Come funziona Linux: Il backup !

Nona parte

di Giuseppe Zanetti

"E vedrai che ti lascerà...." dice una famosa canzone. Il salvataggio di copie di sicurezza (backup) dei propri file è un'operazione che spesso viene sottovalutata e considerata una inutile e dispendiosa perdita di tempo... almeno fino a quando non si rompe qualcosa.

Mi sono reso conto di quanto importante fosse tenere sempre una copia di "backup" aggiornata dei propri file su un supporto diverso dal disco in cui si lavora, quella volta in cui, dall'oggi al domani il mio computer non ne ha più voluto sapere di ripartire. A quei tempi non avevo ancora alle spalle una grandissima esperienza e avevo ingenuamente sottovalutato il problema del backup dei dati.

Per fortuna alla fine me la sono cavata abbastanza bene, in quanto il problema, seppur subdolo, non era così grave come sembrava in un primo momento. Il colpevole è letteralmente "saltato" fuori quando, dopo una mattinata di tentativi vani, stavo ormai per cedere all'idea di aver perso per sempre i miei preziosissimi dati: "vuoi vedere che questo cosino uscito dalla scatola ha qualcosa a che vedere col disco non funzionante?". Probabilmente muovendo la macchina, si era staccato dal disco uno dei ponticelli che permettono di configurarlo come master/slave. Una volta risistemato il ponticello al suo posto la macchina ha ricominciato a funzionare a dovere.

Di aneddoti simili a questo, se non più gravi, se ne potrebbero raccontare a decine. A prescindere da quale fosse stato il motivo che ha causato la perdita di dati, in tutti casi si può trovare un comune denominatore, ovvero la sottovalutazione del rischio.

E' bene ricordare che gli hard disk non sono eterni: il tempo medio stimato fra due problemi hardware (MTBF, mean time between failures) è nell'ordine delle 200.000-500.000 ore per i modelli migliori. In realtà questo valore è tale solamente sulla carta e sarebbe una mancanza di buon senso aspettarsi che il nostro nuovo hard disk possa durare per i prossimi 57 anni.

I valori indicati non possono certamente essere applicati pari pari su un singolo esemplare.

L'MTBF è basato infatti su stime ottenute stressando pochi esemplari in condizioni di laboratorio. Esso perciò non deve essere considerato un valore assolto, bensì solamente un indice di affidabilità utile al più per paragonare modelli diversi.

Il MTBF inoltre non tiene conto di molti fattori importanti che possono causare la perdita di dati, quali problemi del software, interventi errati dell'utente, blackout, urti, fulmini, incendi, furti, pirati informatici, raggi cosmici, ecc. Di solito si consiglia di sostituire l'hard disk circa ogni due anni.

Fatelo bene, fatelo spesso !

Una delle cause principali che spinge a non fare il backup o a non farlo sufficientemente spesso è sicuramente associata alla necessità di intervento da parte dell'utente. Meno l'utente deve preoccuparsi di fare il backup, meno c'è il rischio che egli si dimentichi o non abbia voglia di farlo.

Per questo deve essere valutata all'inizio con molta cura la scelta della strumentazione necessaria (hardware, software, policy e procedure). Bisogna riuscire infatti a fare in modo che il backup sia una operazione il più automatica possibile e che vi sia il minor spazio per eventuali errori. L'ottimo sarebbe che il backup fosse schedulato in modo automatico e che non fosse richiesto l'intervento umano, ad esempio per cambiare nastro a metà backup. Devono essere inoltre attentamente valutate le conseguenze di un eventuale recupero del backup sul normale funzionamento della macchina (richiesta di tempo uomo, fermo macchina, ...).

Scegliere il supporto

Il backup è come una assicurazione: vale la pena spendere qualcosa di più e dormire sonni tranquilli. Ovviamente buttare via soldi e tempo non è mai un'ottima idea e conviene perciò calibrare investimenti e impegno richiesto in base all'importanza dei dati.

La scelta del supporto e dell'hardware per il backup è forse la più importante e difficile da prendere, in quanto condiziona pesantemente tutta la procedura di backup che si utilizzerà in seguito. Ad esempio scegliendo di fare le copie di sicurezza su nastro, il recupero di un singolo file risulterà poi una operazione lenta. Al contrario, facendo il backup su CD, l'accesso ed il recupero dei file salvati sarà immediato, a costo di una certa complicazione in fase di masterizzazione (occorre uno spazio temporaneo su disco di almeno 640Mb dove creare l'immagine ISO9660 del CD-R che si sta creando).

