RAID

Redundant Array of Inexpensive (Independent) Disks

Accesso a disco è essenziale che avvenga in modo efficiente. Si è cercato, se quello era architetturalmente logico (file system), di provare a parallelizzare come specif. di lettura: provo a leggere più blocchi contemporaneamente per un file. Ho più dischi leggibili in // posso usare dischi aggiuntivi anche per rendere + affidabile memorizzazione dei dati.

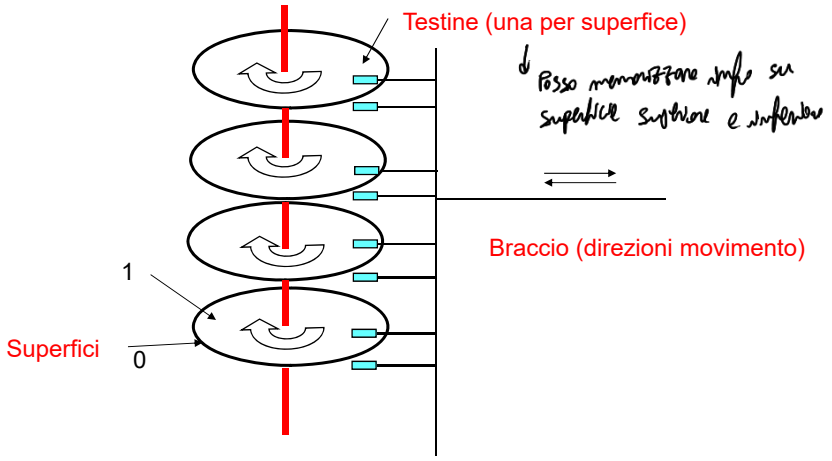
1

I = independent: dischi letti contemporaneamente (migliora lettura di grossi dati)
R = parte di questi dischi usata per rendere affidabile sistema (memorizza copie dati, bit di parità e di correzione...)

Disco magnetico

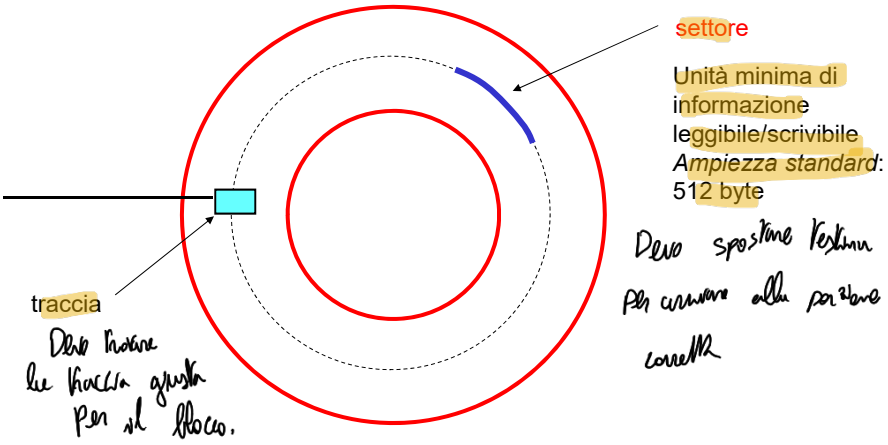
- Costituito da un insieme di *piatti* rotanti (da 1 a 15)
  - Piatti rivestiti di una superficie magnetica
- Esiste una testina (bobina) per ogni faccia del piatto
  - Generalmente piatti a doppia faccia
- Le testine di facce diverse sono collegate tra di loro e si muovono contemporaneamente in modo solidale
- Velocità di rotazione costante (ad es. 10000 RPM)
- La superficie del disco è suddivisa in anelli concentrici (*tracce*)
- Registrazione seriale su tracce concentriche
  - 1000-5000 tracce
  - Tracce adiacenti separate da spazi

Hardware del disco (1)



