QUALE FILE SYSTEM USARE PER LINUX?

Ext2, Ext3 o ReiserFS? Ce n'è qualche altro? E Fat? Ma... cos'è poi un File System?

Un File System è il metodo con il quale i dati vengono organizzati all'interno di un dispositivo di memorizzazione (Hard Disk, CD, DVD, Floppy, memoria USB...)
Detto in termini più semplici, esso è l'interprete tra i dati da registrare e la loro disposizione sul disco. E' facile intuire quanto ciò influisca sulle prestazioni del sistema: per questo, la sua scelta ricopre un ruolo fondamentale.

I VARI FS – Storia, pregi e difetti

FAT

Chi non ha mai usato il Defrag di Windows? E perché bisogna deframmentare?
Le partizioni di tipo FAT (Windows 9x o più recenti) registrano i file a blocchi non necessariamente contigui, e ciò porta ad avere un file suddiviso in tanti piccoli frammenti sparsi. Poiché l'hard-disk viene letto da una testina a braccio mobile (come un... giradischi), questa deve leggere il file nella posizione iniziale, alzarsi, riposizionarsi nella posizione successiva, leggere, e così via fino al completamento della procedura. In tutta l'operazione, la maggior parte del tempo viene “sprecata” nello spostamento della testina. Perciò, in un file System di tipo FAT, si perde del tempo prezioso e i meccanismi interni sono sottoposti a maggiori sollecitazioni. L'avvento di NTFS ha ovviato in parte a questi problemi.

II File System FAT (File Allocation Table), conosciuto anche come MSDOS, è parte della Storia dei sistemi operativi di casa Microsoft, ed è sempre stato afflitto da molte pecche. Le versioni iniziali (FAT 12 e FAT 16) supportavano una dimensione limitata per i dischi fissi (rispettivamente 250 MB e 4 GB) e utilizzavano solo undici caratteri per i nomi dei file: otto per il nome e tre per l'estensione. Inoltre, nei dischi fissi più “grandi” i cluster erano di eccessive dimensioni, al punto da specare molto spazio per l'archiviazione dei files piccoli. Mille files da 1kb potevano occupare anche 16 mb e oltre!

La prima versione di Windows 95, enormemente pubblicizzata come “la svolta” nel campo dei pc-home, girava proprio su FAT16. L'evoluzione di questo sistema, ossia FAT32, ha risolto il problema della dimensione, aumentando il supporto dei dischi fino a 2 TB, mentre l'introduzione del vFAT ha esteso la lunghezza dei nomi a 255 caratteri. Una partizione di questo tipo, per mantenere delle buone prestazioni, necessita di deframmentazioni periodiche, in quanto i blocchi che compongono i files possono essere sparsi per tutto il disco, rallentando notevolmente le operazioni di lettura.

Il vantaggio del file system FAT consiste nel fatto che è ormai talmente diffuso da essere riconosciuto praticamente da tutti i sistemi operativi. L'utilizzo di Linux in combinazione con una partizione FAT non è sempre da disdegnare; una partizione FAT può essere utilizzata per scambiare dati tra Linux e Windows (in dual boot), oppure per mantenere una partizione Windows 98 per retrocompatibilità. Esiste addirittura un file system variante di FAT, chiamato UMSDOS, che permette di installare sia Windows che Linux sulla medesima partizione, senza driver aggiuntivi - anche se con pesanti limitazioni prestazionali rispetto ad un file system nativo.

L'altro file system di casa Microsoft è NTFS (NT, 2000, XP), un file system più recente dotato di una struttura più moderna, che limita la deframmentazione e, soprattutto,

permette la gestione avanzata dei permessi dei file. La versione presente in Windows XP di NTFS è dotata di un sistema di *journaling*.

LINUX
I File System di Linux, invece, sono progettati per registrare tutto il file con una frammentazione minima. Anche se il file viene suddiviso all'interno del disco, si cerca di utilizzare blocchi adiacenti o in sequenza, ove possibile. Questo apporta un notevole incremento di velocità, ed il vantaggio di non dover mai deframmentare il disco (anche se tale operazione è teoricamente possibile anche in Linux). Anche al Polo Nord, teoricamente, è possibile ghiacciare l'acqua in frigorifero anziché lasciarla ghiacciare fuori...

APPLE
Per quanto riguarda Apple, essa iniziò utilizzando il file System MFS (Macintosh File System) fino alla versione 4.1 del Finder, rimpiazzandolo nel 1985 con il celebre HFS (Hierarchical File System). HFS, pur avendo molte caratteristiche comuni a FAT, permetteva di usare fino a 31 caratteri per i nomi dei file (contro gli 8+3 del FAT), supportava i *metadati* (attributi del file), e introduceva il *dual-fork*. Il successore, HFS+, è comparso a metà degli anni '90 (MacOS 8.1), e annovera tra le sue caratteristiche il supporto per file più grandi, l'uso di Unicode e l'estensione fino a 255 caratteri per i nomi dei file. Dal 2003, con il Mac OS 10.3, ad HFS+ sono state aggiunte le caratteristiche di journaling (concetto che svilupperemo nel corso di questo talk). Apple ha deciso di non cambiare il File System per evitare incompatibilità tra le varie versioni del Mac.