La scelta di memorie di massa per il backup è attualmente abbastanza vasta, anche se le alternative migliori sono ancora relativamente costose. Andando al risparmio si rischia però di scegliere soluzioni inadeguate e di pagare il vantaggio iniziale sotto forma di tempo e di costo eccessivo dei supporti: un DAT SCSI costa molto di più di un masterizzatore, ma è anche vero che se si hanno parecchi dati da salvare occorrono parecchi CD-R e parecchio tempo. Ciò inoltre comporta la ne-

cessità di un certo intervento umano, con conseguente possibilità di errori o di non eseguire il backup per pigrizia o mancanza di tempo. Avendo un supporto di capacità superiore a quella dei dati da salvare è invece possibile eseguire tutte le operazioni in modo automatico, limitandosi eventualmente ad eseguire manualmente la sola "rotazione" dei nastri.

A mio vedere il nastro è perciò ancora una delle soluzioni col migliore rapporto prezzo prestazioni.

Scegliere cosa salvare

Come regola è bene sempre tenere separate le cose importanti da quelle non necessarie. Non bisogna però dimenticare di farlo usando il buon senso. Ad esempio è vero che il sistema operativo può essere facilmente reinstallato a partire dal CD originari, ma è anche vero che se non si è avuta cura di salvare i file di configurazione precedenti, tale operazione può fare perdere parecchio tempo. In generale non è una brutta idea tenere in dischi o partizioni separate il sistema operativo e i dati degli utenti. Tale organizzazione presenta anche il vantaggio di rendere più sicure le operazioni di aggiornamento, in quanto è possibile smontare il filesystem con i dati mentre si reinstalla il sistema.

Dal punto di vista del backup, l'evitare di salvare i programmi che possono essere reinstallati può significare un notevole risparmio. Un altro consiglio importante è quello di archiviare una volta per tutte i file relativi a lavori terminati, in modo da non doverli più "backuppare" ogni volta.

Scegliendo con attenzione quali file salvare, è possibile ad esempio far stare tutti i dati su un unico CD-R, rendendo inutile l'acquisto di un dispositivo di backup più costoso.

Copiare singoli file oppure una intera partizione

Esistono due possibilità per eseguire il backup: quella di salvare l'immagine dell'intero disco (o di una partizione dello stesso) e quella di salvare i singoli file e directory. Ognuna delle due scelte presenta i propri vantaggi e svantaggi.

Nel primo caso si otterrà una immagine esatta, eseguita settore per settore, del disco. Per ripristinarla in modo corretto sarà necessario avere un disco di capacità uguale o maggiore di quello originario (in questo caso lo spazio in più verrà perso). Di norma non sarà possibile ripristinare singoli file, ma solamente tutto il disco allo stato in cui si trovava al momento del salvataggio (vedremo fra poco che tale limitazione è aggirabile usando il loop device). Nonostante le limitazioni, tale approccio presenta notevoli vantaggi in termini di velocità. Nel caso del salvataggio di singoli file o directory si è invece indipendenti dalla struttura del disco di partenza e si possono in caso di bisogno recuperare anche singoli file o directory, anche in directory diverse da quelle originarie. Generalmente il salvataggio avviene appoggiandosi ad appositi programmi di archivio, come tar, di cui abbiamo già parlato la volta scorsa, e cpio.

Policy di backup: salvataggi totali o incrementali

Il metodo ideale per eseguire un backup è certamente quello di fare ogni volta una copia completa di tutti i dati utilizzando un supporto diverso da quello usato in precedenza. Se la quantità di dati da salvare è consistente, non sempre questa operazione risulta agevole. In questo caso è possibile eseguire un backup "incrementale", salvando solamente le differenze rispetto al giorno precedente. Ovviamente la prima volta verrà comunque eseguito un backup totale.

Pur non esistendo delle regole valide per tutti i casi, una buona soluzione molto utilizzata, è quella di eseguire il backup completo una volta alla settimana - ad esempio di domenica, quando la macchina ha meno lavoro - e nei giorni successivi salvare solamente i file che sono stati modificati. Ovviamente bisognerà prevedere la rotazione di più nastri, per evitare di scrivere sopra ad un pezzo di backup della stessa serie. Nel caso si debbano recuperare dei dati, occorrerà inoltre ricaricare in sequenza tutte le copie una sopra l'altra nello stesso ordine in cui esse sono state salvate. Il salvataggio incrementale dei dati viene effettuato sfruttando la data di ultima modifica dei file (Linux per ogni file mantiene oltre a questa data anche quella di creazione e di ultimo accesso).

Uso dei "nastri" in Linux

Il termine "nastro" indica genericamente una famiglia di dispositivi di diverso tipo: DAT, DSS, ... che permettono la memorizzazione di dati su una bobina di nastro magnetico. Si tratta, ovviamente, di dispositivi ad accesso sequenziale, in cui per leggere l'ultimo byte è necessario scorrere tutto il supporto. La capacità varia da alcune centinaia di Mb a parecchi Gb. Di solito i modelli migliori (e più costosi) sono connessi al computer mediante una interfaccia di tipo SCSI. Esistono tuttavia periferiche economiche collegabili a controller proprietari, al bus IDE, alla porta parallela o al cavo del floppy disk (ftape).

