

Memoria secondaria

- Struttura del disco
- Scheduling del disco
- Gestione dell'unità a disco
- Gestione dello spazio di swap
- La struttura RAID
- Affidabilità dei dischi
- Cenni sui dispositivi di memoria terziaria

10. La memoria secondaria

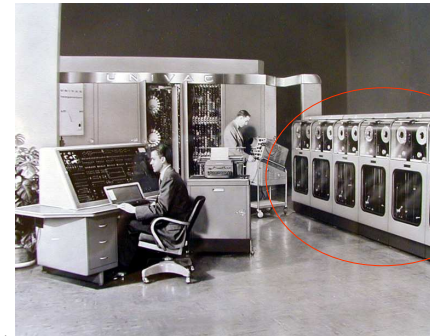
1

marco lapegna

La memoria secondaria

- Oltre alla memoria centrale (veloce, di piccole dimensioni e volatile) i calcolatori necessitano di un **supporto di memorizzazione di più grandi dimensioni e non volatile**

La memoria secondaria



I primi calcolatori utilizzavano come supporto di memorizzazione secondaria i **nastri magnetici**

1951: UNIVAC 1

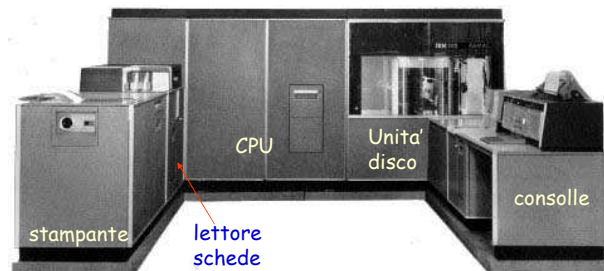
10. La memoria secondaria

2

marco lapegna

La memoria secondaria

- Il principale problema dei nastri e' il loro **accesso sequenziale**
- Nel 1956 l'IBM vende il primo calcolatore con hard disk (RAMAC 305)
 - Capacità 5 Mbyte (50 piatti magnetici di 1 metro di diametro)
 - Costo del disco 50000 USD (10000 USD al Mbyte)