Struttura di un disco rigido

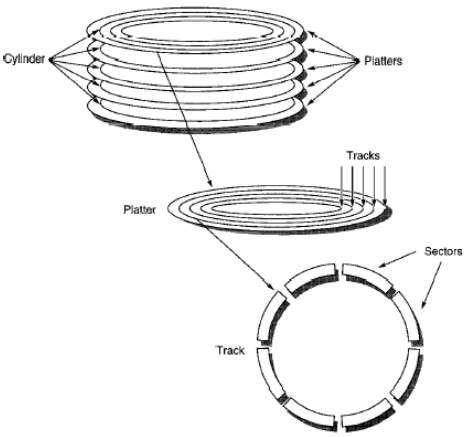
Hardware del disco (2)



- Ogni superficie è divisa in tracce concentriche (una per ogni possibile posizione della testina)

Settori

- Ciascuna traccia è divisa in *settori*
  - Settore: la più piccola unità che può essere trasferita (scritta o letta)
  - Centinaia di settori per traccia, generalmente di lunghezza fissa
  - Il settore contiene un ID del settore, i dati e un codice di correzione di errore: la *capacità formattata* scende del 15%
- Tracce sovrapposte su piatti diversi formano un *cilindro*
- Per individuare un settore va specificato il cilindro, il piatto ed il numero del settore



Formattazione del disco

- Formattazione a basso livello : Struttura di un settore

Permette alla testina di capire che sta iniziando un nuovo settore, fornisce il numero del settore etc

Codici correttori di errore : dati in più per accorgersi se la lettura è andata bene

Lettura/scrittura di un disco

- Posizionamento della testina sul cilindro desiderato (*tempo di seek*)
  - Da 3 a 14 ms (può diminuire se si usano delle ottimizzazioni)
  - Dischi di diametro piccolo permettono di ridurre il tempo di posizionamento
- Attesa che il settore desiderato ruoti sotto la testina di lettura/scrittura (*tempo di rotazione*)
  - In media è il tempo per ½ rotazione
  - Tempo di rotazione medio = 0.5/numero di giri al secondo

    Es.: 7200 RPM → Tempo di rotazione medio = 0.5/(7200/60) = 4.2 ms
- Operazione di lettura o scrittura di un settore (*tempo di trasferimento*)
  - Da 30 a 80 MB/sec (fino a 320 MB/sec se il controllore del disco ha una cache built-in)
  - In più: tempo per le operazioni del disk controller (*tempo per il controller*)

Tempo medio di R/W

- Calcolo del tempo medio necessario a leggere o scrivere un settore di 512 byte sapendo che:
  - Il disco ruota a 10000 RPM
  - Il tempo medio di seek è 6 ms
  - Il transfer rate è di 50 MB/sec
  - L'overhead del controller è di 0.2 ms

Tempo di seek + tempo medio di rotazione + tempo medio di trasferimento + overhead del controller =

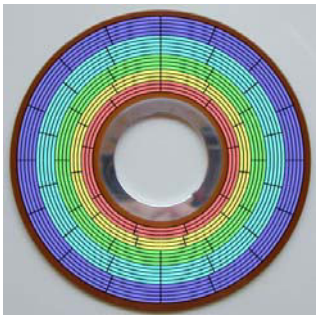
= 6 ms + (0.5/(10000/60))·1000 ms + 0.5 KB/(50 MB/sec)+0.2 ms

= (6.0 + 3.0 + 0.01 + 0.2) ms = 9.2 ms

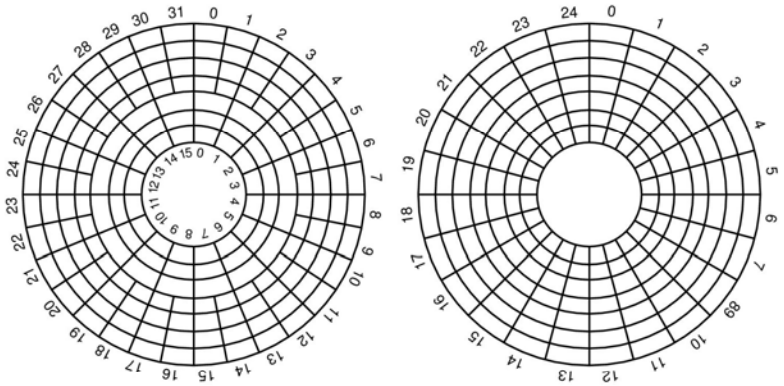
Organizzazione dei dati sul disco

- Nei dischi più vecchi
  - Ogni traccia conteneva lo stesso numero di settori
  - Le tracce esterne (più lunghe) memorizzavano informazioni con densità minore

