Linux@evening

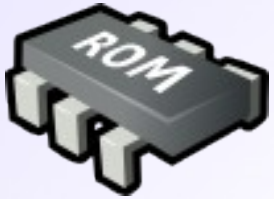
seminari di informatica libera

solira

Struttura ed Evoluzione di Dischi e Filesystem

massimo maiurana

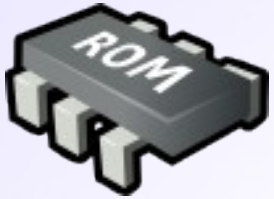
Aree di dati presenti in un computer

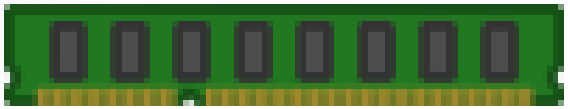


EEprom/CMOS

- saldata su scheda madre
- contiene il BIOS
- programmabile
- non volatile
- con batteria

Aree di dati presenti in un computer





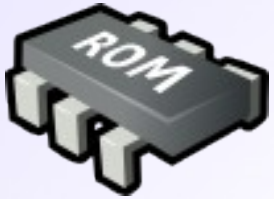
EEprom/CMOS

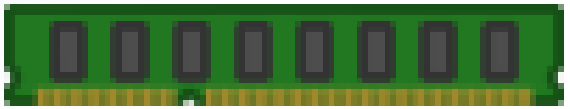
- saldata su scheda madre
- contiene il BIOS
- programmabile
- non volatile
- con batteria

Memoria centrale

- accesso casuale
- veloce
- costo elevato
- volatile
- memoria di lavoro

Aree di dati presenti in un computer





EEprom/CMOS

- saldata su scheda madre
- contiene il BIOS
- programmabile
- non volatile
- con batteria

Memoria centrale

- accesso casuale
- veloce
- costo elevato
- volatile
- memoria di lavoro

Memorie di massa

- capienti
- economiche
- lente
- non volatili

Le memorie di massa

I primi dispositivi di memorizzazione di massa, oltre alle schede perforate, furono i nastri magnetici ad accesso sequenziale.
Dalle grandi bobine si è passati a cartucce più piccole, ma in seguito il loro uso prevalente è diventato il backup dei dati, fino ad andare praticamente in disuso a seguito dell'abbassamento dei costi dei dischi fissi.
La tecnica di memorizzazione a disco ha migliorato notevolmente la velocita' di accesso ai dati.
Il futuro sembra essere dei dispositivi a stato solido.

I dischi

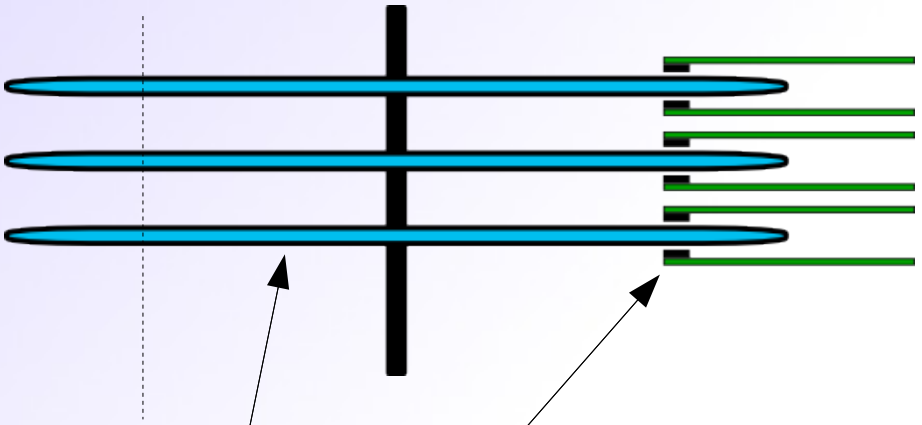
I dischi hanno il vantaggio di consentire tempi di accesso simili a dati che si trovano in diversi punti del disco.
Possiamo classificarli in funzione della tecnica di memorizzazione come:

Dischi magnetici, letti e scritti da testine

Dischi ottici, letti e scritti da laser

Dischi ibridi, scritti da testine dopo riscaldamento

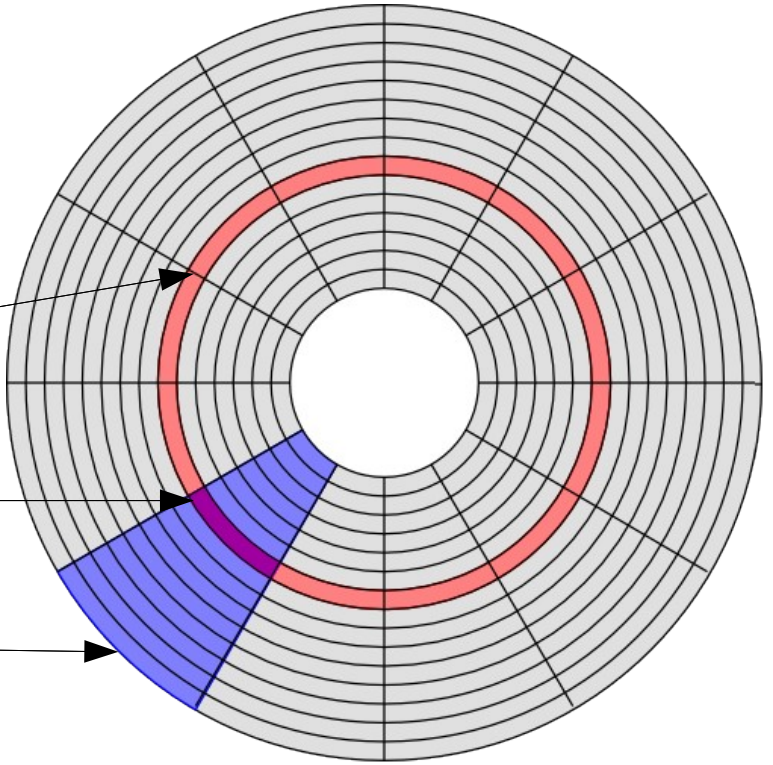
Struttura di un disco fisso



Cilindro

Piatto

Testina



Traccia

Blocco

Settore

L'indirizzamento dei blocchi può avvenire in due modi:
- vecchio metodo CHS (cylinder/head/sector)
- nuovo metodo LBA (logical block addressing)

Il disco può inoltre essere suddiviso in unita' logiche dette partizioni. Il primo blocco (Master Boot Record) contiene il codice di base del bootloader e la tabella delle partizioni primarie (max 4).
Ogni partizione primaria può a sua volta essere una partizione estesa, cioè un contenitore per ulteriori partizioni dette logiche, e in tal caso ospita nel primo blocco una tabella di queste (max 32).
Ogni partizione primaria o logica puo' a sua volta essere avviabile, cioè riservare il primo blocco all'avvio del sistema operativo.

Il filesystem

Il filesystem si occupa della gestione dei dati nel disco.
Dai primi filesystem contenenti unicamente file si è passati a filesystem strutturati ad albero, dove i dati vengono organizzati gerarchicamente.

Il blocco dati del disco (512 byte) non coincide con l'unità minima di allocazione del filesystem, in quanto al momento della formattazione i blocchi fisici vengono raggruppato in blocchi logici detti “cluster”.
La dimensione dei cluster influisce sulla frammentazione interna del filesystem, quindi occorre scegliere una dimensione adatta all'uso che se ne intende fare.

Il filesystem

Per ragioni di efficienza i blocchi non vengono cancellati subito, ma vengono marcati come disponibili.

Esistono utilità che consentono quindi di recuperare dati rimossi accidentalmente dal filesystem ma non ancora sovrascritti, e altre che consentono di eliminare ogni traccia di un file riscrivendo più volte i blocchi che lo contenevano.

