M2 net - http://www.m2net.it.eu.org

# Corso di Linux – Parte II

Alessio Pennasilico <mayhem@spippolatori.org>

Sabato 29 Novembre 2002

# Introduzione

- Chi è Alessio Pennasilico
- Cosa è M2 net

- Finalità del Corso

- Tempi e modi del corso
- Presentazione dei partecipanti

# Alessio Pennasilico

- Svolge attività di consulenza presso diverse aziende, principalmente in merito alle tecnologie legate ad Internet.
- Sicurezza e Cisco sono le cose a cui si interessa anche durante il tempo libero.
- E’ membro di M2 net.

# M2 net

- E’ un’associazione culturale non a scopo di lucro.
- Per frequentare i corsi non è necessario versare alcuna somma, né sottoscrivere tessere.
- Tutte le attività si fondano sulla reciproca collaborazione, sul personale contributo attraverso la propria esperienza e le proprie conoscenze.

# Finalità del corso

- Capire i meccanismi di fault-tolerance delle unita' di memoria di massa.
- Imparare a conoscere i meccanismi che permettono di accedere ai dischi.

# Tempi e modi del Corso

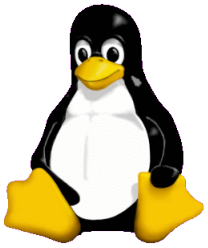
- Il corso inizia alle 9:30 e termina indicativamente alle 12:30
- Ci saranno 2 pause di 10 minuti
- Il corso si basa sulle domande dei presenti e non è richiesto alcun prerequisito per partecipare

# Presentazione dei partecipanti

- Nome
- Nozioni preesistenti in merito agli argomenti della lezione

# Legenda

- Parleremo non solo di Linux, ma anche di OpenBSD: per questa ragione useremo dei simboli quando la sintassi del comando descritto si riferira' ad uno soltanto di questi due sistemi.



Linux

OpenBSD

# Sommario

- Il raid
  - Hardware/software
  - Striping, Mirroring
  - Striping with parity
  - Mirrored striping with parity
  - Gestire il RAID con Linux
  - Trouble shooting e chroot
  - LILO

# Come proteggere i dati

- I dati memorizzati su disco potrebbero essere persi a causa di una rottura dello stesso; la rottura di un disco potrebbe inibire il funzionamento di una macchina che offre importanti servizi (es. Firewall).
- Per questa ragione si cerca di proteggersi da questa eventualita'.

# RAID

- Esistono diverse tecniche, chiamate RAID, Redundant Array of Indipendent (secondo alcuni Inexpensives) Disks, che caratterizzano il tipo gestione del disco che stiamo utilizzando.
- Quello che andremo ad illustrare vale per moltissimi sistemi operativi.

# Hardware o Software

- Tutti i meccansmi di RAID possono essere gestiti direttamente dal controller dei dischi (hardware) o dal sistema operativo (software). Nel primo caso il funzionamento del RAID e' completamente trasparente rispetto al nostro OS.
- Quello HW si trova principalmente sui controller SCSI, tuttavia nulla vieta, in entrambi i casi, di utilizzare dischi IDE.
- Noi ci occuperemo, nella pratica, solo di RAID SW.

# Striping

- Striping, o RAID 0, consiste nel concatenare tra loro piu' dischi, al fine di ottenere un disco piu' grande.
- I dischi vengono riempiti alternativamente, garantendo una maggiore velocita' in lettura scrittura.
- Non prevede fault-tolerance, la rottura di un solo disco causa la perdita del contenuto dell'intero RAID 0.
- Per implementarlo servono almeno due dischi.
- E' il RAID che a parita' di dischi offre la maggior quantita' di spazio utilizzabile.

# Mirroring

- Mirroring, o RAID 1, consiste nell'avere due dischi, di cui il secondo e' una copia fedele e costantemente aggiornata del primo.
- La velocita' in scrittura e' la stessa di un normale disco, la velocita' in lettura e' doppia (dipende).
- E' fault-tolerance (puo' dare problemi di boot).
- Per implementarlo servono due dischi.
- E' il tipo di raid che a parita' di dischi ci offre la minor quantita' di spazio utilizzabile

# Striping with parity

- Striping with parity, o RAID 5, consiste nel creare uno stripe simile al RAID 0, ma che mantiene un'area dati chiamata "parity" che permette, nel caso di perdita di uno dei dischi dell'array, di continuare a lavorare (degraded mode) e di ripristinare la situazione originale inserendo un nuovo disco.
- RAID 5 e' quindi fault-tolerance.
- La velocita' in lettura e' decisamente migliore, quella in scrittura comunque superiore a quella di un normale disco.
- Per essere implementato necessita di almeno tre dischi.
- La perdita di spazio utilizzabile e' comunque accettabile (dal 33% al 20%).