- Nei dischi recenti
  - Per aumentare le prestazioni, si utilizzano maggiormente le tracce esterne: *zoned bit recording* (o multiple zone recording)
  - Tracce raggruppate in *zone* sulla base della loro distanza dal centro
    - Una zona contiene lo stesso numero di settori per traccia
  - Più settori per traccia nelle zone esterne rispetto a quelle interne
    - Densità di registrazione (quasi) costante



Disco virtuale



- Geometria fisica di un disco con due zone:
  - Zona esterna: 4 tracce da 32 settori =128 settori
  - Zona interna: 4 tracce da 16 settori = 64 settori
- Una possibile geometria virtuale per lo stesso disco
  - Unica zona: 8 tracce da 24 settori=192 settori

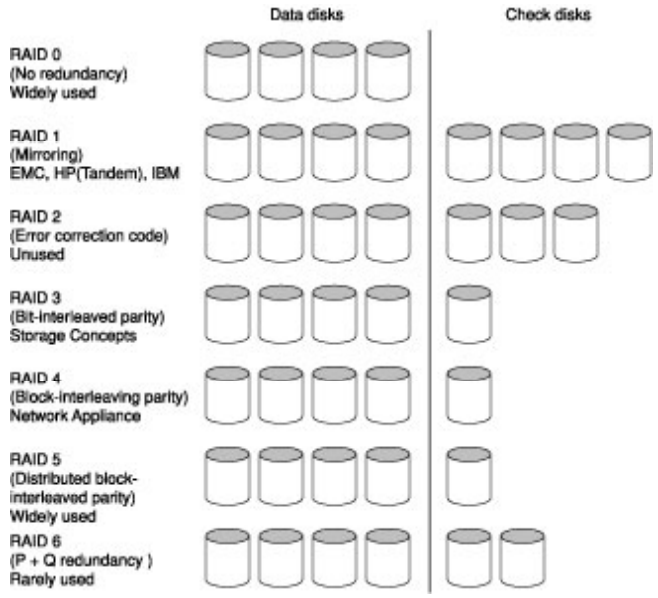
Ci serve sapere disco, superficie, traccia e n. settore.

RAID (prestazioni)

- Idea di Patterson et al. nel 1987: usare in parallelo più dischi per aumentare le prestazioni dei dischi
- *Redundant Array of Independent Disks*
- Sfrutta il parallelismo per rendere l'accesso al disco più veloce
- Il controllore RAID mostra l'array come un unico disco al resto del sistema
- I dati sono distribuiti sui dischi in modo da favorire le letture parallele di parti dello stesso file
  - diverse strategie : RAID livello 0, 1, ….

stesso file su dischi diversi. Posso leggere pezzi di questo file su blocchi diversi.