Una volta configurato in modo opportuno il dispositivo, esso è accessibile dal sistema mediante un file speciale in /dev, gestito in modalità a carattere e dipendente dal dispositivo utilizzato (ad esempio /dev/st0 se si usa un tape SCSI).

Per uniformità si preferisce creare un link simbolico /dev/tape che punti al file speciale opportuno.

Il comando mt

Esistono due versioni dei file speciali che permettono di accedere ad un tape: quella con riavvolgimento automatico quan-

Sicurezza delle copie di backup: backup crittografati

Ha senso mettere firewall a protezione della propria rete aziendale se poi si lascia la cassetta col backup sopra il tavolo alla portata di tutti o la procedura di backup liberamente accessibile ? Raccontata così sembra una storiella, ma è dimostrato che una parte significativa delle fughe di dati avviene ad opera di dipendenti disonesti (asportando delle copie di sicurezza o creandosene a loro volta).

Per evitare problemi sarebbe bene tenere le copie al sicuro o in alternativa scrivere il backup in modo criptato. Per far ciò è in teoria possibile utilizzare qualunque programma di codifica direttamente sull'immagine del CD prima di masterizzarla. In questo caso però la lettura dei dati salvati non risulta particolarmente agevole. Meglio quindi appoggiarsi ad uno dei filesystem criptati disponibili per Linux, come CFS (Cryptographic FileSystem, disponibile su http://www.cryptography.org) oppure l'italiano TCFS (http://tcfs.dia.unisa.it/).

do il software esegue il "close" del file (es: /dev/st0) e quella senza riavvolgimento automatico (dev/nst0). Ciò rende possibile la scrittura di più archivi di seguito sullo stesso nastro, separati fra loro da opportuni marcatori di BOF (Begin Of File) e EOF (End Of File). Il nastro viene terminato "logicamente" da un EOT (End Of Tape). Ovviamente usando il dispositivo senza riavvolgimento automatico, sarà cura dell'utente riportare all'inizio il nastro dopo l'uso. Ciò è possibile mediante il comando mt:

```
# mt rewind
```

Esso offre altre funzioni, che permettono ad esempio di spostarsi fra i vari marcatori, di tendere il nastro oppure di ottenere informazioni sullo stesso (densità di registrazione, lunghezza, ...).

Il backup in pratica

Dopo aver spiegato i concetti base, vediamo come implementare in pratica le nostre procedure di backup su Linux.

Dump di una partizione mediante dd

Il dump settore per settore di una partizione o di un intero disco può essere ottenuto mediante il comando dd, usando come input file speciale in /dev che rappresenta la periferica desiderata. Per ottenere una immagine della prima partizione del secondo hard disk IDE si utilizza:

```
# dd if=/dev/hdb1 of=/tmp/backup.img
```

Le stringhe che seguono if= e of= indicano rispettivamente il nome dei file da cui leggere e scrivere i dati.

Ovviamente l'immagine ottenuta avrà le stesse dimensioni della partizione salvata. Essa può essere trattata come un comune file, ovvero copiata su un altro disco, compressa, ... Se di dimensioni appropriate, essa può essere "bruciata" su un CDROM. In questo caso, invece di un CD in formato ISO9660, si otterrà un CD con sopra un filesystem dello stesso tipo di quello presente in /dev/hdb1, ad esempio ext2. Esso può essere montato in sola lettura (-oro) col comando:

```
# mount -t ext2 -oro /dev/cdrom /mnt/cdrom
```

L'accesso ad un CD contenente un filesystem di tipo diverso da ISO9660 è tuttavia più lento e causa un certo movimento della testine del lettore, in quanto i dati sono organizzati in modo più "sparso" rispetto al formato proprio dei CD, il quale è ottimizzato per evitare tali problemi.

Se si è abilitato il supporto per il "loop device" (il modulo da caricare al boot è /lib/modules/2.2.12-20/block/loop.o), si potrà montare direttamente il file contenente l'immagine come se si trattasse di una partizione. In questo caso è anche possibile montarla in lettura/scrittura, cosa che tuttavia è sconsigliabile nel caso di un backup:

```
# mount -t ext2 -oloop,ro /tmp/ /mnt/backup
```

Dump su nastro

Specificando anche dopo of= il nome di un file speciale che rappresenta una periferica, è possibile scrivere l'immagine direttamente sulla periferica stessa, ad esempio sul nastro:

```
# dd if=/dev/hdb1 of=/dev/tape bs=1024
```

In questo caso l'archivio non viene scritto appoggiandosi ad un filesystem, bensí usando il dispositivo in modo "raw", ovvero settore dopo settore.

