Sistemi Operativi III

Giorgio Richelli

e-mail: giorgio_richelli@it.ibm.com

Il file system NTFS

Un volume NTFS consiste di 4 regioni:

1. Partition boot sector.
2. Master File Table.
3. System files.
4. File area.

- Quando viene formattato un volume NTFS viene stabilito il formato del *cluster* che è sempre un multiplo intero del settore fisico. In questo modo viene mantenuta indipendenza dalla dimensione del settore (che in genere è 512 byte) e vengono utilizzati in maniera più efficiente dischi di grandi dimensioni.
- NTFS accede il disco facendo riferimento ai *Logical Cluster Numbers* (LCN). Per convertire un LCN in un indirizzo fisico è sufficiente moltiplicare il LCN per il *cluster factor*.
- I dati appartenenti ad un file sono acceduti tramite i *Virtual Cluster Numbers* (VCN).
  I VCN non sono necessariamente contigui a livello fisico.
- La directory entry di un file contiene le informazioni sulla corrispondenza fra VCN ed LCN

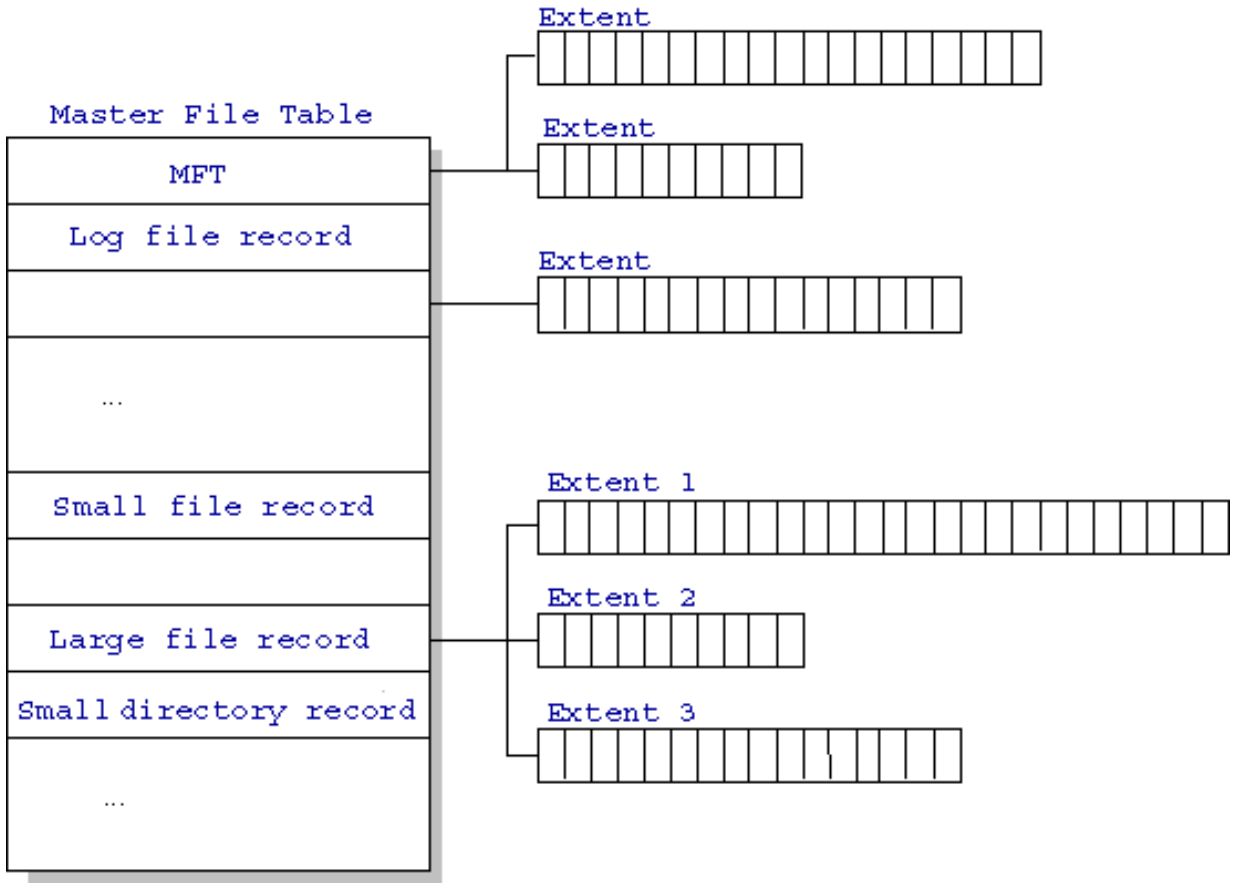
Nel NTFS ogni componente di sistema è un file, comprese le informazioni sul file system stesso quali:

- Strutture usate per localizzare i file;
- Dati per il *bootstrap*;
- *bitmap* che registra lo stato di allocazione dell'intero volume.
- ...