IL JOURNALING

Cosa accade quando avviene un blocco di sistema durante la copia un file importante? In un sistema Fat-Like è andato perduto. Oltre al danno, l'utente subisce anche la beffa di dover aspettare vari minuti la scansione dell'intero disco al riavvio.

Nei file System tradizionali PRIMA vengono scritti i dati sul disco, POI viene aggiornato l'indice dei file contenente nome, dimensioni, data e attributi. Un crash tra la fase di scrittura e quella di indicizzazione porta alla perdita matematica dei dati scritti, quasi senza possibilità di recupero. Inoltre, i programmi di ripristino dovranno impiegare molto tempo, proporzionale inversamente alla potenza del computer e direttamente alla dimensione del disco, per analizzare l'intera superficie alla ricerca di errori.

Per ovviare alla perdita di informazioni e alle lunghe attese per la riparazione del file System, è nato il journaling, letteralmente “diario”. Il funzionamento è tanto semplice quanto intuitivo: una zona del disco contiene informazioni specifiche e viene chiamata “Journal”, e su di essa si basano le operazioni di scrittura, che per comodità dividiamo un quattro fasi.

- Fase 1: annotazione PREVENTIVA nel diario delle operazioni pendenti
- Fase 2: scrittura FISICA dei dati
- Fase 3: aggiornamento dell'indice
- Fase 4: cancellazione dal diario delle operazioni completate.

Se il blocco del computer dovesse avvenire durante una fase di scrittura, si profilano quattro scenari:
- II blocco avviene durante la prima fase. Al riavvio, il Journal viene controllato per primo e, non essendo ancora stati scritti dati, viene cancellato qualsiasi riferimento ai files. Tutto torna a posto, ed è come se il comando di scrittura non fosse mai stato impartito. Il

file è andato perduto, tuttavia ciò sarebbe avvenuto comunque. *Si ha però un guadagno sul tempo impiegato per il ripristino, in questo caso di pochi secondi.*
- II blocco si verifica durante la seconda fase. Al riavvio, si legge il Journal e si verifica cosa sia stato scritto e cosa no. Di conseguenza, si aggiorna l'indice dei file e si toglie il riferimento dal diario. *Si perde ciò che non è ancora stato scritto, senza inutili attese, e si recupera ciò che effettivamente era stato salvato.*
- II blocco avviene nella terza fase. L'unica cosa che deve essere fatta è quella di leggere il diario e aggiornare l'indice di conseguenza. *Il file è interamente salvo e l'operazione richiede pochi secondi per essere completata.*
- II blocco avviene durante l'ultima fase. In questo marginale caso, il Journal viene cancellato; *anche questa operazione richiede pochi secondi.*

Nel primo caso si perdono i dati, nel secondo vi è una perdita parziale e negli ultimi due non si perde nulla: un bel passo in avanti rispetto alla perdita sicura nei sistemi FAT.

Tanta sicurezza si paga in termini di prestazioni. I file system con journaling perdono intorno al 10% e il 15% di performance. Ciò è dovuto al maggior lavoro sul disco. Tuttavia, le ultime evoluzioni in materia di FS fanno ben sperare. I maggiori FS journaling sono Ext3, ReiserFS, JFS e XFS.

SISTEMI *NIX

UFS (UNIX File System) è derivato dal Fast File System (FFS) di Berkley, e, ad oggi, viene usato in FreeBSD, NetBSD, OpenBSD, NeXTStep e Solaris; in Mac OS X è disponibile in alternativa all'HFS. Il file system Ext2, nativo di Linux, è stato scritto “from scratch” (da zero), prendendo come base l'UFS e introducendo ovviamente delle migliorie. Con Solaris 7 (2002 – Sun) ha visto la luce l'UFS Logging, una versione UFS con Journaling. FreeBSD 5 (2003) ha introdotto l'UFS2, una versione migliorata che, tra le varie caratteristiche, aggiunge il supporto per volumi superiori ad un terabyte.
II BFS è invece il file system nativo dello sfortunato BeOS, del quale oggi esistono versioni Open, scritte da zero, come OpenBeOS. Il BFS era un file system con journaling, nato a metà degli anni novanta, quando Windows presentava ancora FAT32. BeOS fu certo sfortunatoto, ma precorse i tempi e le tendenze. Il BFS è famoso per la sua gestione avanzata degli attributi: ai file possono essere associati dei meta-dati avanzati per poi poter ricercare tali file più velocemente tramite una parola chiave o, cosa molto più interessante, per creare delle cartelle speciali (smart folder), che visualizzano una collezione di file corrispondenti a determinati criteri (ad esempio tutti i documenti di Open Office presenti nel computer).