Nell'esempio si è specificata la dimensione dei blocchi di dati da copiare in un singolo ciclo di lettura/scrittura, allo scopo di ottimizzare le prestazioni in funzione delle caratteristiche del nastro. Ovviamente, essendo il nastro una periferica gestita in modalità "a carattere", non è possibile montare l'immagine del filesystem per leggere singoli file, ma solamente recuperare il disco originale usando "al contrario" il comando dd:

```
# dd if=/dev/tape of=/dev/hdb1 (ATTENZIONE: sovrascrive la partizione)
```

Ovviamente il recupero di dati su una partizione attualmente montata è una operazione altamente sconsigliabile. Nel caso si dovesse recuperare la partizione di root di su cui risiede il sistema operativo, sarà perciò bene lavorare facendo il boot della macchina da un dischetto di ripristino.

Il comando dd ovviamente non deve essere necessariamente utilizzato su una partizione. Si può ad esempio fare il backup dell'intero disco, comprese le partizioni, la tabella delle partizioni e il master boot record. Ovviamente in questo caso si dovrà recuperare il dump su un disco identico a quello originario, in quanto il BIOS indirizza i blocchi del disco basandosi sulla sua geometria.

Backup e copia settoriale di dischetti e CD

Un utilizzo molto utile di dd è quello di usarlo come copiatore settoriale oppure per fare la copia di backup di un dischetto:

```
# dd if=/dev/fd0 of=/tmp/miofloppy.img
```

Una volta ottenuta la copia del dischetto, si può montarlo "come se fosse" un floppy reale usando il loop device:

```
# mount -t msdos -oro,loop /tmp/miofloppy.img /mnt/floppy
```

Per riscrivere l'immagine su un nuovo dischetto si utilizza il comando "inverso" al precedente:

```
# dd if=/tmp/miofloppy.img of=/dev/fd0
```

Questo metodo è utilizzato dalle distribuzioni di Linux per distribuire all'interno di un CD copia dei dischetti di boot.

Ovviamente il metodo precedente funziona anche con altri dispositivi. Ad esempio è possibile farsi una copia di un CDROM e poi montarla in modo che Linux lo veda come se fosse un CD reale, oppure usare l'immagine ottenuta per masterizzare un CD-R uguale all'originale.

Backup usando Backup/restore

Nonostante dd sia molto semplice da usare, esso è un me-

todo piuttosto spartano per eseguire il backup di dati importanti: non offre ad esempio alcun controllo di integrità dei dati.

Per sopperire a questo problema sono disponibili le utility backup/restore, derivate dallo UNIX BSD. Esse sono pensate in particolare per eseguire e successivamente recuperare il backup di filesystem ext2. E' possibile scegliere se eseguire un dump totale oppure un backup incrementale dei soli file modificati dall'ultimo salvataggio. Fra le particolarità di dump/restore vi è quella di gestire in modo automatico la rotazione dei nastri (usando un algoritmo basato sulla Torre di Hanoi) e di poter funzionare via rete, mediante un opportuno servizio rmt.

Backup usando cpio

Un altro programma di backup molto utilizzato, in quanto disponibile di serie su tutte le versioni di UNIX, è cpio. Esso legge la lista dei file da salvare dallo standard input. Tale caratteristica è particolarmente utile se usata in abbinamento con find, poichè permette di eseguire backup dipendenti da condizioni anche complesse, ad esempio dei soli file più recenti di due ore:

```
# find / -mmin -120 | cpio -o >/dev/tape
```

Backup con tar

Tar è forse il programma di backup più utilizzato in ambiente UNIX. Ne abbiamo già usato un sottoinsieme di funzioni la volta scorsa per creare semplici archivi di file.

Negli esempi che abbiamo visto, l'archivio veniva creato in un file con estensione .tar. In realtà le cose funzionano benissimo anche specificando come destinazione il nome di un device speciale che rappresenta un dispositivo hardware, ad esempio /dev/tape:

```
# tar cvf /dev/tape /home
```

Abbiamo già visto che per listare il contenuto dell'archivio si utilizza l'opzione -l (ovviamente dopo avere atteso il riavvolgimento automatico del nastro):

```
# tar tvf /dev/tape
```

Per recuperare l'archivio si usa invece l'opzione -x dopo essersi posizionati nella directory da cui si vuole partire a recuperare i file. In questo modo è possibile recuperare un backup in una directory diversa da quella originale senza sovrascrivere i file:

```
# mkdir /tmp/recuperobackup
# cd /tmp/recuperobackup
# tar xvf /dev/tape
```

Volendo si potranno recuperare anche singoli file o sottoalberi (ci si ricordi che tar salva i file in modo relativo, ovvero che non bisogna specificare lo slash iniziale nei nomi dei percorsi):

```
# tar xvf /dev/tape home/beppe
# tar xvf /dev/tape home/lorenz/docs/pippo.txt
```

In tar è possibile indicare esplicitamente o escludere determinati file dal salvataggio. E' possibile specificare quali file inserire in archivio semplicemente scrivendone il nome nella riga di comando, tenendo conto che i contenuti delle directory vengono salvati in modo ricorsivo:

```
# tar cvf /dev/tape /etc /usr/etc /usr/local/etc
```