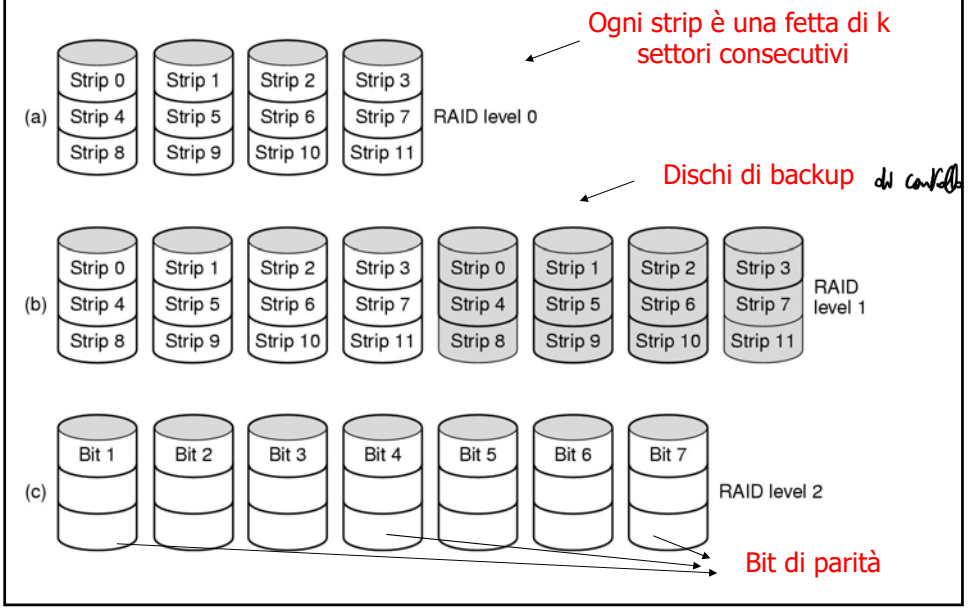
Livelli RAID



Tutte a sinistra sono dischi dedicati ad affidabilità.
RAID 1 spreco più di tutti su affidabilità. Ogni disco ha suo clone. Costo: mi serve il doppio dello spazio.

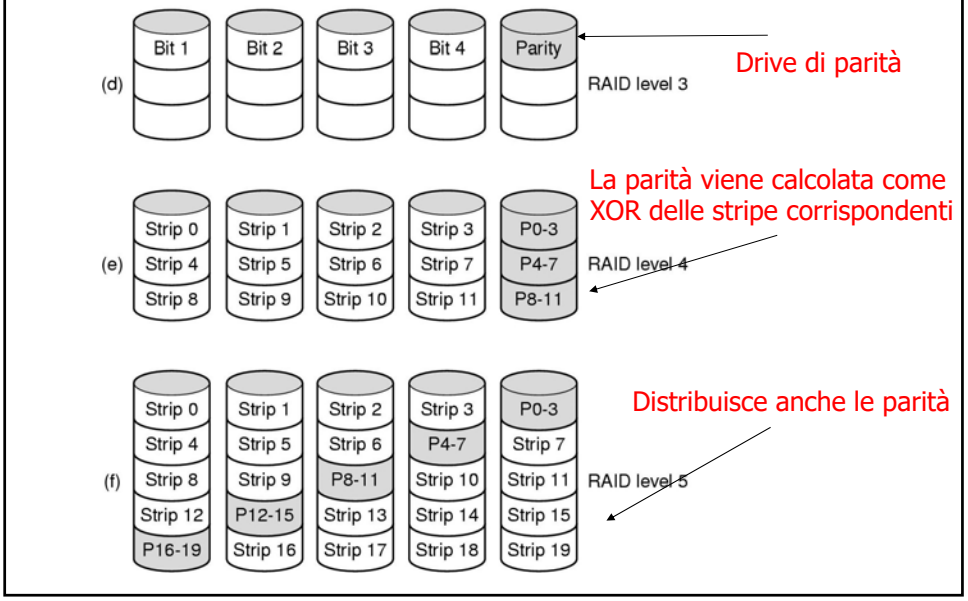
* RAID hanno molti diversi di salvare informazione.

Hardware di un RAID



Differenza organizzativa: come vengono distribuite info dei file su vari dischi.
Qui su sequenze di blocchi. Le run (chiamate strip) vengono scritte su pezzi diversi. Posso leggere al più 4 strip in parallelo. Strip contiene sequenza di blocchi *

Hardware di un RAID (2)



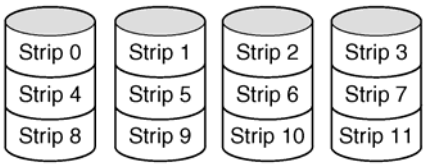
Qui la lettura viene fatta in maniera simultanea su dischi diversi.
Dato diviso per bit e non per blocchi: dato memorizzato in parallelo e bloc associato alla stessa strip.