Scambio dati tra filesystem e RAM

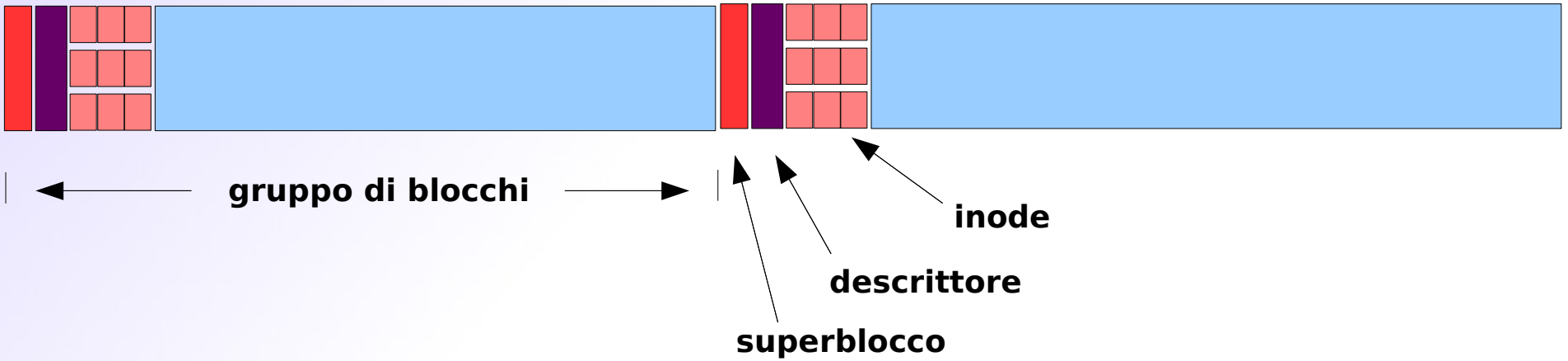
Quando dobbiamo lavorare su un file questo viene copiato nella memoria centrale, e le modifiche apportate si rifletteranno solo sulla copia in RAM.
Periodicamente avviene una sincronizzazione tra la copia su disco e la copia in RAM. E' tuttavia possibile montare un filesystem in modalità sincrona.

La RAM contiene, in aree temporaneamente libere, i dati del disco ad accesso piu' frequente.
Il disco contiene delle piccole partizioni, o dei file, destinati a contenere dati della RAM nel caso questa sia piena.

Il filesystem FAT (file allocation table)

- molto semplice e ancora usato in piccoli dispositivi
- un'unica tabella di allocazione in RAM
- copie della tabella nella stessa regione
- difficile compromesso tra frammentazione interna e occupazione di memoria
- attributi salvati nei file
- eccessiva frammentazione esterna

Il filesystem ext2



- suddivisione in gruppi di blocchi
- un superblocco e un descrittore per gruppo
- tabelle inode nei group descriptor
- inode preallocati contenenti metadati
- occupazione di memoria contenuta
- allocazione preferenziale nei gruppi
- frammentazione esterna contenuta

Oggetti in un filesystem UNIX

Ogni oggetto all'interno del filesystem viene referenziato in un inode, quindi ad ogni interrogazione del filesystem corrisponde il caricamento in memoria degli inode contenenti le informazioni che servono per arrivare alla risorsa.

I primi oggetti che si incontrano sono le directory, ovvero i rami che compongono l'albero del filesystem.
Le directory contengono solamente riferimenti ad altri inode che a loro volta referenziano altre directory o file.

Oggetti in un filesystem UNIX

L'oggetto finale della ricerca e' il contenitore dei dati, cioè il file. Uno stesso file può avere più riferimenti (hardlink) nello (o negli) inode , quindi è possibile risalire fino al file da percorsi diversi all'interno del filesystem. Il file verrà rimosso solo quando l'ultimo hardlink verrà rimosso.

Il collegamento simbolico, o symlink, è invece un file che contiene un riferimento al percorso di un altro file, che quindi puo' trovarsi anche in un filesystem diverso.
Se il file di destinazione viene rimosso il symlink rimane orfano.

Oggetti in un filesystem UNIX

Esistono poi una serie di file speciali che vengono usati in ambito UNIX, di solito dai processi e non dall'utente:

- Le pipe, dei semplici canali di dati tra processi
- I socket, più evoluti delle pipe e full-duplex
- I dispositivi a caratteri
- I dispositivi a blocchi

Il tipo di file viene evidenziato con una lettera all'inizio dell'output del comando “ls” (d, -, l, p, s, c, b).

Metadati in un filesystem UNIX

Negli inode vengono salvati i seguenti attributi:

- la posizione fisica dell'oggetto nel disco
- la dimensione del file
- il proprietario e il gruppo
- i permessi
- le date di creazione/modifica/accesso
- gli hardlink, cioè i nomi con cui è conosciuto

Metadati in un filesystem UNIX

E' possibile anche utilizzare delle apposite estensioni per i più comuni filesystem utilizzati in ambiente Linux:

- gli attributi estesi:
  impostano determinate proprietà del file come la sua immutabilità, la modalità di scrittura, la compressione, ecc..
- le ACL (access control list):
  permettono di specificare maschere di permessi
- le security labels di SELinux
  degli attributi di sicurezza sviluppati dalla NSA che possono anche essere applicati al montaggio e non risiedere nel FS
- le quote disco
  limiti per utente/gruppo all'occupazione del filesystem

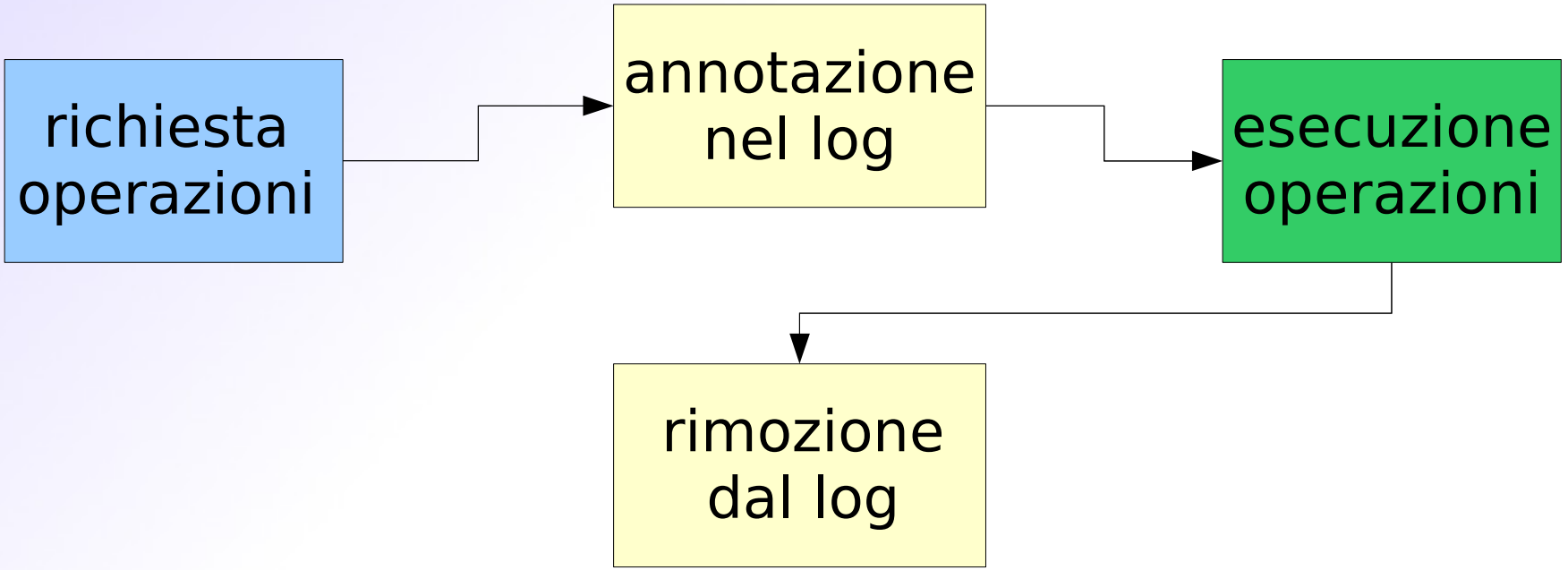
Ops... è andata via la luce!

Che succede a seguito di un arresto repentino del sistema?

Se i buffer non sono stati sincronizzati col disco perdiamo le ultime modifiche. Per le unità rimovibili è necessario richiedere esplicitamente la sincronizzazione.

Se l'arresto avviene proprio mentre è in corso una sincronizzazione si può andare da una corruzione dei dati, risolvibile con appositi strumenti di correzione, ad una corruzione dei metadati con conseguente incoerenza del filesystem.

I filesystem “journaled”



Questi filesystem, come ad esempio ext3 o NTFS, mantengono sul disco un diario delle transazioni che, anche se non e' in grado di garantire i dati, permette di riportare velocemente il filesystem ad una situazione di coerenza al costo di qualche accesso in più al disco.

Il filesystem “Extended”

Il primo filesystem utilizzato in Linux fu ereditato da Minix. Nel '92 fu adottato ufficialmente il filesystem ext, sostituito quasi subito da una versione potenziata destinata a durare parecchio: ext2.

Il passaggio da ext2 a ext3 fu praticamente indolore, poiche' il formato su disco è identico e basta aggiungere il journal per passare da ext2 a ext3. I tool di gestione sono gli stessi.