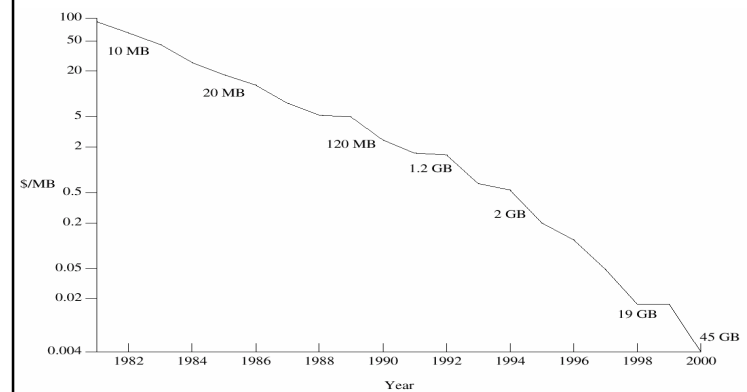


10. La memoria secondaria

3

marco lapegna

Costo al Mbyte degli hard disk

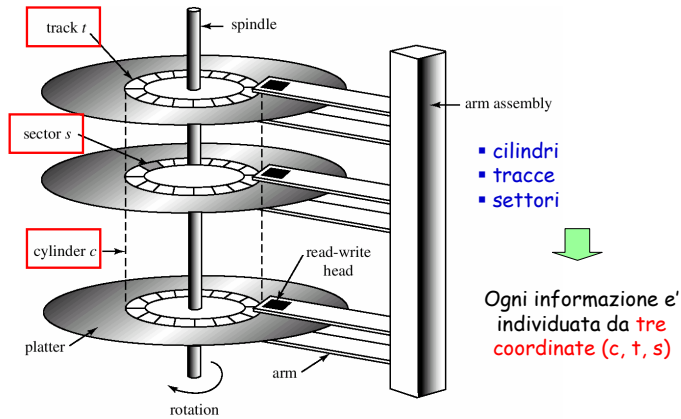


10. La memoria secondaria

4

marco lapegna

Organizzazione fisica dei dischi



10. La memoria secondaria

5

marco lapegna

Organizzazione logica dei dischi

- A causa della lentezza dei tempi di accesso sarebbe **impensabile** effettuare un accesso al disco e leggere un solo byte
- I dischi vengono indirizzati come vettori monodimensionali di blocchi logici, dove il **blocco logico** rappresenta la **minima unità di trasferimento** (dim. tipica 512 byte).
- L'array di blocchi logici viene mappato sequenzialmente nei settori del disco:
 - Il blocco 0 è il primo settore della prima traccia del cilindro più esterno.
 - La corrispondenza prosegue ordinatamente lungo la prima traccia, quindi lungo le rimanenti tracce del primo cilindro, e così via, dall'esterno verso l'interno.

10. La memoria secondaria

6

marco lapegna

Caratteristiche dei dischi

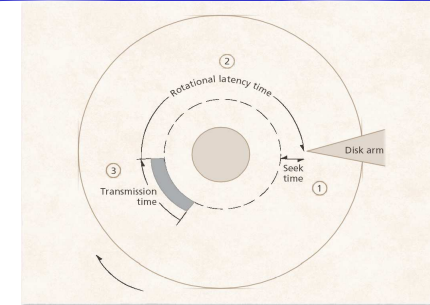
- Il **tempo di accesso** al disco si può scindere in :
 - **Tempo di ricerca (seek time)** — è il tempo impiegato per spostare la testina sulla traccia che contiene il settore desiderato.
 - **Latenza di rotazione (rotational latency)** — è il tempo necessario perché il disco ruoti fino a portare il settore desiderato sotto alla testina.
 - **Tempo di trasferimento (trasmission time)** — e' il tempo necessario per leggere/scrivere i 512 byte del blocco logico
- La **larghezza di banda** del disco è il numero totale di byte trasferiti, diviso per il tempo trascorso fra la prima richiesta del servizio e il completamento dell'ultimo trasferimento

10. La memoria secondaria

7

marco lapegna

Caratteristiche dei dischi



Modello	seek time (ms)	rotational latency (ms)
Maxtor DiamondMax Plus 9 (High-end desktop)	9.3	4.2
WD Caviar (High-end desktop)	8.9	4.2
Toshiba MK8025GAS (Laptop)	12.0	7.14
WD Raptor (Enterprise)	5.2	2.99
Cheetah 15K.3 (Enterprise)	3.6	2.0

10. La

1

Scheduling del disco

- Il SO è responsabile dell'uso efficiente dell'hardware. Per i dischi ciò significa garantire **tempi di accesso contenuti** e ampiezze di banda maggiori.
- Una richiesta di accesso al disco può venire soddisfatta immediatamente se unità a disco e controller sono disponibili; altrimenti **la richiesta deve essere aggiunta alla coda** delle richieste inevase per quell'unità.



- Il SO ha l'opportunità di scegliere quale delle richieste inevase servire per prima: **uso di un algoritmo di scheduling**

Scheduling del disco

- Il tempo di trasferimento dipende dal supporto hardware.
- L'ordine con cui vengono esaudite le richieste determinano il

tempo medio di ricerca
(quale traccia accedere per prima?)

Latenza media di rotazione
(quale settore della traccia accedere per prima?)



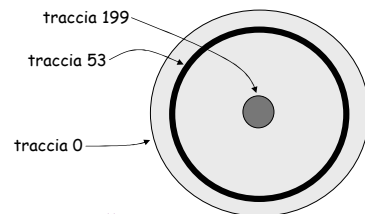
OBIETTIVO:
Massimo numero di richieste nell'unità di tempo
Oppure
Minimo tempo medio di accesso

Ottimizzazione del seek time

- Gli algoritmi di scheduling del disco verranno testati sulla **coda di richieste per le tracce (0-199)**:

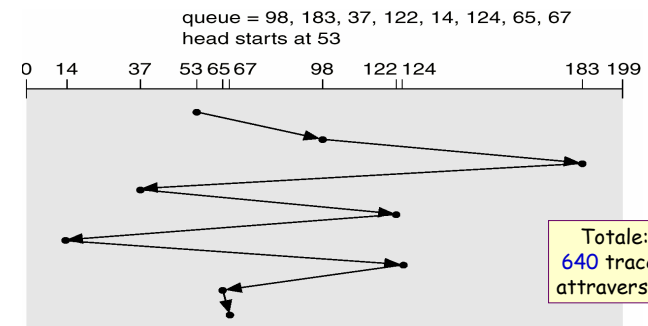
98, 183, 37, 122, 14, 124, 65, 67

La testina dell'unità a disco è inizialmente posizionata sulla **traccia 53**.