RAID (affidabilità)

- Un array di dischi (senza ridondanza dei dati) è inaffidabile!
  Affidabilità di un array da N dischi = Affidabilità di 1 disco/N
- Replicando i dati sui vari dischi dell'array e definendo un'organizzazione dei dati memorizzati sui dischi in modo da ottenere un'elevata affidabilità (*tolleranza ai guasti*)
- RAID: **Redundant Array of Inexpensive (Independent) Disks**
  - Insieme di dischi a basso costo ma coordinati in azioni comuni per ottenere diversi livelli di tolleranza ai guasti

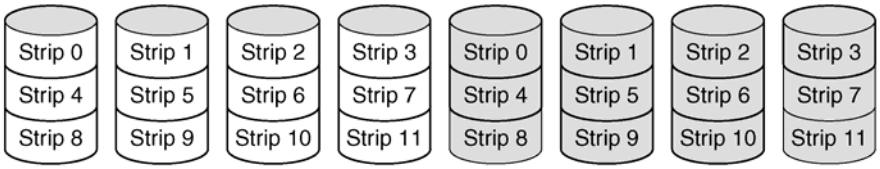
RAID 0

- Nessuna ridondanza dei dati
- Solo *striping* dei dati
  - Striping: allocazione di blocchi logicamente sequenziali (memorizzanti p.e. lo stesso file, che quindi è suddiviso in più blocchi) su dischi diversi per aumentare le prestazioni rispetto a quelle di un singolo disco
  - Lettura e scrittura in parallelo di *stripe* (strisce) su dischi diversi
- Non è un vero RAID perché non c'è nessuna ridondanza
- E' la migliore soluzione in scrittura, perchè non ci sono overhead per la gestione della ridondanza, ma non in lettura (se più blocchi appartengono allo stesso disco)



RAID 1

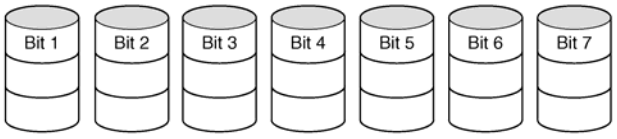
- Mirroring (o shadowing)
- Ciascun disco è completamente replicato su un disco ridondante (mirror), avendo così sempre una copia
  - Usa il doppio dei dischi rispetto a RAID 0
- Ottime prestazioni in lettura
  - Molte possibilità di migliorare le prestazioni (es.: leggere dal disco con il minimo tempo di seek, leggere due file contemporaneamente su dischi "gemelli")
- Una scrittura logica richiede due scritture fisiche
- E' la soluzione RAID più costosa



Ha un vantaggio prestazionale. Con Raid 0 parallelismo si ha se leggo blocchi che appartengono a dischi diversi. Con RAID 1 con disco gemello, posso leggere in // anche blocchi su stesso disco. Contro: deve scrivere due volte ogni volta che scrivo un blocco. Soluzione costosa.

RAID 2

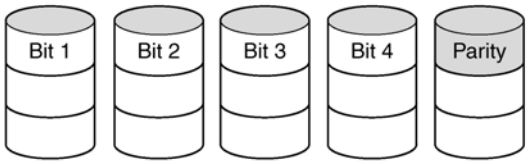
- Rivelazione e correzione degli errori (*codice di Hamming*)
- *Striping a livello di parola o di byte* (in RAID 0 e 1 strip di settori) Dati distribuiti per parola o byte
  - Es. in figura: 4 bit (*nibble*) più 3 bit (codice di Hamming a 7 bit)
- Svantaggio: rotazione dei dischi *sincronizzata* (Testine non sono indipendenti)
- Resiste a guasti semplici
- Ad ogni scrittura bisogna aggiornare i dischi di "parità"
  anche per la modifica di un singolo bit di informazione
- Forte *overhead* per pochi dischi (in figura +75%), ha senso con molti dischi, ad esempio:
  - Parola da 32 bit+(6+1) bit di parità ⇒ 39 dischi
  - Overhead del 22% (=7/32)
- In disuso



Ogni bit di controllo in più, molto in disco in più.

RAID 3

- Un bit di parità
- Resiste ad un guasto (transiente o permanente) alla volta
- Overhead abbastanza contenuto
- Solo un'operazione su disco per volta
  - Ciascuna operazione coinvolge tutti i dischi
- Soluzione diffusa per applicazioni che operano su grandi quantità di dati in lettura (come nei video games o nelle fruizioni multimediali.