Se i file da salvare sono molti, risulta più comodo inserire la lista dei file desiderati su un file ed indicare a tar di leggerlo usando l'opzione –files-from. La lista può essere eventualmente generata al volo usando ad esempio find:

```
# find / -user beppe -print >/tmp/filedibeppe.txt
# tar cvf /dev/tape –files-from=/tmp/filedibeppe.txt
```

Ovviamente il backup totale si può ottenere semplicemente indicando la directory root:

```
# tar cvf /dev/tape /
```

Tale scelta tuttavia presenta degli inconvenienti. Ad esempio se il CDROM è montato in /mnt/cdrom, ne viene fatto un inutile backup. Il comando precedente inoltre finisce in un ciclo senza fine quando tenta di leggere alcuni file speciali in /proc che generano in continuazione dati. Per ovviare a questo inconveniente conviene escludere tali directory dalla copia. Per far ciò si utilizza l'opzione –exclude=pattern

```
# tar cvf /dev/tape / –exclude=/proc
```

oppure –exclude-from= qualora si desiderasse ottenere la lista dei file da escludere da un file:

```
# tar cvf /dev/tape / –exclude-from= /tmp/danoncopiare.txt
```

Backup su floppy e multivolume

Se i dati da salvare sono pochi, si può pensare di fare il backup su dischetto:

```
# tar cvf /dev/fd0 /etc
```

Se un solo floppy non è sufficiente, si può usare l'opzione -M per abilitare il modo "multivolume":

```
# tar cvMf /dev/fd0 /home
```

Così facendo, il programma tar si accorgerà quando il dischetto è pieno e chiederà di sostituirlo col "volume" successivo. Ovviamente bisogna avere cura di numerare in ordine crescente i dischetti, per evitare di invertirli al momento del ripristino. Il meccanismo funziona anche con altri dispositivi: ad esempio è possibile fare il backup multivolume su più nastri (alle volte però è necessario indicare esplicitamente a tar la dimensione del nastro, poichè alcuni modelli hanno problemi a segnalare l'End Of Tape).

Backup su ZIP/Jaz

Tali dispositivi sono visti da Linux come dei dischi SCSI (anche la versione parallela), generalmente come /dev/sda4. Si potrebbe pensare che il metodo di backup più semplice sia quello di crearci un filesystem, montarlo e copiarci i file

necessari. Tuttavia, data la loro limitata capacità, risulta più semplice usarli col multivolume come se fossero delle periferiche "raw" invece di doversi preoccupare di suddividere i file da copiare su più dischetti:

```
# tar cvMf /dev/sda4 /usr /var /etc
```

Backup incrementali

Usando tar è possibile eseguire backup incrementali, ovvero salvare tutti i file più recenti della data dell'ultimo backup:

```
# tar cvf /dev/tape –newer="Oct 24" /
```

Tale funzione risulta particolarmente utile per automatizzare gli script di backup incrementale.

I backup incrementali possono essere salvati anche di seguito su un unico archivio tar usando l'opzioni –listed-incremental.

Backup compressi

Abbiamo visto la volta scorsa che il comando tar supporta la compressione in modo trasparente. E' possibile applicare tale funzione ad un backup aggiungendo l'opzione "z":

```
# tar cvzf /dev/tape /dev /etc /home /usr /var
```

Tuttavia non sempre tale approccio è consigliabile. Infatti, comprimendo tutto il "flusso" di dati con gzip può accadere che un singolo bit rovinato nel backup comporti l'impossibilità di leggere tutti i file seguenti nell'archivio.

Per eseguire backup compressi conviene invece utilizzare il programma afio, il quale esegue la compressione su ogni singolo file invece che sull'intero archivio e gestisce in modo migliore eventuali file corrotti. A causa delle molteplici opzioni che esso supporta, di solito afio non viene utilizzato come un programma a sè stante, bensí come un "archive engine" per altri script o software di backup. La distribuzione di afio contiene parecchi esempi di script già pronti, mentre programmi più complessi che supportano afio come motore di backup si possono reperire su ftp://sunsite.unc.edu/pub/linux/system/backup oppure eseguendo una ricerca su http://www.freshmeat.net/.

Backup su hard disk e altre periferiche removibili

Spesso comperare un altro hard disk è assai più economico che acquistare l'unità di backup ed è per questo motivo che spesso si sceglie di eseguire le copie di sicurezza proprio su questo supporto. In questo caso sarebbe bene acquistare un apposito "frame" removibile, in modo da poter togliere il disco e metterlo al sicuro (non ha senso fare il backup se poi lo si lascia nelle stesse condizioni di insicurezza del resto del sistema). Questa operazione tuttavia causa un certo tempo di fermo macchina e non in tutti i casi ciò è accettabile. Inoltre diventa costoso tenere più copie di backup.