Il file più importante del NTFS è il Master File Table (MFT).

Il contenuto della MFT si può analizzare con il tool **nfi** disponibile da `http://support.microsoft.com/support/kb/articles/Q253/0/66.asp`

- la MFT è una lista di tutti i contenuti del volume NTFS (*directory* di tutti gli altri file appartenenti al volume) organizzata come un insieme di *righe* in un database relazionale.

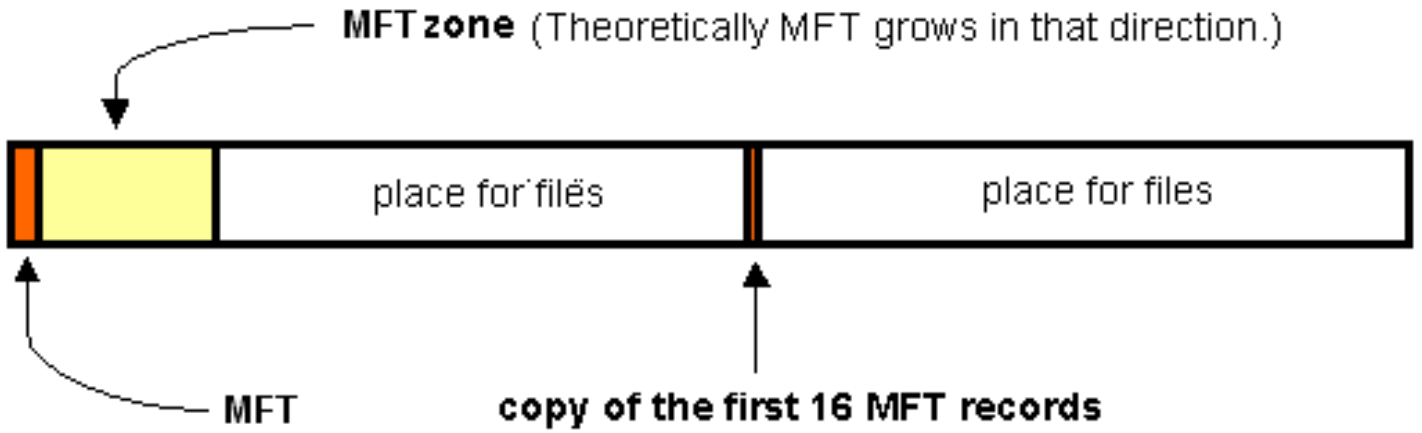

Master File Table
MFT
Log file record
...
Small file record
Large file record
Small directory record
...
Extent
Extent
Extent
Extent 1
Extent 2
Extent 3

- La dimensione della MFT è definita nel primo record che descrive la MFT stessa;
- Ogni record ha un numero di riferimento (simile ai numeri di *inode* sotto UNIX).
- Ogni *record* (*riga*) della tabella descrive un file oppure una "cartella";
  - La dimensione del record viene letta dal boot sector.
- Un file sufficientemente piccolo (*e.g.*, $< 1KB$ byte) è interamente contenuto in una riga del MFT.
- NTFS riserva i primi 16 record per informazioni speciali;
- I file utenti iniziano al record #27.

Segment Number	File Name	Purpose
0	`$MFT`	Describes all files on the volume, including file names, timestamps, stream names and lists of cluster numbers where data streams reside, indexes, security identifiers, and file attributes like "read only", "compressed", "encrypted", etc.
1	`$MFTMirr`	Is a duplicate of the first vital entries of $MFT, usually 4 entries (4 KiB).
2	`$LogFile`	Contains transaction log of file system changes for metadata consistency.
3	`$Volume`	Contains information about the volume, namely the volume object identifier, volume label, file system version, and volume flags (mounted, chkdsk requested, requested $LogFile resize, mounted on NT 4, volume serial number updating, structure upgrade request). The volume serial number is in $Boot file.
4	`$AttrDef`	A table of NTFS attributes used with names, numbers and descriptions.
5	.	Root directory.
6	`$Bitmap`	A table of bit entries representing if particular cluster on the volume is used or free.
7	`$Boot`	Volume boot record. This file located at first cluster on the volume includes bootstrap code (used to find and launch NTLDR/ BOOTMGR and a BIOS parameter block including volume serial number and cluster numbers of $MFT and $MFTMirr.
8	`$BadClus`	A file which contains all the clusters marked as having bad sectors. This file simplifies cluster management by the chkdsk utility, both as a place to put newly discovered bad sectors, and for identifying unreferenced clusters.
9	`$Secure`	Access control list database which reduces overhead having many identical ACLs stored with each file, by uniquely storing these ACLs in this database only (contains two indices $SII: perhaps[*citation needed*] *Security ID Index* and $SDH: *Security Descriptor Hash* which index the stream named $SDS containing actual ACL table).[25]
10	`$UpCase`	A table of unicode uppercased characters for ensuring case insensitivity in Win32 and DOS namespaces.
11	`$Extend`	A filesystem directory containing various optional extensions, such as $Quota, $ObjId, $Reparse or $UsnJrnl.
12 ... 23	Reserved.	
usually 24	`$Extend\$Quota`	Contains information regarding disk quotas.
usually 25	`$Extend\$ObjId`	Contains information used for distributed link tracking.
usually 26	`$Extend\$Reparse`	Contains backreferences of all reparse points (such as symbolic links) on the volume
27 ...	*`file.txt`*	Beginning of regular file entries.