RAID 3: esempio

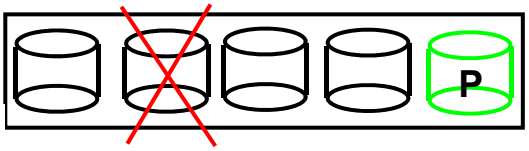
P contiene il bit di parità dei bit memorizzati negli altri dischi

$b_4(i) = b_0(i) \oplus b_1(i) \oplus b_2(i) \oplus b_3(i)$

Se un disco fallisce (in modo transiente o permanente), utilizzando P, i bit degli altri dischi si recupera l'informazione mancante

$b_1(i) = b_0(i) \oplus b_4(i) \oplus b_2(i) \oplus b_3(i)$

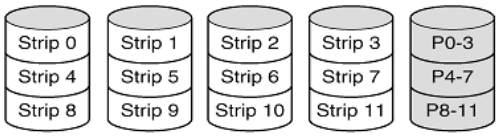
Overhead accettabile (in genere 1/(n-1) se n sono i dischi utilizzati)



1	1	1	0	1
0	1	0	1	0
0	0	0	1	1
1	0	1	1	1
0	1	0	1	0
0	1	0	1	0
1	0	1	0	0
1	1	1	0	1

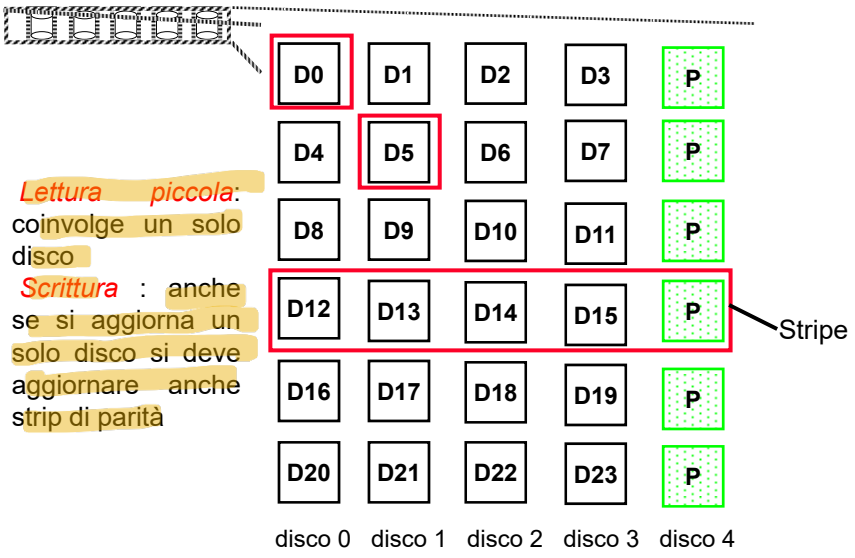
RAID 4

- Evoluzione di Raid 3 con striping a blocchi (come RAID 0)
  - la stripe nell'ultimo disco contiene i bit di parità dell'insieme di bit omologhi di tutte le altre stripe
- No rotazione sincronizzata (come in RAID 2 e 3)
- Resiste a guasti singoli (transienti e permanenti)
- Consente letture indipendenti sui diversi dischi
  - Se si legge una quantità di dati contenuta in una sola strip
- Il disco di parità è il collo di bottiglia



Scrittura coinvolge anche disco di parità che deve essere ricalcolato

RAID 4: lettura e scrittura



Lettura piccola: coinvolge un solo disco

Scrittura : anche se si aggiorna un solo disco si deve aggiornare anche strip di parità