Per risolvere tale problema è possibile utilizzare invece di un hard disk un'altra periferica a blocchi removibile (ZIP, JAZ, LS-120, ...). Ai fini pratici le operazioni da compiere per eseguire il backup sono le medesime e consistono nel mounting del filesystem in una directory e nella creazione al suo interno di un file tar.

Una alternativa molto valida se si hanno parecchie macchine è quella di eseguire il backup via rete su un altro server Linux/UNIX o NT. In questo modo è possibile ripartire i costi fra più macchine. Per fare un backup via rete esistono parecchie soluzioni, la più semplice delle quali è quella di montare un disco del server mediante NFS o Samba e salvarci sopra un archivio tar (la semplice copia di file perde le informazioni di ownership e i permessi dei file).

Un semplice script per eseguire il backup su disco può essere simile al seguente.

```
#!/bin/sh

cd /

/bin/mount -t ext2 /dev/hdc1 /backup/

rm -f /backup/backup.*.tar.gz

/bin/tar cvzf /backup/backup.`date +%Y%m%d`.tar.gz bin
boot dev
  etc home lib root sbin usr var >/tmp/backup.`date
+%Y%m%d`.log

/bin/umount /backup
```

Si noti che, per ragioni di sicurezza, è preferibile tenere montato il disco o la partizione su cui si esegue il backup solamente durante le operazioni relative. A voler fare le cose per bene bisognerebbe prima di creare un nuovo archivio cancellare i precedenti, magari tenendo i più recenti se lo spazio lo consente. Una buona idea sarebbe anche quella di aggiungere allo script un controllo di corrispondenza fra il backup appena effettuato ed il contenuto del disco (ad esempio utilizzando l'opzione -W di tar). A questo proposito è bene notare che è preferibile eseguire i backup mentre non si eseguono altri programmi che possano andare a modificare il filesystem, onde evitare disallineamenti fra quanto salvato ed il contenuto del disco. Ciò si può ottenere eseguendo il backup in modalità "single user" oppure, se questo non fosse possibile, di notte quando l'attività del sistema è più limitata.

Backup via rete

Con poche modifiche alla riga che monta il filesystem su cui eseguire il backup, lo script precedente può essere adattato per utilizzare un server di rete, montando un volume condiviso mediante NFS (ovviamente è necessario avere gli opportuni permessi):

```
/bin/mount -t nfs lxserver.profuso.com:/discone /backup/
```

Usando un server Windows invece si può montare il disco nel seguente modo:

```
/bin/mount -o username=pippo,password=pippo,netbio-
sname=freddy
  -t smbfs //ntserver/JAZZ /backup/
```

Per eseguire il backup su una macchina Windows, può essere anche utilizzato il programma smbtar, presente nella distribuzione di Samba.

Il backup di rete può essere effettuato anche senza passare attraverso un filesystem montato se nel server di backup è presente il server rsh (Remote Shell), usando una sintassi del tipo:

```
# tar cvf lxserver.profuso.com:/dev/tape /home
```

In questo modo tar salverà l'archivio tar direttamente sul tape collegato alla macchina lxserver.profuso.com.

Archiviare su CDROM: cdrecord

Il metodo generalmente utilizzato per eseguire un backup su CD-R o CD-RW è quello di creare una immagine su disco del filesystem ISO9660 e poi di andarla a "bruciare" sul supporto ottico.

Queste operazioni possono essere eseguite usando gli strumenti a linea di comando (mkiso9660 e cdrecord) oppure in modo più semplice utilizzando un frontend grafico agli stessi quale potrebbe essere xcdroast. E' bene tenere a mente che il filesystem ISO9660 non supporta tutte le caratteristiche di un vero filesystem UNIX (permessi, lunghezza dei file e dei percorsi, ...). Per questo è bene abilitare il supporto per le cosidette Rock Ridge Extensions.

Per eseguire il backup in questo modo è necessario disporre di uno spazio temporaneo su disco pari alla dimensione dell'immagine che si intende creare. Una alternativa possibile che non fa uso dello spazio temporaneo è quella di scrivere direttamente l'archivio tar settore per settore sul CD, usando un pipe (i simboli - indicano rispettivamente di mandare l'output sullo standard output e di leggere dallo standard input):

```
# tar cf - | cdrecord -v speed=4 dev=0,0,0 -
```

Tale espediente risulta però abbastanza critico, in quanto per la corretta scrittura di un CD è necessario che i dati vengano forniti al masterizzatore in modo continuo. Nel caso il processo tar che crea l'archivio trovasse degli errori leggendo dei blocchi di un file, è possibile che il tempo necessario per la rilettura possa essere eccessivo. I moderni masterizzatori contengono comunque al proprio interno dei buffer di dimensione sufficiente a consentire pause dell'ordine di parecchi secondi. Per minimizzare i problemi si consiglia comunque di utilizzare le velocità di scrittura più basse (in modo che il buffer contenga più "secondi").