- il primo record (`$MFT`) descrive l'MFT stesso seguito dal *mirror* record dell'MFT (`$MFTmirr`);
  - se il primo record MFT è corrotto, NTFS legge il secondo record per individuare il mirror file del MFT;
- il terzo record (`$LogFile`) contiene il *log file* usato per il *recovery*;
- il quarto record (`$Volume`) contiene informazioni come il nome del volume, la versione del file system, *etc.*;
- il quinto record (`$AttrDef`) definisce il tipo di attributi supportati;

- il sesto record (\) contiene la directory *root*;
- il settimo record (`$Bitmap`) contiene la bitmap dello spazio libero sul volume;
- il nono record (`$Secure`) contiene descrittori di sicurezza validi per tutti i file del volume;



- Lo spazio disco é logicamente diviso in due parti
  - Circa il 12% é riservato all'MFT
  - Il resto é dedicato al resto dei file

Ogni file è rappresentato da un record nel MFT ed è costituito da una collezione di "attributi".

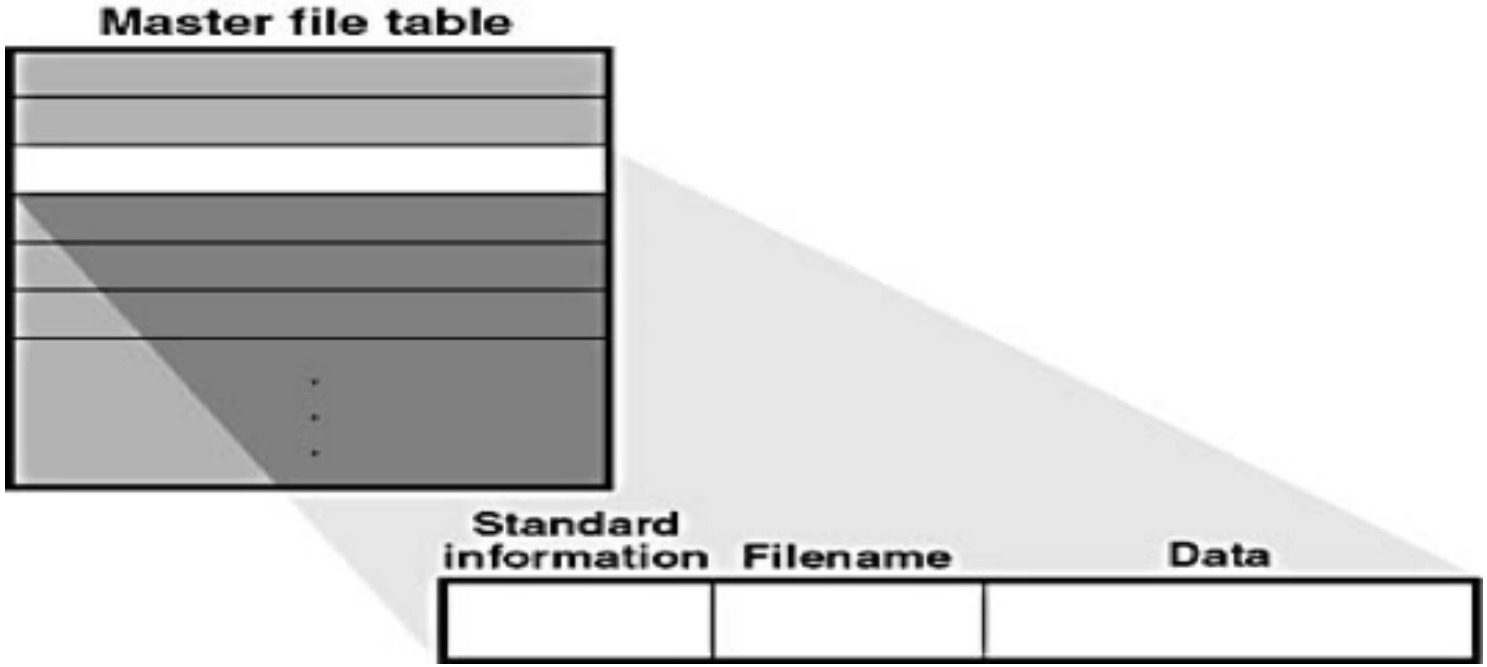
- Se un file ha un numero elevato di attributi viene allocato più di un record.
  - Anche i dati di un file sono visti come un attributo.
- Ogni attributo ha un header ed un valore che è detto "residente" se è contenuto direttamente nell'entry del MFT.
- Ad ogni tipo di attributo è associato un indice e possono esserci più istanze di un attributo per un file:
  - **Standard Information (#16):** Include informazioni come i timestamp ed il *link count*.
  - **Attribute List:** Lista tutti i record di attributi che non trovano spazio nel record MFT.
  - **File Name (#48):** Un attributo "ripetibile" (può essere presente più volte).
    Nella forma estesa il nome può arrivare a 255 caratteri Unicode.