# Mirrored striping with parity

- Mirrored striping with parity, o RAID 10 consiste in un mirror (RAID1) di due array RAID5.

- Le combinazioni che si possono creare sono molte, e vi invito a consultare I siti indicati per ottenere maggiori informazioni.

# Spare Disk

- Lo spare disk e’ un disco che non viene normalmente utilizzato, ma resta a disposizione del sistema, al fine di riportare immediatamente in modalita’ normale un array di cui un disco si sia danneggiato.

# RAID e Backup

- Il RAID e' un utile, oramai indispensabile strumento per garantirci la continuita' di servizio e per evitare perdite di dati.
- **Non e'**, non puo' e non deve essere considerato un sostituto, **un'alternativa al backup.**
- Il contenuto dell'array va comunque regolarmente salvato su nastro magnetico, o su altri supporti "sicuri" e che tengano uno storico.

# Kernel e supporto delle funzionalita’

- Ogni sistema operativo ha una interfaccia tra il software che noi utilizziamo e l’hardware della macchina: il Kernel.
- Se il nostro sistema e’ in grado di utilizzare dischi IDE piuttosto che SCSI, se supporta il RAID o meno, dipende dalla configurazione del kernel.
- Diamo per il momento scontato di lavorare su macchine dove il kernel e’ gia’ impostato nel modo in cui a noi serve.

# Linux e la gestione del RAID

- I file che stanno nella /dev corrispondono ai device hardware installati sul nostro computer: esistono quindi anche i file per i dischi:
- /dev/hd* - dischi IDE
- /dev/sd* - dischi SCSI
- /dev/md* - dischi in RAID software

# Individuare il device IDE corretto

- I dischi IDE sono cosi' numerati:
  - /dev/hda – primary master
  - /dev/hdb – primary slave
  - /dev/hdc – secondary master
  - /dev/hdd – secondary slave

- NB: un cd-rom ide sara' gestito come se fosse un hard disk in sola lettura.

# Individuare il device SCSI corretto

- I device SCSI, /dev/sd*, sono numerati in sequenza, indipendentemente dall'ID.
- Es. Due hard disk, uno con ID 0 ed uno con ID 6 verranno riconosciuti come /dev/sda e /dev/sdb.
- I cd-rom SCSI verranno rilevati sempre come /dev/sd*, mentre le unita' a nastro SCSI verranno gestite come /dev/st* (tipicamente /dev/st0).

# I RAID device

- I dischi in RAID software verranno gestiti come /dev/md*, ed un numero crescente, assegnato da noi in fase di configurazione.
- Poiche' il RAID hardware e' trasparente al sistema operativo in uso, un RAID 5 hardware, con dischi SCSI, verra' utilizzato accedendo a /dev/sd*.

# Le partizioni

- Le partizioni possono essere primarie o estese.
- Le partizioni su un disco possono essere al massimo 4 primarie o 3 primarie ed 1 estesa.
- Le partizioni estese ospitano a loro volta dischi logici (logical volume).
- Ogni partizione puo' essere di diverso tipo (83 Linux, 82 Linux Swap, 7 NTFS)

# Numerazione delle partizioni

- Le partizioni primarie sono numerate da 1 a 4 (hda1/hda4, sda1/sda4).
- La partizione estesa ha sempre il numero 5 (hda5, sda5).
- La quantita' di logical volume e' a nostra discrezione e viene numerata da 6 in poi (hda6, sda6).

# Dischi ed OpenBSD

- Tutto quello che abbiamo detto vale anche per OpenBSD, presi i dovuti accorgimenti.

- Il cd-rom si chiama /dev/cd0a.
- I dischi IDE /dev/wd0.
- I dischi SCSI /dev/sd0.

# OpenBSD e partizioni

- OBSD usa sempre e comunque una sola partizione per disco (A6 OpenBSD).
- All’interno della partizione OBSD vengono create le diverse label, che possono poi essere ridimensionate e modificate dinamicamente.

# Configurare il RAID

- La configurazione del RAID si effettua attraverso il file /etc/raidtab.
- Tale file contiene il tipo di RAID che vogliamo stabilire, con quali partizioni ed in quale ordine.
- NB: le partizioni devono esistere ed essere geometricamente uguali .