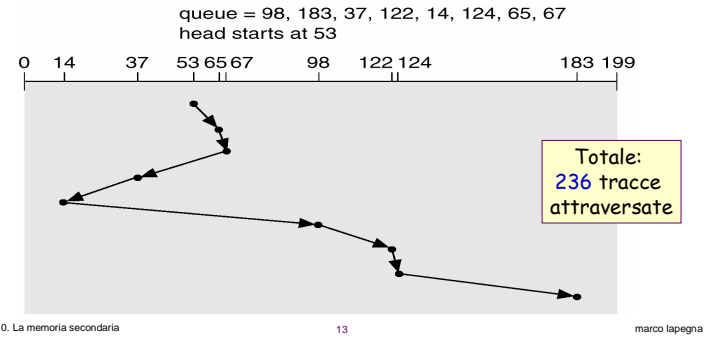
Scheduling First Come First Served (FCFS)

- Soddisfa le richieste secondo **l'ordine di arrivo**



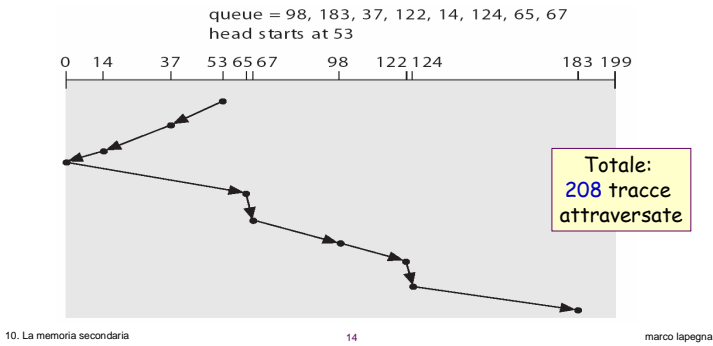
Scheduling Shortest Seek Time First (SSTF)

- Seleziona la richiesta con il **minor tempo di seek a partire dalla posizione corrente** della testina.
- Ordine di richieste molto **localizzato**
- Rischio di **attesa indefinita** delle richieste sulle tracce esterne e interne



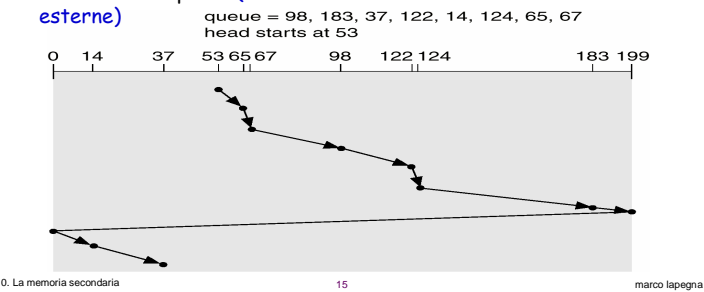
Scheduling SCAN

- Il braccio della testina si muove **da un estremo all'altro del disco**, servendo sequenzialmente le richieste;
- giunto ad un estremo inverte la direzione di marcia e, conseguentemente, l'ordine di servizio. È chiamato anche **algoritmo dell'ascensore**.