Nella forma breve si usa il classico formato 8.3 (case-insensitive).
Contiene anche la entry nella MFT della directory che contiene il file.

- **Object ID (#64):** Un identificatore di file unico per ogni volume.
- **Security Descriptor:** Descrive il proprietario del file e chi lo può accedere.
- **Data (#128):** Contiene i dati del file.
  NTFS permette di avere più "flussi" di dati per file (*Alternate Data Stream*).
- **Logged Tool Stream:** Simile ad un data stream, ma le operazioni sono registrate nel log file del NTFS (come i cambi dei metadati).
  È utilizzato dalla funzione di cifratura.
- **Reparse Point:** Usato per il mount dei volumi.
- **Index Root (#144):** La radice del **B-Tree**. Usato per implementare cartelle ed altri indici.
- **Index Allocation (#160):** I sottonodi del **B-Tree**. Usato per implementare cartelle ed altri indici.
- **Bitmap (#176):** Fornisce una mappa che rappresenta i record in uso nel MFT o nella cartella

- **Volume Information:** Usato solo nel file di sistema `$Volume`. Contiene la versione del volume.
- **Volume Name:** Usato solo nel file di sistema `$Volume`. Contiene l'etichetta del volume.

Gli attributi sono ordinati (in ordine ascendente) rispetto ai loro codici numerici.



Il Security Descriptor contiene una o due Access Control List (ACL) e due Security Identifier (SID) che indicano il proprietario del file ed il gruppo.

- La prima ACL è opzionale e contiene informazioni per l'auditing.
- La seconda ACL contiene una o più Access Control Entry (ACE).
  - Ogni ACE indica un utente ed un'azione ("chi-può-fare-cosa")
  - Un tipico SID appare come: S-1-5-21-646518322-1873620750-619646970-1110 (non proprio banale da interpretare...).
- Il modello di sicurezza di NTFS di tipo "discrezionale" ed è più flessibile di quello standard Unix/Linux.

Per manipolare le ACL è possibile usare il comando standard CACLS. Oppure la versione "estesa" XCACLS

(`http://www.microsoft.com/windows2000/techinfo/reskit/tools/existing/xcacls-o.asp`)

Le directory nel NTFS sono indicizzate in modo da rendere più veloce l'accesso ad una entry specifica.

- Vengono ordinate in un **B-Tree** in ordine alfabetico.
- Quando i file sono cancellati da una directory, i corrispondenti nodi sono rimossi e l'albero è riordinato.
- Diverso da quanto avviene sotto UNIX o con il file system FAT sotto Windows.

Quando un file viene cancellato, il flag `IN_USE` è azzerato nella entry della MFT ma gli attributi non sono cancellati!

Nel NTFS un file può avere più flussi di dati indipendenti (*Alternate Data Stream*). Ad ogni flusso è associato un nome che identifica un nuovo attributo del file.

Gli *Alternate Data Stream* (ADS) possono essere gestiti come una singola unità: hanno lock e formati separati, ma permessi comuni.

Per creare dalla linea comandi un ADS:

```
echo text>program:source_file
more<program:source_file
```

Quando si copia un file NTFS su un volume diverso, (*e.g.*, su un floppy, penna USB, CD..) *data stream* ed altri attributi sono persi (viene comunque, in genere, inviato un warning).

Funzionamento del NTFS

NTFS legge i *metadati* dal disco e costruisce strutture dati interne per rendere accessibili i contenuti del volume all'atto del *mount*.