Il recente ext4 non offre compatibilità piena; il formato è diverso, e alcune funzionalità possono essere sfruttate solo in un filesystem creato ex-novo.

Novità nel filesystem ext4

- Extent:

È un gruppo di blocchi con proprio indirizzo, utile con grossi file per risparmiare riferimenti

- Allocazione parallela multiblocco:

Un metodo per allocare più dati con un'unica chiamata

- Allocazione ritardata:

L'allocazione dei blocchi avviene alla scrittura per evitare operazioni inutili

- Deframmentazione in linea
- Preallocazioni persistenti
- Controlli integrità più veloci

Il futuro per Linux

Al momento il filesystem di grido nel mondo UNIX è il ZFS di Sun, ma per motivi di licenza non verrà incluso in Linux e potrà essere utilizzato solo tramite FUSE.

Oracle ha però avviato lo sviluppo di un filesystem con simili caratteristiche ma con licenza GPL: il Btrfs, o ButterFS.

Questi sono filesystem a 128-bit di tipo copy-on-write, che cioè non sovrascrivono i blocchi. Possono così creare delle istantanee per il ripristino di metadati e dati, e anche creare dei cloni scrivibili dei file.

Altre caratteristiche di Btrfs

- Possibilità di variare la dimensione del FS e dei blocchi
- Possibilità di eseguire controlli al volo
- Compressione trasparente
- Sottovolumi montabili separatamente
- Aggiunta di dischi e inode
- Mirroring e/o striping per oggetto
- Checksumming di dati e metadati

Non solo dischi

Un filesystem può essere creato in qualunque dispositivo a blocchi. Anche un file puo' contenere un filesystem, e in questo caso si parla di immagini.

Per creare il file contenitore:

```
# dd if=/dev/zero of=./file.img bs=1k count=1440
```

Per creare il filesystem:

```
# mke2fs -F ./file.img
```

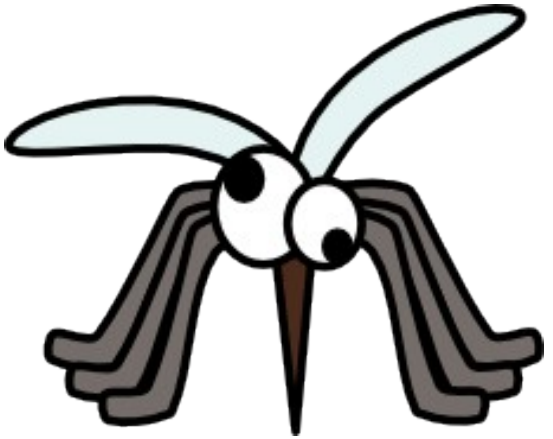
Per montare il filesystem:

```
# mount -o loop -t ext2 ./file.img /mnt/loop
```

Redundant Array of Independent Disks

Il RAID è un sistema, hardware o software, ideato negli anni '80 per utilizzare dispositivi economici in batteria, eventualmente anche con hot-spare, allo scopo di aumentare le prestazioni e/o l'affidabilità .

RAID 0	**striped**
RAID 1	**mirrored**
RAID 4	**parity disk**
RAID 5	**distributed parity**
RAID 10	**1 + 0**