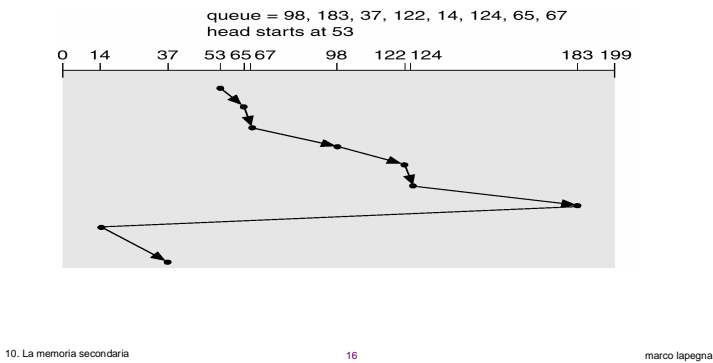
Scheduling Circular SCAN (CSCAN)

- Garantisce un **tempo di attesa più uniforme rispetto a SCAN**.
- La testina si muove da un estremo all'altro del disco servendo sequenzialmente le richieste. Quando raggiunge l'ultima traccia ritorna immediatamente all'inizio del disco, **senza servire richieste durante il viaggio di ritorno**.
- Considera le tracce come una lista circolare, con l'ultima traccia adiacente alla prima (**minore discriminazione delle tracce interne e esterne**)



Scheduling Circular LOOK (CLOOK)

- Versione ottimizzata (e normalmente implementata) di C-SCAN.
- Il braccio serve l'ultima richiesta in una direzione e poi inverte la direzione **senza arrivare al termine del disco**.



Scelta di un algoritmo di scheduling

- C-LOOK e C-SCAN hanno una varianza del tempo di risposta inferiore alle rispettive versioni non circolari
- Alcuni studi hanno comunque mostrato che la migliore politica di scheduling dovrebbe essere basata su due livelli
 - Versione circolare se il carico del disco e' alto
 - Versione non circolare se il carico del disco e' basso

Scelta di un algoritmo di scheduling

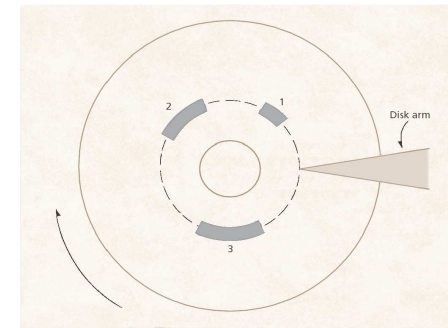
- SSTF è comune ed ha un comportamento "naturale".
- LOOK e C-LOOK hanno migliori prestazioni per sistemi con un grosso carico di lavoro per il disco (minor probabilità di blocco indefinito).
- Le prestazioni dipendono dal numero e dal tipo di richieste.
- Le richieste di servizio al disco possono essere influenzate dal metodo di allocazione dei file.
- L'algoritmo di scheduling del disco dovrebbe rappresentare un modulo separato del SO, che può essere rimpiazzato da un algoritmo diverso qualora mutassero le caratteristiche del sistema di calcolo.
- Sia SSTF che LOOK sono scelte ragionevoli per un algoritmo di tipo generale.

Ottimizzazione della latenza di rotazione

- Anni fa il tempo di ricerca dominava il tempo di accesso
- Oggi il tempo di ricerca e la latenza di rotazione hanno lo stesso ordine di grandezza
- Recentemente si e' cercato di migliorare le prestazioni del disco riducendo la latenza di rotazione

Shortest-latency-time-first scheduling (SLTF)

- SLTF serve per prima le richieste con la piu' piccola latenza di rotazione indipendentemente dall'ordine di arrivo
- Facile da realizzare
- Ottiene prestazioni quasi ottimali

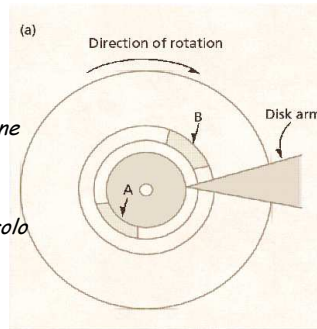


Shortest-positioning-time-first (SPTF)

- Tempo di posizionamento = tempo di ricerca + latenza di rotazione
- SPTF esaudisce per prima le richieste con il **piu' piccolo tempo di posizionamento**
- Buone prestazioni, ma rischio di attesa indefinita

A stessa traccia ma distante $\frac{1}{2}$ rotazione
B traccia adiacente e vicina