1. legge nel *boot sector* l'indirizzo fisico della MFT;
2. legge nel file record della MFT il mapping VCN-LCN della MFT;
3. legge i record della MFT corrispondenti ad alcuni file di sistema;
4. esegue le eventuali operazioni di *recovery*.

- Durante il normale funzionamento, NTFS registra sul *log* file ($LogFile) tutte le operazioni che modificano la struttura del volume o delle singole directory.
- NTFS registra lo stato di allocazione del volume nel file ($Bitmap). Ogni bit dell'attributo *data* rappresenta un cluster del volume.

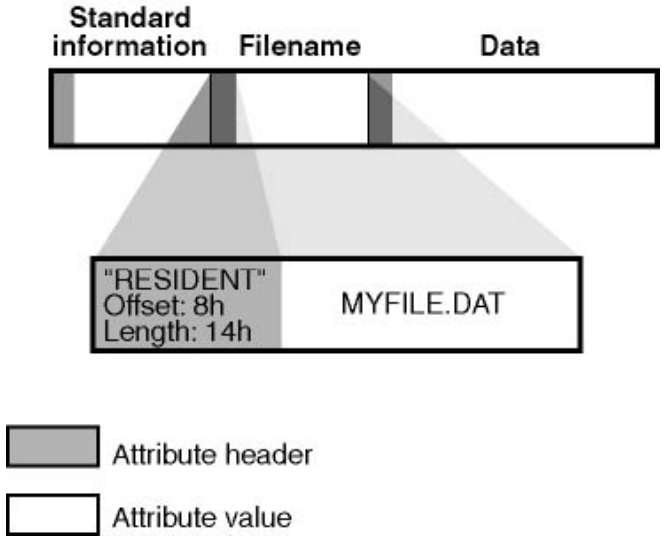
- La prima volta che apre un file NTFS inizia la ricerca nel file record della *root* directory. Successivamente, NTFS salva il numero d i *file reference* in modo da poter accedere direttamente il record del file nella MFT quando legge o scrive il file.
- Un file è identificato da un valore a 64 bit detto *file reference*.

  - il *file number* corrisponde alla posizione del record del file nella MFT meno 1;
  - il *sequence number*, che viene incrementato ogni volta che una entry della MFT è riutilizzata, permette di eseguire check di consistenza interna.
- NTFS genera automaticamente nomi alternativi nella forma

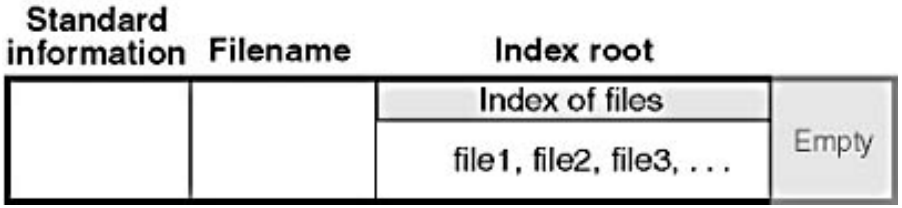
MS-DOS (8.3) per file i cui nomi non sono utilizzabili altrimenti (è possibile vederli con l'opzione `-x` del comando `dir`).

  - l'*algoritmo* garantisce che non possano essere creati due nomi uguali.

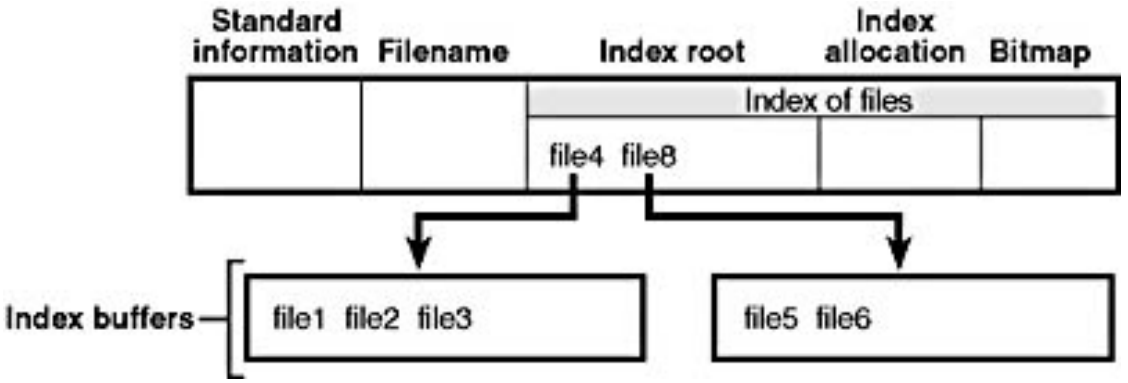
- Gli attributi iniziano con un header che contiene info come:
  - se è residente;
  - l'offset di inizio del valore dell'attributo;
  - la lunghezza del valore dell'attributo.