LINUX: EXT

Uno dei primi FS di Linux è stato Ext, sostituito da Ext2, “written from scratch” - da zero - prendendo come modello l'UFS. Ext2, oltre ad essere uno dei file system più veloci oggi disponibili, ha introdotto il supporto per volumi fino a 4 TB e la gestione dei nomi lunghi e il supporto completo ai file Unix.

Per poter essere compatibile con future versioni, utilizza gli *hook* (agganci), una sorta di plug-in che permette di estenderne le funzionalità, preservandone la struttura base. Per questo si è facilmente sviluppato Ext3, ovvero Ext2 con journaling. Ext3 è marginalmente penalizzante in termini di prestazioni, dovendosi portare dietro la struttura di Ext2 – ma esiste la possibilità di abilitare il journaling per i soli metadati, o per la copia dei files durante la fase iniziale di aggiornamento del diario, guadagnando così velocità.

Sta di fatto che i file system di tipo ext2 o ext3 sono lo standard per la maggior parte delle distribuzioni.

REISERFS
ReiserFS è un file System con journaling sviluppato dal team di Hans Reiser. Lavora usando particolari metadati associati ai file, cosa che gli permette di recuperare i file dopo eventuali blocchi di sistema con una velocità e un'affidabilità superiore ad altri file system. ReiserFS è disponibile in Linux a partire dal kernel 2.4.1 ed è così valido da essere adottato come default da molte distribuzioni.
Il suo più grande vantaggio consiste nel non essere legato a tecnologie precedenti. Per converso, proprio in virtù di questo, per convertire una partizione Ext2, è necessario fare il backup dei dati, quindi riformattare il tutto nel nuovo formato e ricopiare i dati al suo interno.

Attualmente è in testing il Reiser4, suo successore, riscritto da zero: le sue qualità sono una maggiore velocità, il supporto "improved" per la gestione di cartelle contenenti numerosi file, l'ottimizzazione del sistema di journaling, l'integrazione dei metadati all'interno dello spazio dei nomi dei file, il supporto per i plug-in e molte altre.

JFS
JFS è un file system di journaling sviluppato da IBM, utilizzato inizialmente per il celebre OS/2 e poi per AIX, lo Unix di IBM.
Questo sistema è stato "regalato" alla comunità Open Source nel Febbraio 2000, e risulta integrato in Linux a partire dal kernel 2.4. Tra le particolarità, spicca il sistema di journaling basato su una tecnica di log concepita per i database: questo significa che, per riparare la partizione, il sistema legge soltanto i log più recenti, invece dei metadati di tutti i file.

Altra importante innovazione è l'uso di blocchi a dimensione variabile: ciò influisce significativamente sul livello di deframmentazione e sulla velocità del disco. Infine, vengono introdotte l'allocazione dinamica dell'*inode*, una diversa organizzazione in base alla grandezza delle directory, e un'allocazione particolare dei file.

XFS
XFS è un file System journaling dalle prestazioni elevate, creato da SGI (Silicon Graphics Incorporated) per Irix, implementazione proprietaria di Unix. Nel maggio 2001, SGI ha rilasciato XFS sotto licenza Open Source, ed è disponibile in Linux a partire dal kernel 2.5, 2.4.x patched e per FreeBSD tramite porting.

Il motto di XFS è "Think Big", inteso in termini di affidabilità e prestazioni. A questo è diretta l'implementazione di una particolare tecnologia definita "Allocation Group", che consiste nel dividere il disco in otto o più regioni autonome e di uguali dimensioni, in modo da poter lavorare su ognuna simultaneamente. XFS utilizza un sistema di Journaling molto più sicuro e performante, implementando una tecnica chiamata Delayed Allocation.

TIRANDO LE SOMME...
Non esiste un File System migliore: dipende tutto dall'utilizzo che se ne fa. Il file System Ext3 è compatibile con il suo predecessore Ext2, e addirittura può essere trasformato (reversibilmente!) in Ext2 usando il comando tune2fs.
Il formato ReiserFS rappresenta uno standard, con delle ottime caratteristiche e delle prestazioni molto buone. Reiser4, pur avendo prestazioni eccezionali e migliorie rilevanti, è ancora troppo poco maturo per poter essere ampiamente supportato.
Tra i due restanti, quello che si fa notare per una maggiore affidabilità e velocità, è

sicuramente XFS, più maturo di JFS.

E' utile osservare come lo sviluppo dei File System si sia reso parallelo a quello dei database: questa probabilmente è la chiave di sviluppo per il futuro dei FS. La velocità di ricerca di informazioni, la possibilità di associare a un file diverse viste, in base a qualche parola chiave o al suo contenuto, rappresentano un obiettivo interessante.

Programmi che permettono di "lavorare" con i file systems del vostro PC
-QTParted – http://qtparted.sourceforge.net
-GNU Parted - www.gnu.org/software/parted
-DiskDrake - www.mandrakelinux.com/diskdrake

Quest'ultimo è attualmente insuperabile, è utilizzabile facendo il boot dal primo di CD d'installazione della distribuzione Mandrakelinux.

Testo curato da Andrea Serrajotto per il MontelLUG – andreaserra@freesurf.fr