Se il seek time e' sufficientemente piccolo
viene servita prima B e poi A



10. La memoria secondaria

21

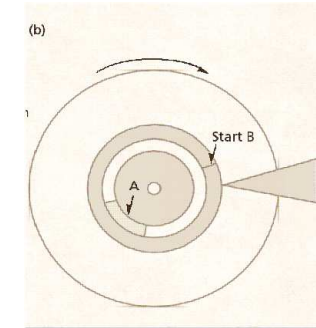
marco lapegna

Shortest-access-time-first scheduling (SATF)

- Tempo di accesso = tempo di posizionamento + tempo di trasmissione
- SATF esaudisce per prima le richieste con il **piu' piccolo tempo di accesso**
- Maggiore throughput ma rischio di attesa indefinita per richieste grandi

A lontana ma piccola
B vicina ma grande

viene servita prima A e poi B



10. La memoria secondaria

22

Altri compiti: formattazione e gestione

- Formattazione di basso livello, o formattazione fisica** —il disco e' suddiviso in settori con una struttura dati per ognuno:

- Intestazione
 - Area dati
 - Coda
- Di solito 256, 512 o 1024 byte

- Formattazione logica o "creazione di un file system"**

- Si partiziona il disco in uno o più gruppi di tracce (partizioni)
- Si crea una struttura dati per la "descrizione del disco (FAT) e una directory vuota

- Il blocco di *boot* inizializza il sistema.
 - Il *bootstrap* è memorizzato in ROM.
 - Il caricamento del sistema avviene grazie al *bootstrap loader*.

Esempio Configurazione
del disco in MS-DOS

sector 0	boot block
sector 1	FAT
	root directory
	data blocks (subdirectories)

10. La memoria secondaria

23

marco lapegna

Altri compiti: gestione dello spazio di swap

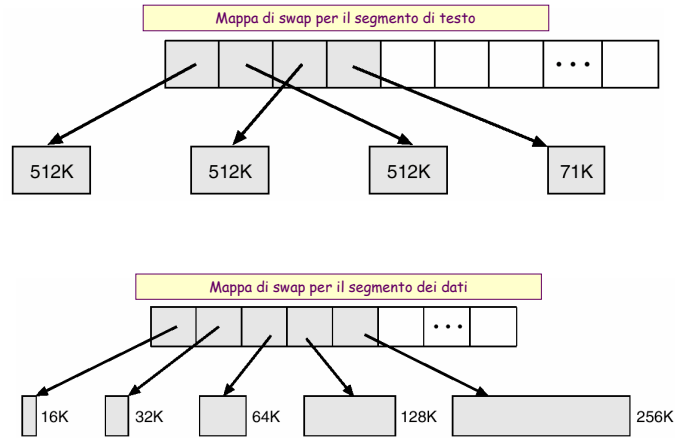
- Spazio di swap:** la memoria virtuale impiega lo spazio su disco come un'estensione della memoria centrale.
- Lo spazio di swap può essere ricavato all'interno del normale file system o, più comunemente, si può trovare in una partizione separata del disco.
- Esempio: Gestione dello spazio di swap di 4.3BSD UNIX:**
 - Alloca lo spazio di swap all'avvio del processo; si riserva spazio sufficiente per il *segmento di testo* e il *segmento dei dati*.
 - Il *kernel* impiega due *mappe di swap* per ogni processo per tener traccia dell'uso dello spazio di swap.
 - La mappa per il segmento di *testo* indicizza blocchi di 512K.
 - La mappa per il segmento dei dati ha dimensione fissa, ma contiene indirizzi relativi a blocchi di dimensioni variabili. Ogni nuovo blocco aggiunto è grande il doppio del precedente.