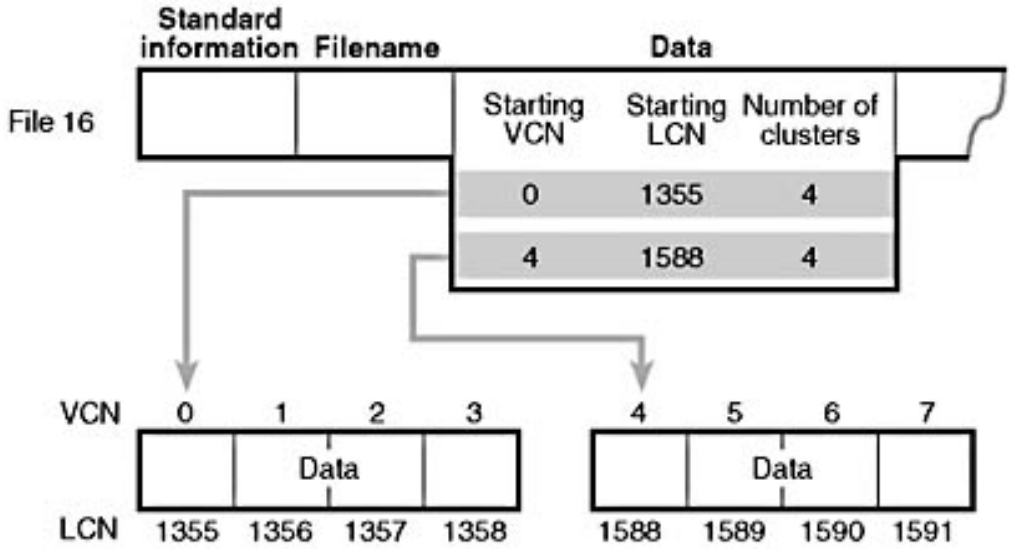
- Come detto gli attributi per un “piccolo” file o directory possono essere tutti residenti.
  Ad esempio il record nella MFT per una piccola directory ha la seguente struttura:



  Mentre, nel caso, di una directory contenente molti file si ha:



- Solo gli attributi che possono variare in dimensione possono essere “non residenti”
  - Le *standard information* ed il nome del file sono sempre residenti.
- Quanto gli attributi di un file (o di una directory) non possono essere accomodati nel record della MFT, NTFS utilizza un *mapping* tra VCN e LCN per individuare i cluster “esterni”:



Indicizzazione nel NTFS

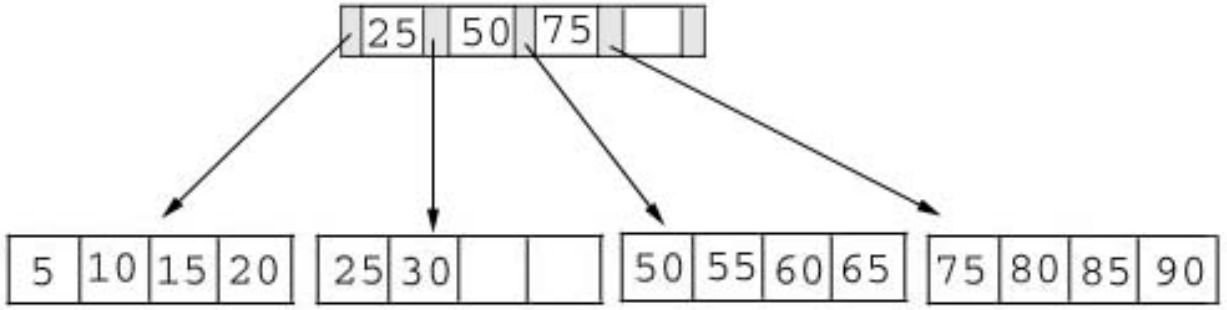
Nel NTFS una directory è semplicemente una collezione di nomi di file e delle loro *file reference* organizzata in modo da permettere un accesso efficiente.

- Il record nella MFT di una directory contiene nell'attributo *index root* una lista ordinata dei file nella directory.
- Oltre al nome sono riportati la *file reference* del file nella MFT, i *time stamp* e la dimensione. In questo modo risultano velocizzate le operazioni di *browsing* della directory.
- Ogni *index buffer* di 4 Kbyte contiene tra le 20 e le 30 *filename entry*.