Per leggere l'archivio salvato non serve montare il CD, ma è sufficiente accederci con tar in modo "raw":

```
# tar tvf /dev/cdrom
```

Un programma alternativo all'uso di tar+cdrecord è cdbackup, prelevabile da http://www.cableone.net/ccondit/cdbackup/. Esso permette di eseguire anche più backup sullo stesso CD, usando tracce differenti:

```
# tar cf - /home | cdbackup -dev /dev/cdrom -rdev 2,0 -label "/home backup"
# tar cf - /etc | cdbackup -dev /dev/cdrom -rdev 2,0 -label "/etc backup"
# cdrestore -dev /dev/cdrom -track 2 | tar xvf -
```

L'alternativa al backup: RAID

RAID, acromino di "Redundant Arrays of Inexpensive Disks" è una tecnologia che permette di utilizzare assieme più dischi allo scopo di migliorare le prestazioni e/o garantire una maggiore affidabilità.

I miglioramenti di prestazioni vengono ottenuti dividendo i dati da scrivere su più dischi, in modo da sfruttare i parallelismi interni dell'hardware per ottenere una maggiore velocità (se si utilizzano n dischi, su ognuno viene salvata solo una quantità pari ad 1/n delle informazioni).

L'affidabilità invece viene ottenuta creando una ridondanza nei dati. Tale ridondanza può essere ottenuta in modo semplice scrivendo contemporaneamente gli stessi dati su più dischi oppure ricorrendo ad algoritmi più complessi di suddivisione delle informazioni. In ogni caso il risultato che si vuole ottenere è quello di fare in modo che non vi sia perdita di dati anche qualora uno o più dischi si dovessero guastare. In questo senso il RAID è una valida alternativa al backup, anche se non mette al sicuro in caso di guai particolarmente seri (come il furto dell'intera macchina).

Non tutti i "livelli" previsti da RAID sono in grado di garantire contemporaneamente massima velocità e massima sicurezza (ecco spiegato quell' "e/o"). La scelta di utilizzare un livello RAID piuttosto che un altro è dettata da diversi fattori, che possono privilegiare di volta in volta uno dei due aspetti. Si noti che i livelli RAID non sono ordinati in modo gerarchico. Non necessariamente RAID 3 è migliore di RAID 2: sono semplicemente diversi.

Alcuni livelli RAID sono implementati direttamente via software dal kernel di Linux, ad esempio lo "striping senza parità" (livello RAID 0), che permette di vedere più dischi come un unico grande volume (senza offrire ridondanza ma migliorando notevolmente le prestazioni) oppure il "mirroring" (livello RAID 1), che permette di scrivere contemporaneamente gli stessi dati su più dischi, creando una sorta di backup automatico in cui si possono guastare n-1 dischi. Tale funzione si paga in termini di costo e di velocità di scrittura (le prestazioni in lettura invece aumentano, poichè è possibile sfruttare il parallelismo dei dischi).

Linux supporta anche i livelli 4 e 5 ("striping con parità"), che offrono una capacità di memorizzazione pari a n-1 volte la dimensione del singolo disco, gestendo il possibile crash di uno qualunque degli n dischi mediante meccanismi di parità. Il livello 5 mette assieme ad un costo contenuto le migliori caratteristiche dei livelli 0 e 1, garantendo ottime prestazioni ed un buon grado di affidabilità (anche se minore del livello 1).

Il supporto del RAID via software tuttavia rappresenta un carico piuttosto pesante per la macchina, motivo percui spesso si preferisce utilizzare un apposito controller SCSI che lo gestisca in modo trasparente. Di solito tali periferiche offrono anche funzionalità di "hot swap", ovvero permettono la sostituzione "a caldo" del disco difettoso senza necessità di fermare il sistema.

Come si intuisce, il costo da pagare per adottare soluzioni di questo tipo è più alto rispetto che utilizzando un singolo disco associato ad un sistema di backup. Tuttavia in molti casi le funzioni offerte e la possibilità di non fermare il lavoro possono essere dei buoni motivi per scegliere tali soluzioni.

Per maggiori informazioni si consulti l'HOWTO Software-RAID-HOWTO, disponibile su http://www.linuxdoc.org/.

Se le proprie partizioni sono di dimensione inferiore ai 640Mb, è possibile eseguire una copia settore per settore della partizione su CD-R senza passare per l'immagine, usando il comando:

```
# cdrecord -v speed=4 dev=0,0,0 </dev/hda1
```

Il CD ottenuto è una copia identica della partizione e deve essere montato nel filesystem specificando il tipo corretto di filesystem. Se la partizione era di tipo vfat, si userà il seguente comando:

```
# mount -t vfat -oro /dev/cdrom /mnt/backup
```

Scrivere CD "al volo"

E' in fase di realizzazione il supporto per Linux della tecnologia UDF, usata in WIndows da programmi tipo Direct CD, la quale permette di montare un CD-RW come se fosse un dispositivo scrivibile, in modo simile ad un hard disk. Purtroppo tale tecnologia è utilizzabile solamente con le ultimissime versioni del kernel (2.3.x e 2.4.x).