# /etc/raidtab – RAID 1

```
mayhem@coniglio:~$ cat /etc/raidtab
raiddev                 /dev/md0
raid-level              1
nr-raid-disks           2
nr-spare-disks          0

device                  /dev/hda4
raid-disk               0

device                  /dev/hdc4
raid-disk               1
```

# /etc/raidtab – RAID 5

```
mayhem@coniglio:~$ cat /etc/fstab
raiddev                 /dev/md0
raid-level              5
nr-raid-disks           3
chunk-size              32
parity-algorithm        left-symmetric
nr-spare-disks          0
device                  /dev/sda3
raid-disk               0
device                  /dev/sdb3
raid-disk               1
device                  /dev/sdc3
raid-disk               2
```

# /etc/raid.conf - RAID 1

```
START array
# numRow numCol numSpare
1 2 0

START disks
/dev/sd20e
/dev/sd21e

START layout
# sectPerSU SUsPerParityUnit SUsPerReconUnit RAID_level_1
128 1 1 1

START queue
fifo 100
```

# /etc/raid.conf - RAID 5

```
START array
# numRow numCol numSpare
1 3 0

START disks
 /dev/sd1e
 /dev/sd2e
 /dev/sd3e

START layout
# sectPerSU SUsPerParityUnit SUsPerReconUnit RAID_level_5
32 1 1 5

START queue
fifo 100
```

# Attivare il RAID con Linux

- Prima inizializziamo il device:

```
root@coniglio:~# mkraid /dev/md0
```

- Poi lo formattiamo con il file-system di Linux:

```
root@coniglio:~# mke2fs /dev/md0
```

- Verifichiamo che la costruzione stia funzionando correttamente:

```
root@coniglio:~# cat /proc/mdstat
```

# Attivare il RAID con OBSD

- Prima inizializziamo il device:

```
# raidctl -C /etc/raid.conf raid0
```

Poi lo formattiamo con il file-system di Linux:

```
# newfs raid0
```

- Verifichiamo che la costruzione stia funzionando correttamente:

```
# raidctl -s raid0
```

# Ricostruire un RAID in degraded-mode con Linux

- Verifichiamo lo stato del RAID:

```
root@coniglio:~# cat /proc/mdstat
```

- Rimuoviamo dalla configurazione il disco danneggiato:

```
root@coniglio:~# raidhotremove /dev/md0 /dev/hdc1
```

- Dopo avere sostituito il disco danneggiato con uno integro:

```
root@coniglio:~# raidhotadd /dev/md0 /dev/hdc1
```

- Verifichiamo che la ricostruzione proceda correttamente:

```
root@coniglio:~# cat /proc/mdstat
```

# Ricostruire un RAID in degraded-mode con OBSD

- Verifichiamo lo stato del RAID:

```
# raidctl -s raid0
```

- Rimuoviamo dalla configurazione il disco danneggiato:

```
# raidctl -f /dev/sda2 raid0
```

- Dopo avere sostituito il disco danneggiato con uno integro:

```
# raidctl -F /dev/sda2 raid0
```

- Verifichiamo che la ricostruzione proceda correttamente:

```
# raidctl -s raid0
```

# Accedere ad un disco

- Ai diversi dischi/cd-rom si accede attraverso l'operazione di mount.
- Mount specifica quale disco, e con quali proprieta', verra' associato ad una cartella vuota ed esistente del file system.
- Es. Accedere ad un cd-rom secondary slave:

```
mayhem@coniglio:~$ mount /dev/hdd /mnt/cdrom/
```

# /etc/fstab

- Il file /etc/fstab contiene la tabella delle definizioni delle associazioni tra le partizioni ed i loro mount-point, con le relative opzioni.

```
mayhem@coniglio:~$ cat /etc/fstab
/dev/hda2       swap            swap        defaults          0   0
/dev/hda1       /               ext3        defaults          1   1
/dev/cdrom      /mnt/cdrom      iso9660     noauto,owner,ro,users  0   0
/dev/fd0        /mnt/floppy     auto        noauto,owner      0   0
none            /dev/pts        devpts      gid=5,mode=620    0   0
/proc           /proc           proc        defaults
```

# /etc/mtab

- L’elenco delle partizioni o device effettivamente montati si trova in /etc/mtab.

```
mayhem@coniglio:~$ cat /etc/mtab
/dev/hda1 / ext3 rw 0 0
none /dev/pts devpts rw,gid=5,mode=620 0 0
/proc /proc proc rw 0 0
```

# “Montare” un disco

- Se un device ha la sua voce in /etc/fstab ci bastera’ il comando “mount /dev/hdd” o “mount /mnt/cdrom” per avere quella partizione montata con tutte le opzioni da noi specificate.

# “Smontare” un disco

- Una volta terminato di lavorare con una partizione/floppy/cd-rom eseguiremo l’operazione inversa:

```
mayhem@coniglio:~$ umount /dev/hdd
umount: /dev/hdd is not mounted (according to mtab)
```

# Operazioni di scrittura

- Il nostro sistema operativo, in quanto multiutente e multitasking, scrive quando ha il tempo di farlo.
- Ad esempio scrivere un file su un floppy e' una operazione onerosa, che viene rimandata il piu' possibile.
- Per questa ragione e' sempre indispensabile effettuare un umount prima di rimuovere un supporto magnetico.