NTFS usa la struttura dati **B+** che rappresenta un tipo di albero bilanciato ideale per organizzare dati su disco in quanto minimizza il

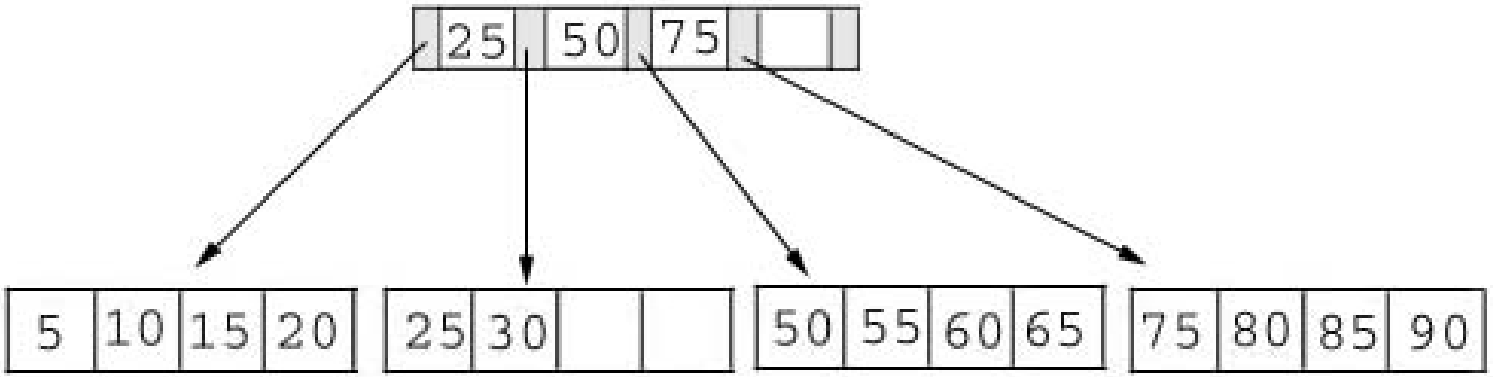
numero di accessi necessari per recuperare un'informazione.

- in un *B+ tree* ogni cammino dalla radice dell'albero ad una foglia ha la stessa lunghezza;
- ogni nodo intermedio ha tra $n/2$ ed n figli, dove n è fissato.



- l'attributo *index root* nel record (MFT) di una directory contiene i nomi dei file che svolgono il ruolo di indici di secondo livello del **B+** tree.

- Questi nomi di file possono avere un puntatore ad un *index buffer*;
- l'index buffer contiene nomi di file con valore lessicografico inferiore a quello del file a cui corrisponde il puntatore.



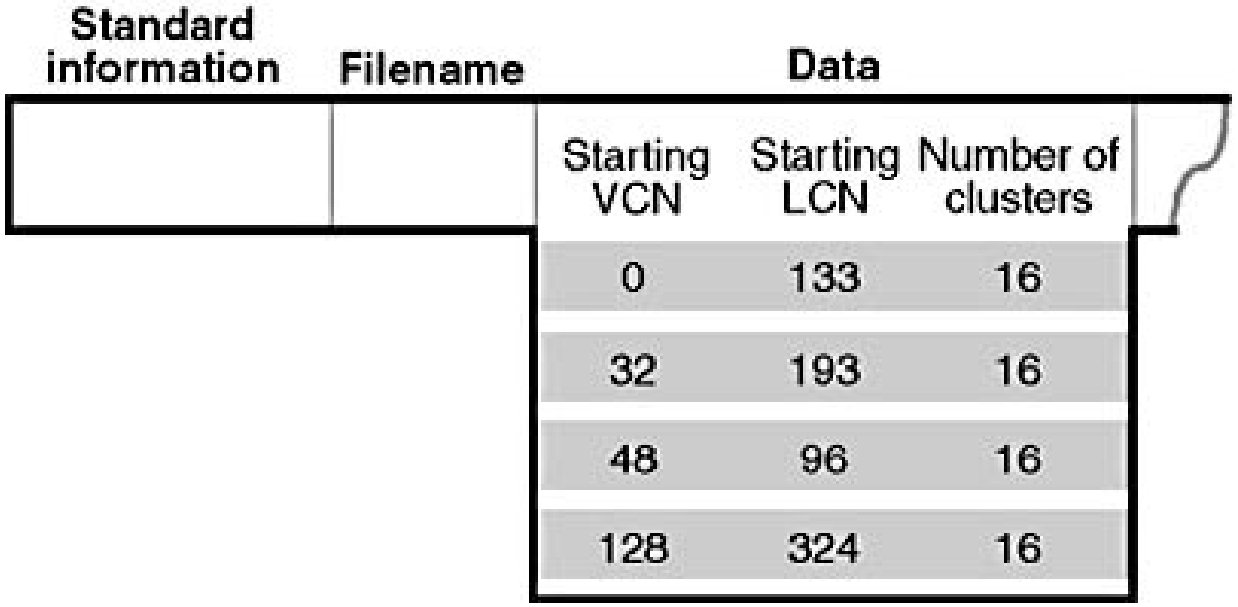
Compressione dei dati nel NTFS

NTFS supporta la compressione a diversi livelli:

- per file;
- per directory;
- per volume.

ma solo sui dati utente (non sui *metadati*).