10. La memoria secondaria

24

marco lapegna

Esempio: 4.3 BSD UNIX



10. La memoria secondaria

25

marco lapegna

Altri compiti: ottimizzazione

- Deframmentazione
 - Sistema dati correlati tra loro in settori contigui
 - Diminuisce il numero di ricerche necessarie sul disco
 - Partizionamento può contribuire a ridurre la frammentazione
- Compressione
 - I dati consumano meno spazio sul disco
 - migliora il trasferimento e il tempo di accesso
 - Aumenta l'overhead necessario per la compressione / decompressione

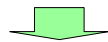
10. La memoria secondaria

26

marco lapegna

Strutture RAID

La velocità di trasferimento dei dischi **non cresce** allo stesso modo della velocità operativa della CPU



Peggioramento del throughput



IDEA

Utilizzare più dischi da utilizzare "in parallelo"



RAID, Redundant Array of Independent Disks

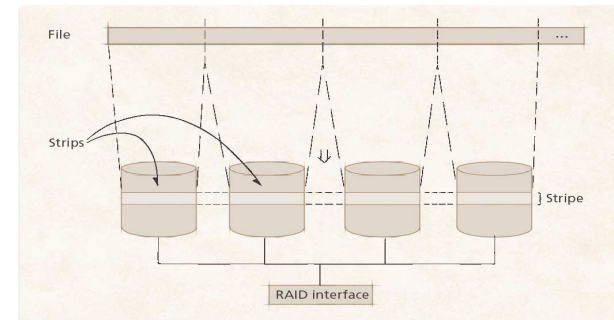
10. La memoria secondaria

27

marco lapegna

Caratteristiche dei RAID

- I dischi sono visti come un'unica unità di memorizzazione
- I file sono divisi in strisce (strips) distribuite sui vari dischi
- I pezzi della striscia relativi allo stesso file hanno la stessa locazione in ogni disco



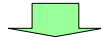
10. La memoria secondaria

28

marco lapegna

Caratteristiche dei RAID

- La gestione dei RAID e' un'operazione dispendiosa che puo' essere causa di rallentamento della CPU e del S.O.



Hardware specializzato dedicato allo scopo (RAID controller)

- Un maggior numero di dischi aumenta la probabilita' di un guasto con conseguente perdita di dati



Ridondanza delle informazioni memorizzate

Quindi

Motivazioni dei RAID

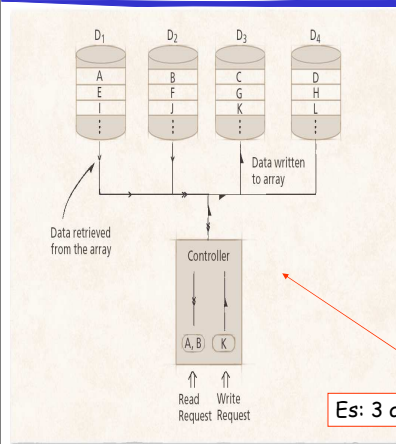
Efficienza
(maggiore
velocita' di
accesso ai dischi)

Affidabilita'
(maggiore
sicurezza di non
perdere dati)



Esistono molteplici schemi di gestione dei RAID chiamati
Livelli RAID

Livello 0 (striping)



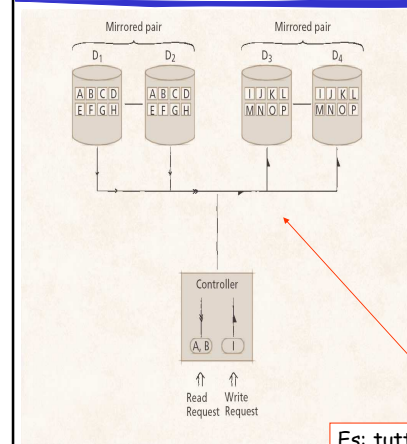
- Piu' semplice schema RAID
- striping senza ridondanza
- elevata efficienza
- rischio di perdere i dati



indicato quando le esigenze di performance superano quelle dell'affidabilita'

Es: 3 accessi contemporanei

Livello 1 (mirroring)



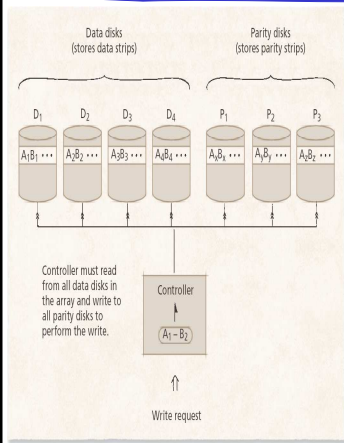
- Ogni disco ha un suo disco di copia
- ridondanza senza striping
- Buona performance in lettura, meno in scrittura
- Alto overhead di storage