Scrittura in RAID 3 e RAID 4

- *Opzione 1*: si leggono i dati sugli altri dischi, si calcola la nuova parità P' e la si scrive sul disco di parità (come per RAID 3)

$$S'_4(i) = S'_0(i) \oplus S_1(i) \oplus S_2(i) \oplus S_3(i)$$

  - Es.: 1 scrittura logica = 3 letture fisiche + 2 scritture fisiche
- *Opzione 2*: poiché il disco di parità ha la vecchia parità, si confronta il vecchio dato D0 con il nuovo D0', si aggiunge la differenza a P, e si scrive P' sul disco di parità

$$S'_4(i) = S'_0(i) \oplus S_1(i) \oplus S_0(i) \oplus S_0(i) \oplus S_2(i) \oplus S_3(i)$$

$$S'_4(i) = S_4(i) \oplus S'_0(i) \oplus S_0(i)$$

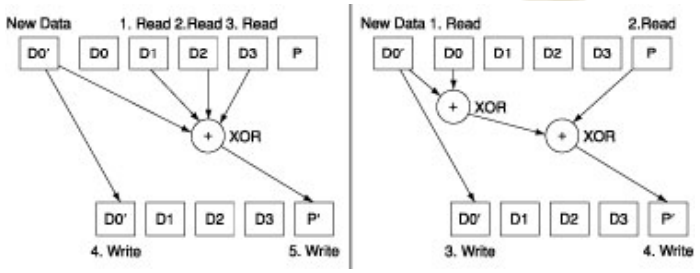
Aggiunto nella somma

$x \oplus x = 0$

$x \oplus 0 = x$

So ⊕ S1 ⊕ S2 ⊕ S3 = vecchia parità

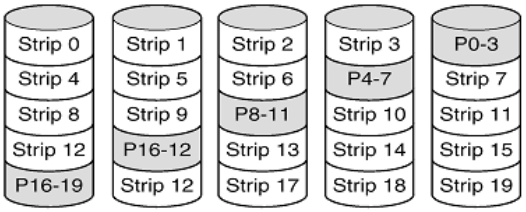
  - Es.: 1 scrittura logica = 2 letture fisiche + 2 scritture fisiche



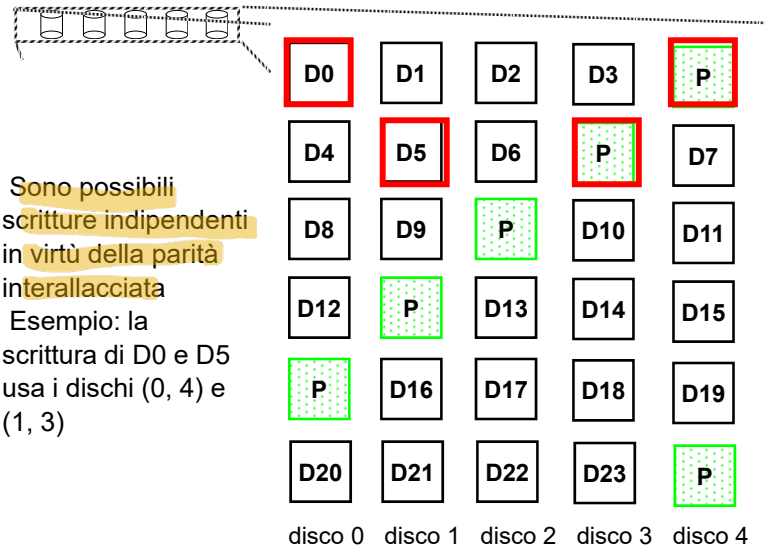
Scrittura coinvolge sempre disco di parità = collo di bottiglia

RAID 5

- Blocchi di parità distribuita
- Le stripe di parità sono distribuite su più dischi in modalità round-robin (circolare)
- Si evita il collo di bottiglia del disco di parità in RAID 4
- La scrittura è gestita come in RAID 4



RAID 5: scrittura



Sono possibili scritture indipendenti in virtù della parità interallacciata

Esempio: la scrittura di D0 e D5 usa i dischi (0, 4) e (1, 3)

RAID 6

- Ridondanza P+Q (si aumenta la distanza di Hamming)
- Anziché la parità, si usa uno schema che consente di ripristinare anche un secondo guasto
  - la singola parità consente di recuperare un solo guasto
- Overhead di memorizzazione doppio rispetto a RAID 5

Maggiore ridondanza.