# Protetti dagli errori

- Premendo il tasto eject di un lettore cd o di uno iomega zip, il supporto non verra' espulso, almeno non fino a quando non avremo eseguito il corretto umount.
- Rimuovendo un floppy prima di averlo smontato, rischiamo di perdere I dati che pensavamo di averci scritto sopra.
- Noteremo, non appena impartito il comando umount, una attivita' sul supporto in questione.

# fsck

- fsck e’ il programma in grado di verificare la coerenza del filesystem datogli come argomento, sia fisica (es. bad sector) che logica (es. missing index).
- Puo’ essere eseguito solo su file system non in uso (non presenti in /etc/mtab).

# fsck e boot

- Nei rari casi in cui la partizione di root/boot sia danneggiata a tal punto da non permettere l'esecuzione dei normali test di coerenza al boot sara' possibile fare boot con un media alternativo (es. Cd-rom) sistemare tutto e fare ripartire correttamente il sistema.

# fsck e root directory

- La partizione principale del nostro sistema, root (/), e' sempre in uso, quindi non potremmo mai verificare la sua coerenza.
- Il sistema provvede a verificarla da solo in fase di boot ogni n (di solito 30) reboot, o in caso di non corretto spegnimento del sistema.
- Tutto questo se il sistema funziona in modo corretto (boot, inizializzazione del sistema, etc).

# Utilizzare un live CD

- Facendo boot con un live CD abbiamo a disposizione le funzionalita' complete di una macchina Linux funzionante.
- Potremo eseguire fsck sulla partizione normalmente di boot/root.
- Una volta sistemato tutto un semplice reboot fara' ripartire il sistema.

# chroot

- Quando dopo avere ripristinato la coerenza di un filesystem di boot/root abbiamo bisogno di lavorare su di esso per sistemare altre funzionalita' che richiedono che la partizione sia montata come / (es. Ripristinare il boot da MBR) possiamo usare il comando chroot.

# Utilizzo di chroot

```
root@coniglio:/mnt/hd# chroot .
root@coniglio:/# ls
  bin/  boot/  dev/  etc/  home/
  root/  sbin/  tmp/  usr/  var/
root@coniglio:/# exit
root@coniglio:/mnt/hd#
```

# LILO

- LILO, forma abbreviata di LInux LOader, e' uno dei piu' diffusi boot manager per Linux.
- Oltre a gestire la fase di boot supporta molti parametri di configurazione del sistema che possono essere passati direttamente al kernel.

# /etc/lilo.conf

- I parametri di configurazione del LILO si trovano nel file /etc/lilo.conf.
- Tale file viene utilizzato per scrivere nell'MBR o nel Root Superblock: una modifica al file soltanto non modifica I parametri di boot.
- E' necessario eseguire "lilo" per rendere attive le modifiche apportate al file di configurazione.

# Es. /etc/lilo.conf

```
mayhem@coniglio:~$ cat /etc/lilo.conf
# Global
append="console=ttyS1,9600"
boot = /dev/hda
delay = 5
prompt
# VESA framebuffer console @ 1024x768x64k
vga = 791
# End LILO global section
# Linux bootable partition config begins
image = /boot/bzImage
  root = /dev/hda1
  label = linux
  read-only # Non-UMSDOS filesystems should be mounted read-only for
   checking
# Linux bootable partition config ends
```

# Modifica dei parametri al boot

- Se LILO e' configurato per proporre una scelta al boot (prompt) ci offre una riga di comando:

```
Boot:
```

- Possiamo specificare azione diverse da quelle di default:

```
Boot: bzImage root=/dev/hda1 ro
```

# OBSD Boot prompt

- Anche OpenBSD ci permette di decidere cose diverse dal default in fase di boot.
- Il nostro prompt sara' infatti analogo a quello di Linux:

```
boot:
```

- A questo punto noi potremmo ad esempio decidere di fare partire il sistema con vecchio kernel, digitando il suo nome:

```
boot: bsd.old
```

# Riferimenti

- http://www.acnc.com/04_01_00.html (specifiche dei diversi tipi di RAID)
- http://www.pluto.linux.it (una buona traduzione degli how-to in italiano)
- http://www.openbsd.org (tutta la documentazione su OpenBSD)
- man :)

# Disclaimer

- Queste slides sono realizzate da Alessio Pennasilico per M2 net e sono soggette alla licenza GPL, sempre nella sua corrente versione, possono pertanto essere distribuite liberamente ed altrettanto liberamente modificate, a patto che se ne citi l'autore e la provenienza.
- Sarò lieto di ricevere domande, suggerimenti, correzioni al mio indirizzo di e-mail, mayhem@spippolatori.org