Adatto quando le esigenze di affidabilita' sono importanti

Es: tutti i dati sono duplicati

Livello 2 (Hamming error correcting code)



- **Ridondanza e striping**
- **Dischi per i dati e dischi per "bit di parita'" per correggere errori**
- **Overhead di storage (ma meno del livello 1) e performance**

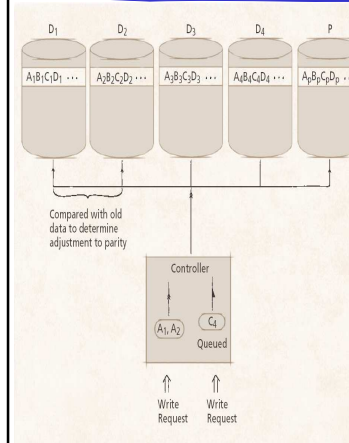
Poco utilizzato

10. La memoria secondaria

33

marco lapegna

Livello 3 (XOR error correcting code)



- **Ridondanza e striping**
- **simile al livello 2**
- **correzione di errori con diverso meccanismo**
- **1 solo disco per i bit di parita'**
- **scarsa efficienza (bottleneck del disco di parita')**

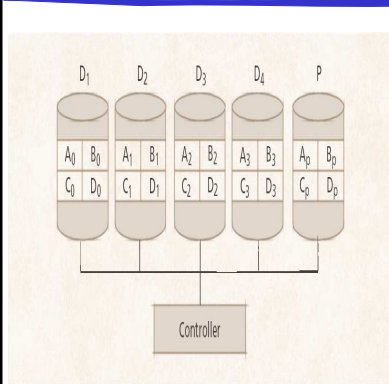
Poco utilizzato

10. La memoria secondaria

34

marco lapegna

Livello 4 (block XOR eec)



- **Ridondanza e striping**
- **versione a blocchi del livello 3**

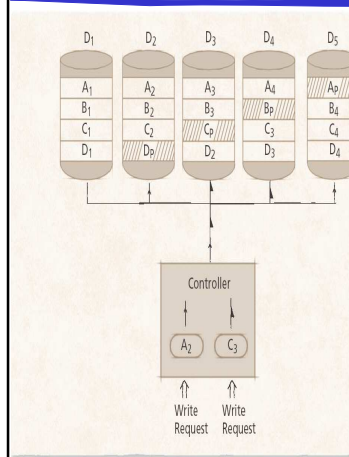
Poco utilizzato

10. La memoria secondaria

35

marco lapegna

Livello 5 (block distributed XOR eec)



- **I blocchi di parita' sono distribuiti sui dischi dei dati**
- **Rimozione del bottleneck**
- **Piu' veloce dei livelli 2, 3 e 4**
- **Difficile da implementare**

Miglior compromesso tra efficienza e affidabilita'

Soluzione piu' utilizzata per i sistemi general purpose

10. La memoria secondaria

36

marco lapegna

Livelli RAID

- Ce ne sarebbero tanti altri...
 - RAID level 6 utilizza altre informazioni per migliorare la fault tolerance
 - Varie combinazioni
 - Livelli 0+1, 1+0, 0+3, 0+5, 1+5, ...
- Ognuno con un *differente compromesso efficienza / affidabilità*

Dispositivi di memorizzazione terziaria

- I compiti principali del SO sono la gestione dei dispositivi fisici e la presentazione di una macchina virtuale astratta alle applicazioni utente.
- La caratteristica fondamentale della **memorizzazione terziaria** è **il basso costo**.
- Generalmente, viene effettuata su **mezzi rimovibili**, ad esempio floppy disk, CD-ROM, nastri magnetici.
- La maggior parte dei SO gestisce i dischi rimovibili come i dischi fissi: un nuovo supporto viene formattato, e un file system vuoto viene generato sul disco.
- I nastri sono presentati come un mezzo di memorizzazione a basso livello, e le applicazioni non aprono un file, ma l'intero nastro.
- In genere l'unità a nastro è riservata per un'applicazione alla volta.