Ulteriori informazioni sulla scrittura di CD in Linux può essere reperita su http://www.linuxdoc.org/HOWTO/CD-Writing-HOWTO.html

Schedulazione automatica del backup

Una volta scritta la nostra procedura di backup, possiamo fare in modo che essa venga eseguita in modo automatico. Per far ciò possiamo utilizzare lo schedulatore di Linux "crontab". Per aggiungere un programma alla propria crontab (ne esiste una per ogni utente del sistema) è sufficiente eseguire il comando

```
# crontab -e
```

Si aprirà in questo modo un editor (vi) che permetterà di modificare la tabella relativa. Ogni linea ha il seguente formato:

```
mm hh dd mo wd      linea di comando
```

Dove mm e hh rappresentano rispettivamente minuto e ora in cui effettuare l'operazione indicata. Le etichette dd, mo e wd indicano invece il giorno del mese, il mese e il giorno della settimana (0 o 7 indicano la domenica). Al posto di qualunque di questi valori è possibile indicare degli intervalli (es: 8-10), delle liste (es: 5,10,15,20), degli intervalli (/2 indica "ogni 2") oppure una combinazione di questi valori. Il simbolo *, come nel caso delle wildcard, equivale a "qualunque" valore.

La seguente riga esegue lo script di backup ogni domenica e mercoledì alle 04:30 di mattina:

```
30 4 * * 0,3 /usr/local/bin/backup.sh
```

La riga seguente invece esegue un programma ogni giorno ad intervalli di 2 ore:

```
0 */2 * * * /usr/local/bin/mioprogramma
```

Si noti che l'output e gli errori generati vengono spediti per e-mail all'utente proprietario della crontab. Per evitare ciò si possono ridirezionare standard error e standard output in /dev/null:

```
30 4 * * 0 /usr/local/bin/backup.sh 2>/dev/null >/dev/null
```

Frontend per i programmi di backup e software commerciali

Personalmente preferisco eseguire il backup affidandomi ai programmi standard presenti di serie su tutte le installazioni di Linux, principalmente in quanto, nel malaugurato caso dovessi eseguire il restore di una macchina da zero, non dovrei prima installare software aggiuntivo.

Nel caso fare uno script di backup risulti troppo complesso, una buona soluzione può essere quella di utilizzare uno dei moltissimi programmi che fungono da "frontend" nei confronti dei programmi standard di backup di Linux. Cercando su un motore di ricerca come http://www.freshmeat.net/ se ne dovrebbero trovare a decine.

Per chi si sentisse più a proprio agio usando un software commerciale, esistono diversi prodotti adatti a tutte le esigenze. Essi di solito semplificano di molto ciò che si potrebbe ottenere anche usando le funzioni standard. Questo non è certamente un male, in quanto la possibilità di fare il backup facilmente invoglia l'utente a farlo. Generalmente tali software si basano su una interfaccia che permette di scegliere in modo intuitivo cosa salvare o recuperare. Alcuni di essi permettono il backup centralizzato di più macchine Linux o Windows su uno stesso server di rete.

Alcuni esempi di software commerciali famosi per Linux sono: Arkeia, BRU, Perfect Backup.

Conclusioni: attenzione alle false sicurezze !

Non ha senso fare il backup se poi non si è in grado di recuperarlo. Pare una cosa ovvia ma, credetemi, ho visto anni di lavoro buttati al vento perchè nessuno si era mai preso la briga di verificare che uno script di backup funzionasse correttamente.

Ricordatevi sempre di verificare il backup, ad esempio recuperandolo in un disco o una directory vuota. Alcuni software di backup permettono di verificare la corrispondenza di quanto archiviato con i file originali. Sarebbe bene compiere tale operazione ogni volta che si esegue il backup, in quanto i fattori che possono portare ad un backup corrotto sono molteplici.

Verificate anche che col passare del tempo i dati non aumentino in modo tale da riempire il nastro o il CD di backup, altrimenti correrete il rischio di ritrovarvi con backup incompleti.

Per la massima sicurezza sarebbe bene tenere più copie del backup, possibilmente in luoghi diversi, in modo da cautelarsi da eventuali eventi disastrosi. MC

Se credi che la leucemia sia un male inguaribile devi farci un favore.

Piantarla.

Dall' 8 al 10 dicembre
nella tua città trovi
le Stelle di Natale per sostenere
la ricerca e la cura
delle leucemie e dei linfomi.

ASSOCIAZIONE ITALIANA
CONTRO LE LEUCEMIE-LINFOMI
ONLUS

Sede Nazionale Via Ravenna, 34 - 00161 Roma

c/c Postale n. 46716007

www.ail.it

Se vuoi sapere quali sono le piazze con le Stelle dell'Ail chiama il numero 06/4402696

TRENT'ANNI DI ATTIVITÀ