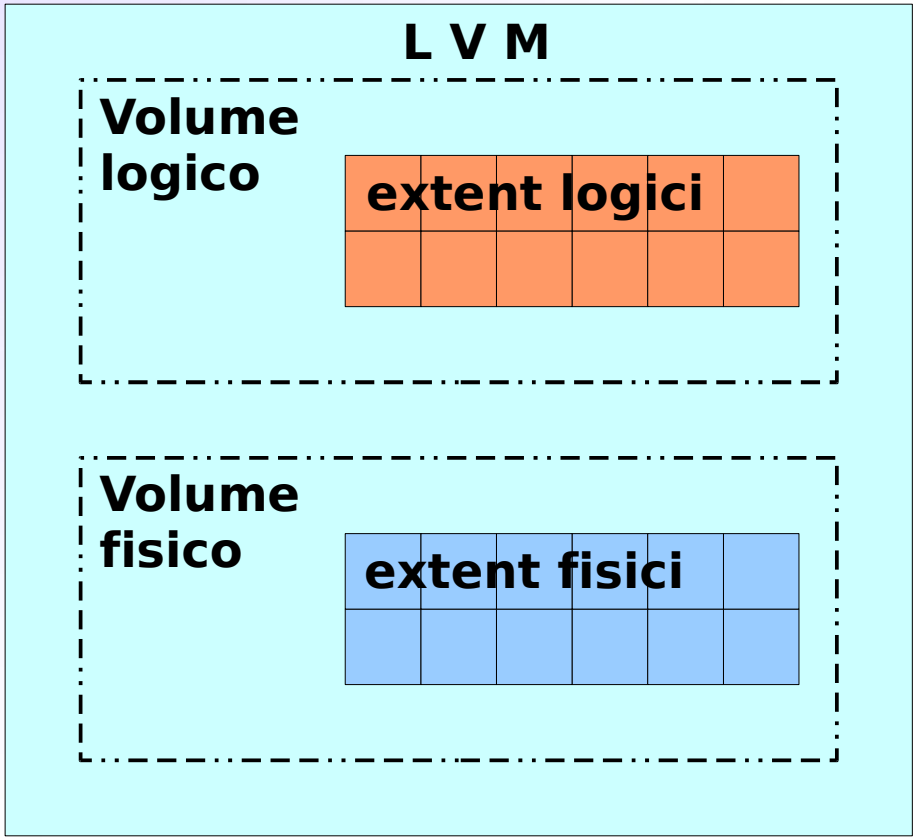
La parità dei byte

La parità dei dati è un meccanismo matematico che consente di verificare, ed eventualmente correggere, un errore.

A livello di byte, viene aggiunto un bit di parita': in caso di parità pari, il bit viene posto a 1 se la somma dei bit di valore 1 restituisce un numero pari.

Es: se il payload è 1001101, il bit di parità pari è 1
se ci arriva 10-11011 sappiamo che il bit mancante è 0

Logical Volume Management



LVM permette di gestire in modo flessibile le unità logiche. Ogni unità viene suddivisa in extent di pari dimensione, e viene mantenuta una mappa di corrispondenza tra unità logiche e fisiche.

Il sistema può essere ridefinito in ogni momento per ragioni di manutenzione, ad esempio per riditribuire lo spazio tra unità logiche e/o utenti. Alcune funzionalità dipendono comunque dal FS soprastante.

Limitazioni di RAID e LVM

- Il boot loader potrebbe non supportarli, quindi è preferibile una partizione d'avvio.

- Per usare una partizione radice dentro un dispositivo multiplo potrebbe essere necessario un initram.

- Si possono ridimensionare a piacimento i volumi, ma ridimensionare i filesystem può essere complicato.

Filesystem in rete

Un filesystem può anche non essere locale, cioè può non risiedere sulla nostra macchina ma essere distribuito all'interno di una rete.

Il filesystem di rete lavora al di sopra di filesystem locali in modo del tutto trasparente agli utenti.

Centralizzando alcune risorse su macchine dedicate è possibile risparmiare sui costi di amministrazione, ad esempio grazie all'adozione di sistemi RAID.

L'accesso alle risorse avviene in genere a seguito di autenticazione, e in alcuni casi è possibile gestire accessi concorrenti e/o utilizzare dati crittografati.

Filesystem in rete

I filesystem di rete “storici” sono NFS in ambienti UNIX e SMB/CIFS in ambienti Microsoft, poco scalabili ma spesso usati anche solo per condividere risorse in rete.

Per sistemi più grandi ci sono filesystem come AFS, che ha introdotto interessanti funzionalità per aumentare la scalabilità (suddivisione in celle e volumi, migrazione facilitata, caching locale), e il nuovo GlusterFS, filesystem adattabile a contesti piccoli o grandi. GlusterFS è modulare, adotta metodi simili a quelli del RAID (striping, mirroring), e la struttura viene definita sui client per mezzo dei translator.

Filesystem per applicazioni specifiche

- per dischi ottici (ISO-9660, UDF)
  scrittura sequenziale, assenza di frammentazione esterna, possibilità di aggiungere dati in più sessioni.

- per memorie flash (JFFS, exFAT)
  distribuzione uniforme dei dati, correzione degli errori.

- per dischi virtuali (tmpfs, initramfs)
  allocati in RAM e usati, ad esempio, per rendere disponibili driver all'avvio del sistema.

- pseudo-filesystem (sysfs, devfs)
  interfacce del kernel che non rappresentano dati su disco.