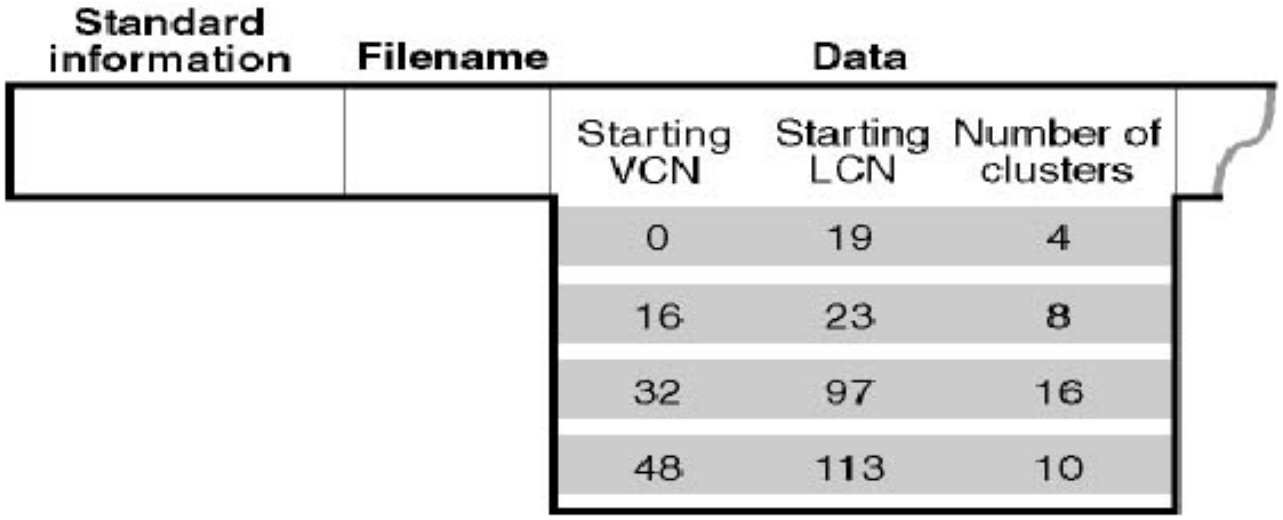
- La compressione a livello di file agisce "immediatamente".
- La compressione a livello di directory o di volume definisce lo stato per i **futuri** file (o directory) che sono creati successivamente.
- Una delle tecniche di compressione del NTFS è semplicemente rimuovere le sequenze di zeri.



- Se si cerca di leggere da un "buco" non allocato, NTFS ritorna degli zeri senza accedere il disco. Ovviamente in caso di scrittura, NTFS alloca lo spazio su disco e scrive realmente.

Compressione dei dati nel NTFS

- In generale, NTFS divide i dati del file da comprimere in unità di compressione aventi dimensione pari a 16 cluster.
  - La compressione avviene solo se si riduce di almeno un cluster la dimensione dell'unità.
  - NTFS assume che ogni *run* piú corto di 16 cluster non sia compresso



Reparse Point

- Il *Reparse Point* é un tipo di attributo che permette di estendere le funzionalitá di NTFS aggiungendo ulteriori informazioni alla *directory entry*.
- Ad una generica operazione (e.g. *open*) possono essere quindi associate operazioni diverse da quelle standard.
- Questo permette la creazione di *junction points* (link a directory), link simbolici ed *hard links* (anche verso directory)

Il *change journal* file

Il file `$Extend$UsnJrnl` viene creato quando viene abilitato il logging delle modifiche.

- Nel data stream `$J` vengono registrati i cambiamenti. In particolare:
  - la data della modifica;
  - il tipo di modifica (cancellazione, estensione, rinomina,...);
  - gli attributi del file (o della directory);
  - il nome del file (o della directory);
  - il *reference number* del file o della directory;
  - il *reference number* della directory che contiene il file;

Il file di log è gestito in modo da non andare mai in *overflow*.

- Quando viene raggiunta la dimensione massima predefinita, NTFS azzera i dati che precedono una finestra di dimensione anche essa predefinita.

Programmi per l'analisi del NTFS

Il pacchetto **Sleuthkit** (`http://www.sleuthkit.org`) include una serie di programmi per l'analisi di file system che supportano NTFS:

- `fsstat`
- `istat`
- `icat`