDO)
- Tre IBM UltraStar 9ES 4.5 GB, SCSI U2W
- Adaptec 2940U2W
- Un IBM UltraStar 9ES 4.5 GB, SCSI UW
- Adaptec 2940 UW
- Kernel 2.2.7 con le RAID patch

I tre dischi U2W pianterebbero il controller U2W, and il disco UW ingolferebbe il controller UW.

Sembra impossibile instradare più di 30 MB/s di dati attraverso i bus SCSI del sistema usando o meno il RAID. Quello che credo è che, siccome il sistema è piuttosto vecchio, sia la banda passante della memoria a mancare, e questo limita quello che può essere inviato attraverso un controller SCSI.

Chunk size	Block size	Read KB/s	Write KB/s
4k	1k	19712	18035
4k	4k	34048	27061
8k	1k	19301	18091
8k	4k	33920	27118
16k	1k	19330	18179
16k	2k	28161	23682
16k	4k	33990	27229
32k	1k	19251	18194
32k	4k	34071	26976

Chunk size	Block size	Read KB/s	Write KB/s
32k	4k	33617	27215

7.1 RAID-0

La lettura è un **"Sequential block input"**, e **la scrittura** è un **"Sequential block output"**. La dimensione del file è stata di 1 GB per tutti i test. I test sono stati fatti in modalità singolo utente (single user). Il driver SCSI è stato configurato per non usare il "tagged command queuing".

a questo, sembra che la dimensione del chunk non faccia molta differenza. Comunque, la dimensione del blocco dell'ext2fs dovrebbe essere più grande possibile, quindi 4 KB (ovvero la dimensione della pagina) sui sistemi IA-32.

7.2 RAID-0 con TCQ

Questa volta il driver SCSI è stato configurato per utilizzare il "tagged command queuing", con una profondità della coda di 8. Per il resto tutto come prima.

Nessun altro test è stato fatto. TCQ sembra incrementare un poco le prestazioni in scrittura, ma in realtà non è che ci sia poi tutta questa differenza.

7.3 RAID-5

L'array è stato configurato per lavorare in modalità RAID-5, dei test simili ai precedenti sono stati fatti.

Ora, sia la dimensione del chunk che quella del blocco del filesystem sembrano effettivamente fare la differenza.

Chunk size	Block size	Read KB/s	Write KB/s
8k	1k	11090	6874
8k	4k	13474	12229
32k	1k	11442	8291
32k	2k	16089	10926
32k	4k	18724	12627

Chunk size	Block size	Read KB/s	Write KB/s
32k	1k	13753	11580
32k	4k	23432	22249

7.4 RAID-10

Un array RAID-10 è composto da “mirrored stripes”, ovvero, un array RAID-1 di due array RAID-0. La chunk-size è la dimensione del chunk sia dell’array RAID-1 che di quello RAID-0. Non ho compiuto dei test nel caso in cui le due dimensioni del chunk differiscano, per quanto sia perfettamente lecito.

Nessun altro test è stato fatto. La dimensione del file era 900MB, perché le quattro partizioni coinvolte erano da 500 MB ognuna, il che non permetteva di avere spazio per un file da 1 GB in questa configurazione (RAID-1 su due array da 1000MB).

8 Contributi

Le seguenti persone hanno contribuito alla creazione di questa documentazione:

- Ingo Molnar
- Jim Warren
- Louis Mandelstam
- Allan Noah
- Yasunori Taniike
- Martin Bene
- Bennett Todd
- Le persone della Linux-RAID mailing list
- Quelli che ho dimenticato, spiacente :)

Per favore inviate correzioni, suggerimenti ecc. all’autore. Questo è il solo modo in cui questo HOWTO può migliorare.